HINTS/SOLUTIONS TO SELECTED EXERCISES

Section 1.1 Review

1. A set is a collection of objects.

2. A set may be defined by listing the elements in it. For example, $\{1, 2, 3, 4\}$ is the set consisting of the integers 1, 2, 3, 4. A set may also be defined by listing a property necessary for membership. For example,

$$\{x \mid x \text{ is a positive, real number}\}$$

defines the set consisting of the positive, real numbers.

3.

Set	*Description*	*Examples of Members*
Z	Integers	$-3, 2$
Q	Rational numbers	$-3/4, 2.13074$
R	Real numbers	$-2.13074, \sqrt{2}$
$\mathbf{Z}^+$	Positive integers	$2, 10$
$\mathbf{Q}^+$	Positive rational numbers	$3/4, 2.13074$
$\mathbf{R}^+$	Positive real numbers	$2.13074, \sqrt{2}$
$\mathbf{Z}^-$	Negative integers	$-12, -10$
$\mathbf{Q}^-$	Negative rational numbers	$-3/8, -2.13074$
$\mathbf{R}^-$	Negative real numbers	$-2.13074, -\sqrt{2}$
$\mathbf{Z}^{nonneg}$	Nonnegative integers	$0, 3$
$\mathbf{Q}^{nonneg}$	Nonnegative rational numbers	$0, 3.13074$
$\mathbf{R}^{nonneg}$	Nonnegative real numbers	$0, \sqrt{3}$

4. The cardinality of X (i.e., the number of elements in X)

5. $x \in X$ **6.** $x \notin X$ **7.** $\varnothing$

8. Sets X and Y are equal if they have the same elements. Set equality is denoted $X = Y$.

9. Prove that for every x, if x is in X, then x is in Y, and if x is in Y, then x is in X.

10. Prove one of the following: (a) There exists x such that $x \in X$ and $x \notin Y$. (b) There exists x such that $x \notin X$ and $x \in Y$.

11. X is a subset of Y if every element of X is an element of Y. X is a subset of Y is denoted $X \subseteq Y$.

12. To prove that X is a subset of Y, let x be an arbitrary element of X and prove that x is in Y.

13. Find x such that x is in X, but x is not in Y.

14. X is a proper subset of Y if $X \subseteq Y$ and $X \neq Y$. X is a proper subset of Y is denoted $X \subset Y$.

15. To prove that X is a proper subset of Y, prove that X is a subset of Y and find x in Y such that x is not in X.

16. The power set of X is the collection of all subsets of X. It is denoted $\mathcal{P}(X)$.

17. X *union* Y is the set of elements that belong to either X or Y or both. It is denoted $X \cup Y$.

18. The union of $\mathcal{S}$ is the set of elements that belong to at least one set in $\mathcal{S}$. It is denoted $\cup\mathcal{S}$.

19. X *intersect* Y is the set of elements that belong to both X and Y. It is denoted $X \cap Y$.

20. The intersection of $\mathcal{S}$ is the set of elements that belong to every set in $\mathcal{S}$. It is denoted $\cap\mathcal{S}$.

21. $X \cap Y = \varnothing$

22. A collection of sets $\mathcal{S}$ is pairwise disjoint if, whenever X and Y are distinct sets in $\mathcal{S}$, X and Y are disjoint.

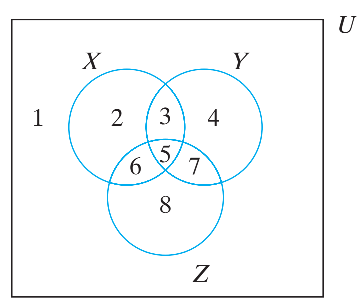
23. The difference of X and Y is the set of elements that are in X but not in Y. It is denoted $X - Y$.

24. A universal set is a set that contains all of the sets under discussion.

25. The complement of X is $U - X$, where U is a given universal set. The complement of X is denoted $\overline{X}$.

26. A Venn diagram provides a pictorial view of sets. In a Venn diagram, a rectangle depicts a universal set, and subsets of the universal set are drawn as circles. The inside of a circle represents the members of that set.

27.



Region 1 represents elements in none of X, Y, or Z. Region 2 represents elements in X, but in neither Y nor Z. Region 3 represents elements in X and Y, but not in Z. Region 4 represents elements in Y, but in neither X nor Z. Region 5 represents elements in X, Y, and Z. Region 6 represents elements in X and Z, but not in Y. Region 7 represents elements in Y and Z, but not in X. Region 8 represents elements in Z, but in neither X nor Y.

28. $(A \cup B) \cup C = A \cup (B \cup C)$, $(A \cap B) \cap C = A \cap (B \cap C)$

29. $A \cup B = B \cup A$, $A \cap B = B \cap A$

30. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$, $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

31. $A \cup \varnothing = A$, $A \cap U = A$

32. $A \cup \overline{A} = U$, $A \cap \overline{A} = \varnothing$

33. $A \cup A = A$, $A \cap A = A$

34. $A \cup U = U$, $A \cap \varnothing = \varnothing$

35. $A \cup (A \cap B) = A$, $A \cap (A \cup B) = A$

36. $\overline{\overline{A}} = A$

37. $\overline{\varnothing} = U$, $\overline{U} = \varnothing$

38. $\overline{(A \cup B)} = \overline{A} \cap \overline{B}$, $\overline{(A \cap B)} = \overline{A} \cup \overline{B}$

39. A collection $\mathcal{S}$ of nonempty subsets of X is a partition of X if every element in X belongs to exactly one member of $\mathcal{S}$.

40. The Cartesian product of X and Y is the set of all ordered pairs (x, y) where $x \in X$ and $y \in Y$. It is denoted $X \times Y$.

41. The Cartesian product of $X_1, X_2, \ldots, X_n$ is the set of all n-tuples $(x_1, x_2, \ldots, x_n)$ where $x_i \in X_i$ for $i = 1, \ldots, n$. It is denoted $X_1 \times X_2 \times \cdots \times X_n$.

Section 1.1

1. $\{1, 2, 3, 4, 5, 7, 10\}$

4. $\{2, 3, 5\}$

7. $\varnothing$

10. U

13. $\{6, 8\}$

16. $\{1, 2, 3, 4, 5, 7, 10\}$

17. 0

20. 5

21. If $x \in A$, then x is one of 3, 2, 1. Thus $x \in B$. If $x \in B$, then x is one of 1, 2, 3. Thus $x \in A$. Therefore, $A = B$.

24. If $x \in A$, then x satisfies $x^2 - 4x + 4 = 1$. Factoring $x^2 - 4x + 4$, we find that $(x - 2)^2 = 1$. Thus $(x - 2) = \pm 1$. If $(x - 2) = 1$, then $x = 3$. If $(x - 2) = -1$, then $x = 1$. Since $x = 3$ or $x = 1$, $x \in B$. Therefore $A \subseteq B$.

If $x \in B$, then $x = 1$ or $x = 3$. If $x = 1$, then

$$x^2 - 4x + 4 = 1^2 - 4 \cdot 1 + 4 = 1$$

and thus $x \in A$. If $x = 3$, then

$$x^2 - 4x + 4 = 3^2 - 4 \cdot 3 + 4 = 1$$

and again $x \in A$. Therefore $B \subseteq A$. We conclude that $A = B$.

25. Since $1 \in A$, but $1 \notin B$, $A \neq B$.

28. Note that $A = B \cap C = \{2, 4\}$. Since $1 \in B$, but $1 \notin A$, $A \neq B$.

29. Equal

32. Not equal

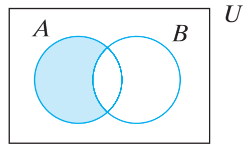
33. Let $x \in A$. Then $x = 1$ or $x = 2$. In either case, $x \in B$. Therefore $A \subseteq B$.

36. First note that $B = \mathbf{Z}^+$. Now let $x \in A$. Then $x = 2n$ for some $n \in \mathbf{Z}^+$. Since $2 \in \mathbf{Z}^+$, $2n \in \mathbf{Z}^+ = B$. Therefore $A \subseteq B$.

37. Since $3 \in A$, but $3 \notin B$, A is not a subset of B.

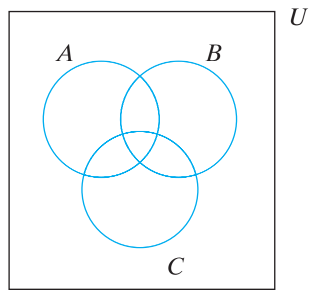
40. Since $3 \in A$, but $3 \notin B$, A is not a subset of B.

41.



44. Same as Exercise 41

47.



49. The shaded area represents the beverage, which has great taste *and* is less filling.

50. 10

53. 64

55. 4

57. $\{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c)\}$

60. $\{(a, a), (a, b), (a, c), (b, a), (b, b), (b, c), (c, a), (c, b), (c, c)\}$

61. $\{(1, a, \alpha), (1, a, \beta), (2, a, \alpha), (2, a, \beta)\}$

64. $\{(a, 1, a, \alpha), (a, 2, a, \alpha), (a, 1, a, \beta), (a, 2, a, \beta)\}$

65. The entire xy-plane

68. Parallel horizontal lines spaced one unit apart. There is a lowest line [passing through $(0, 0)$] but the lines continue indefinitely above the lowest line.

71. Parallel planes stacked one above another one unit apart. The planes continue indefinitely in both directions [above and below the origin $(0, 0, 0)$].

73. $\{\{1\}\}$

76. $\{\{a, b, c, d\}\}$, $\{\{a, b, c\}, \{d\}\}$,
$\{\{a, b, d\}, \{c\}\}$, $\{\{a, c, d\}, \{b\}\}$, $\{\{b, c, d\}, \{a\}\}$,
$\{\{a, b\}, \{c\}, \{d\}\}$, $\{\{a, c\}, \{b\}, \{d\}\}$,
$\{\{a, d\}, \{b\}, \{c\}\}$,
$\{\{b, c\}, \{a\}, \{d\}\}$, $\{\{b, d\}, \{a\}, \{c\}\}$, $\{\{c, d\}, \{a\}, \{b\}\}$,
$\{\{a, b\}, \{c, d\}\}$, $\{\{a, c\}, \{b, d\}\}$, $\{\{a, d\}, \{b, c\}\}$,
$\{\{a\}, \{b\}, \{c\}, \{d\}\}$

77. True

80. True

83. $\varnothing$, $\{a\}$, $\{b\}$, $\{a, b\}$. All but $\{a, b\}$ are proper subsets.

86. $2^n - 1$

87. $A \subseteq B$

90. $B \subseteq A$

91. $\{1, 4, 5\}$

94. The center of the circle

Section 1.2 Review

1. A proposition is a sentence that is either true or false, but not both.

2. The truth table of a proposition P made up of the individual propositions $p_1, \ldots, p_n$ lists all possible combinations of truth values for $p_1, \ldots, p_n$, T denoting true and F denoting false, and for each such combination lists the truth value of P.

3. The conjunction of propositions p and q is the proposition p *and* q. It is denoted $p \wedge q$.

4.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

5. The disjunction of propositions p and q is the proposition p *or* q. It is denoted $p \vee q$.

6.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

7. The negation of proposition p is the proposition *not* p. It is denoted $\neg p$.

8.

p	$\neg p$
T	F
F	T

Section 1.2

1. Is a proposition. Negation: $2 + 5 \neq 19$

4. Not a proposition; it is a question.

7. Not a proposition; it is a command.

10. Not a proposition; it is a description of a mathematical expression (i.e., $p - q$, where p and q are primes).

12. Ten heads were not obtained. (Alternative: At least one tail was obtained.)

15. No heads were obtained. (Alternative: Ten tails were obtained.)

16. True

19. True

22.

p	q	$p \wedge \neg q$
T	T	F
T	F	T
F	T	F
F	F	F

25.

p	q	$(p \wedge q) \wedge \neg q$
T	T	F
T	F	F
F	T	F
F	F	F

28.

p	q	$(p \vee q) \wedge (\neg p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee \neg q)$
T	T	F
T	F	F
F	T	F
F	F	F

30. $p \wedge q$; false

33. Lee does not take computer science.

36. Lee takes computer science or Lee does not take mathematics.

39. You play football and you miss the midterm exam.

42. It is not the case that you play football or you miss the midterm exam, or you pass the course.

44. Today is Monday or it is raining.

47. (Today is Monday and it is raining) and it is not the case that (it is hot or today is Monday).

49. $\neg p$

52. $\neg p \wedge \neg q$

55. $p \wedge \neg q$

58. $\neg p \wedge \neg r \wedge \neg q$

60. $p \wedge r$

63. $(p \vee q) \wedge \neg r$

67. Inclusive-or: To enter Utopia, you must show a driver's license or a passport or both. Exclusive-or: To enter Utopia, you must show a driver's license or a passport but not both. Exclusive-or is the intended meaning.

70. Inclusive-or: The car comes with a cupholder that heats or cools your drink or both. Exclusive-or: The car comes with a cupholder that heats or cools your drink but not both. Exclusive-or is the intended meaning.

73. Inclusive-or: The meeting will be canceled if fewer than 10 persons sign up or at least 3 inches of snow falls or both. Exclusive-or: The meeting will be canceled if fewer than 10 persons sign up or at least 3 inches of snow falls but not both. Inclusive-or is the intended meaning.

74. No, assuming the interpretation: It shall be unlawful for any person to keep more than three [3] dogs and more than three [3] cats upon his property within the city. A judge ruled that the ordinance was "vague." Presumably, the intended meaning was: "It shall be unlawful for any person to keep more than three [3] dogs *or* more than three [3] cats upon his property within the city."

75. `"national park" "north dakota" OR "south dakota"`

Section 1.3 Review

1. If p and q are propositions, the conditional proposition is the proposition if p then q. It is denoted $p \rightarrow q$.

2.

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

3. In the conditional proposition $p \to q$, p is the hypothesis.

4. In the conditional proposition $p \to q$, q is the conclusion.

5. In the conditional proposition $p \to q$, q is a necessary condition.

6. In the conditional proposition $p \to q$, p is a sufficient condition.

7. The converse of $p \to q$ is $q \to p$.

8. If p and q are propositions, the biconditional proposition is the proposition p if and only if q. It is denoted $p \leftrightarrow q$.

9.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

10. If the propositions P and Q are made up of the propositions $p_1, \ldots, p_n$, P and Q are logically equivalent provided that given any truth values of $p_1, \ldots, p_n$, either P and Q are both true or P and Q are both false.

11. $\neg(p \vee q) \equiv \neg p \wedge \neg q$, $\neg(p \wedge q) \equiv \neg p \vee \neg q$

12. The contrapositive of $p \to q$ is $\neg q \to \neg p$.

Section 1.3

1. If Joey studies hard, then he will pass the discrete mathematics exam.

4. If Katrina passes discrete mathematics, then she will take the algorithms course.

7. If you inspect the aircraft, then you have the proper security clearance.

10. If the program is readable, then it is well structured.

11. (For Exercise 1) If Joey passes the discrete mathematics exam, then he studied hard.

13. True **16.** False **19.** False

21. True **24.** True **27.** True

30. True **31.** True **34.** False

37. True **40.** $p \to q$ **43.** $q \leftrightarrow (p \wedge \neg r)$

44. $p \to q$

47. $q \leftrightarrow (p \wedge r)$

50. If today is Monday, then it is raining.

53. It is not the case that today is Monday or it is raining if and only if it is hot.

56. Let p: $4 < 6$ and q: $9 > 12$.
Given statement: $p \to q$; false.
Converse: $q \to p$; if $9 > 12$, then $4 < 6$; true.
Contrapositive: $\neg q \to \neg p$; if $9 \leq 12$, then $4 \geq 6$; false.

59. Let p: $|4| < 3$ and q: $-3 < 4 < 3$.
Given statement: $q \to p$; true.
Converse: $p \to q$; if $|4| < 3$, then $-3 < 4 < 3$; true.
Contrapositive:
$\neg p \to \neg q$, if $|4| \geq 3$, then $-3 \geq 4$ or $4 \geq 3$; true.

60. $P \not\equiv Q$ **63.** $P \not\equiv Q$

66. $P \not\equiv Q$ **69.** $P \not\equiv Q$

70. Pat will not use the treadmill and will not lift weights.

73. To make chili, you do not need red pepper or you do not need onions.

74.

p	q	p *imp1* q	q *imp1* p
T	T	T	T
T	F	F	F
F	T	F	F
F	F	T	T

Since p *imp1* q is true precisely when q *imp1* p is true, p *imp1* $q \equiv q$ *imp1* p.

77.

p	q	$p \to q$	$\neg p \vee q$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

Since $p \to q$ is true precisely when $\neg p \vee q$ is true, $p \to q \equiv \neg p \vee q$.

Section 1.4 Review

1. Deductive reasoning refers to the process of drawing a conclusion from a sequence of propositions.

2. In the argument $p_1, p_2, \ldots, p_n / \therefore q$, the hypotheses are $p_1, p_2, \ldots, p_n$.

3. "Premise" in another name for hypothesis.

4. In the argument $p_1, p_2, \ldots, p_n / \therefore q$, the conclusion is q.

5. The argument $p_1, p_2, \ldots, p_n / \therefore q$ is valid provided that if p_1 and p_2 and ... and p_n are all true, then q must also be true.

6. An invalid argument is an argument that is not valid.

7. $$\begin{array}{l} p \to q \\ \underline{p} \\ \therefore q \end{array}$$

8. $$\begin{array}{l} p \to q \\ \underline{\neg q} \\ \therefore \neg p \end{array}$$

9. $$\begin{array}{l} \underline{p} \\ \therefore p \vee q \end{array}$$

10. $$\begin{array}{l} \underline{p \wedge q} \\ \therefore p \end{array}$$

11. $$\begin{array}{l} p \\ \underline{q} \\ \therefore p \wedge q \end{array}$$

12. $$\begin{array}{l} p \to q \\ \underline{q \to r} \\ \therefore p \to r \end{array}$$

13. $$\begin{array}{l} p \lor q \\ \underline{\neg p} \\ \therefore q \end{array}$$

Section 1.4

1. Valid $$\begin{array}{l} p \to q \\ \underline{p} \\ \therefore q \end{array}$$

4. Invalid $$\begin{array}{l} (p \lor r) \to q \\ \underline{q} \\ \therefore \neg p \to r \end{array}$$

6. Valid. If 4 megabytes is better than no memory at all, then we will buy a new computer. If 4 megabytes is better than no memory at all, then we will buy more memory. Therefore, if 4 megabytes is better than no memory at all, then we will buy a new computer and we will buy more memory.

9. Invalid. If we will not buy a new computer, then 4 megabytes is not better than no memory at all. We will buy a new computer. Therefore, 4 megabytes is better than no memory at all.

11. Invalid **14.** Invalid

17. An analysis of the argument must take into account the fact that "nothing" is being used in two very different ways.

18. Addition

21. Let p denote the proposition "there is gas in the car," let q denote the proposition "I go to the store," and let r denote the proposition "I get a soda." Then the hypotheses are as follows:

$$p \to q$$
$$q \to r$$
$$p$$

From $p \to q$ and $q \to r$, we may use the hypothetical syllogism to conclude $p \to r$. From $p \to r$ and p, we may use modus ponens to conclude r. Since r represents the proposition "I get a soda," we conclude that the conclusion does follow from the hypotheses.

24. We construct a truth table for all the propositions involved:

p	q	$p \to q$	$\neg q$	$\neg p$
T	T	T	F	F
T	F	F	T	F
F	T	T	F	T
F	F	T	T	T

We observe that whenever the hypotheses $p \to q$ and $\neg q$ are true, the conclusion $\neg p$ is also true; therefore, the argument is valid.

27. We construct a truth table for all the propositions involved:

p	q	$p \land q$
T	T	T
T	F	F
F	T	F
F	F	F

We observe that whenever the hypotheses p and q are true, the conclusion $p \land q$ is also true; therefore, the argument is valid.

Section 1.5 Review

1. If $P(x)$ is a statement involving the variable x, we call P a propositional function if for each x in the domain of discourse, $P(x)$ is a proposition.

2. A domain of discourse for a propositional function P is a set D such that $P(x)$ is defined for every x in D.

3. A universally quantified statement is a statement of the form for all x in the domain of discourse, $P(x)$.

4. A counterexample to the statement $\forall x\, P(x)$ is a value of x for which $P(x)$ is false.

5. An existentially quantified statement is a statement of the form for some x in the domain of discourse, $P(x)$.

6. $\neg(\forall x\, P(x))$ and $\exists x\, \neg P(x)$ have the same truth values. $\neg(\exists x\, P(x))$ and $\forall x\, \neg P(x)$ have the same truth values.

7. To prove that the universally quantified statement $\forall x\, P(x)$ is true, show that for every x in the domain of discourse, the proposition $P(x)$ is true.

8. To prove that the existentially quantified statement $\exists x\, P(x)$ is true, find one value of x in the domain of discourse for which the proposition $P(x)$ is true.

9. To prove that the universally quantified statement $\forall x\, P(x)$ is false, find one value of x in the domain of discourse for which the proposition $P(x)$ is false.

10. To prove that the existentially quantified statement $\exists x\, P(x)$ is false, show that for every x in the domain of discourse, the proposition $P(x)$ is false.

11. $$\begin{array}{l} \underline{\forall x\, P(x)} \\ \therefore P(d) \text{ if } d \in D \end{array}$$

12. $$\begin{array}{l} \underline{P(d) \text{ for every } d \in D} \\ \therefore \forall x\ P(x) \end{array}$$

13. $$\begin{array}{l} \underline{\exists x\, P(x)} \\ \therefore P(d) \text{ for some } d \in D \end{array}$$

14. $$\begin{array}{l} \underline{P(d) \text{ for some } d \in D} \\ \therefore \exists x\ P(x) \end{array}$$

Section 1.5

1. Is a propositional function. The domain of discourse could be taken to be all integers.

4. Is a propositional function. The domain of discourse is the set of all movies.

7. 11 divides 77. True.

10. For every positive integer n, n divides 77. False.

12. True **15.** False **18.** False

21. $P(1) \land P(2) \land P(3) \land P(4)$

24. $P(1) \lor P(2) \lor P(3) \lor P(4)$

27. $P(2) \land P(3) \land P(4)$

28. Every student is taking a math course.

31. Some student is not taking a math course.

34. (For Exercise 28) $\exists x\, \neg P(x)$. Some student is not taking a math course.

35. Every professional athlete plays soccer. False.

38. Either someone does not play soccer or some soccer player is a professional athlete. True.

41. Everyone is a professional athlete and plays soccer. False.

43. (For Exercise 35) $\exists x(P(x) \wedge \neg Q(x))$. Someone is a professional athlete and does not play soccer.

44. $\forall x(P(x) \rightarrow Q(x))$

47. $\exists x(P(x) \wedge Q(x))$

48. (For Exercise 44) $\exists x(P(x) \wedge \neg Q(x))$. Some accountant does not own a Porsche.

49. False. A counterexample is $x = 0$.

52. True. The value $x = 2$ makes $(x > 1) \rightarrow (x^2 > x)$ true.

55. (For Exercise 49) $\exists x(x^2 \leq x)$. There exists x such that $x^2 \leq x$.

57. The literal meaning is: No man cheats on his wife. The intended meaning is: Some man does not cheat on his wife. Let $P(x)$ denote the statement "x is a man," and $Q(x)$ denote the statement "x cheats on his wife." Symbolically, the clarified statement is $\exists x(P(x) \wedge \neg Q(x))$.

60. The literal meaning is: No environmental problem is a tragedy. The intended meaning is: Some environmental problem is not a tragedy. Let $P(x)$ denote the statement "x is an environmental problem," and $Q(x)$ denote the statement "x is a tragedy." Symbolically, the clarified statement is $\exists x(P(x) \wedge \neg Q(x))$.

63. The literal meaning is: Everything is not sweetness and light. The intended meaning is: Not everything is sweetness and light. Let $P(x)$ denote the statement "x is sweetness and light." Symbolically, the clarified statement is $\exists x \neg P(x)$.

66. The literal meaning is: No circumstance is right for a formal investigation. The intended meaning is: Some circumstance is not right for a formal investigation. Let $P(x)$ denote the statement "x is a circumstance," and $Q(x)$ denote the statement "x is right for a formal investigation." Symbolically, the clarified statement is $\exists x(P(x) \wedge \neg Q(x))$.

67. (a)

p	q	$p \rightarrow q$	$q \rightarrow p$
T	T	T	T
T	F	F	T
F	T	T	F
F	F	T	T

One of $p \rightarrow q$ or $q \rightarrow p$ is true since in each row, one of the last two entries is true.

(b) The statement, "All integers are positive or all positive numbers are integers," which is false, in symbols is

$$(\forall x(I(x) \rightarrow P(x))) \vee (\forall x(P(x) \rightarrow I(x))).$$

This is *not* the same as the given statement

$$\forall x((I(x) \rightarrow P(x)) \vee (P(x) \rightarrow I(x))),$$

which is true. The ambiguity results from attempting to distribute $\forall$ across the *or*.

70. Universal instantiation

71. Let $P(x)$ denote the propositional function "x has a graphing calculator," and let $Q(x)$ denote the propositional function "x understands the trigonometric functions." The hypotheses are $\forall x\ P(x)$ and $\forall x(P(x) \rightarrow Q(x))$. By universal instantiation, we have $P(\text{Ralphie})$ and $P(\text{Ralphie}) \rightarrow Q(\text{Ralphie})$. The modus ponens rule of inference now gives $Q(\text{Ralphie})$, which represents the proposition "Ralphie understands the trigonometric functions." We conclude that the conclusion does follow from the hypotheses.

74. By definition, the proposition $\forall x\ P(x)$ is true when $P(x)$ is true for all x in the domain of discourse. We are given that $P(d)$ is true for any d in the domain of discourse D. Therefore, $\forall x\ P(x)$ is true.

Section 1.6 Review

1. For every x and for every y, $P(x, y)$. Let the domain of discourse be $X \times Y$. The statement is true if, for every $x \in X$ and for every $y \in Y$, $P(x, y)$ is true. The statement is false if there is at least one $x \in X$ and at least one $y \in Y$ such that $P(x, y)$ is false.

2. For every x, there exists y such that $P(x, y)$. Let the domain of discourse be $X \times Y$. The statement is true if, for every $x \in X$, there is at least one $y \in Y$ for which $P(x, y)$ is true. The statement is false if there is at least one $x \in X$ such that $P(x, y)$ is false for every $y \in Y$.

3. There exists x such that for every y, $P(x, y)$. Let the domain of discourse be $X \times Y$. The statement is true if there is at least one $x \in X$ such that $P(x, y)$ is true for every $y \in Y$. The statement is false if, for every $x \in X$, there is at least one $y \in Y$ such that $P(x, y)$ is false.

4. There exists x and there exists y such that $P(x, y)$. Let the domain of discourse be $X \times Y$. The statement is true if there is at least one $x \in X$ and at least one $y \in Y$ such that $P(x, y)$ is true. The statement is false if, for every $x \in X$ and for every $y \in Y$, $P(x, y)$ is false.

5. Let $P(x, y)$ be the propositional function "$x \leq y$" with domain of discourse $\mathbf{Z} \times \mathbf{Z}$. Then $\forall x \exists y P(x, y)$ is true since, for every integer x, there exists an integer y (e.g., $y = x$) such that $x \leq y$ is true. On the other hand, $\exists x \forall y P(x, y)$ is false. For every integer x, there exists an integer y (e.g., $y = x - 1$) such that $x \leq y$ is false.

6. $\exists x \exists y \neg P(x, y)$

7. $\exists x \forall y \neg P(x, y)$

8. $\forall x \exists y \neg P(x, y)$

9. $\forall x \forall y \neg P(x, y)$

10. Given a quantified propositional function, you and your opponent, whom we call Farley, play a logic game. Your goal is to try to make the propositional function true, and Farley's goal is to try to make it false. The game begins with the first (left) quantifier. If the quantifier is $\forall$, Farley chooses a value for that variable; if the quantifier is $\exists$, you choose a value for that variable. The game continues with the second quantifier. After values are chosen for all the variables, if the propositional function is true, you win; if it is false, Farley wins. If you can always win regardless of how Farley chooses values for the variables, the quantified propositional function is true, but if Farley can choose values for the variables so that you cannot win, the quantified propositional function is false.

Section 1.6

1. Everyone is taller than everyone.

4. Someone is taller than someone.

5. (For Exercise 1) In symbols: $\exists x \exists y \neg T_1(x, y)$. In words: Someone is not taller than someone.

6. (For Exercise 1) False; Garth is not taller than Garth.

9. (For Exercise 1) False; Pat is not taller than Pat.

10. Everyone is taller than or the same height as everyone.

13. Someone is taller than or the same height as someone.

14. (For Exercise 10) In symbols: $\exists x \exists y \neg T_2(x, y)$. In words: Someone is shorter than someone.

15. (For Exercise 10) False; Erin is not taller than or the same height as Garth.

18. (For Exercise 10) True

19. For any two people, if they are distinct, the first is taller than the second.

22. There are two people and, if they are distinct, the first is taller than the second.

23. (For Exercise 19) In symbols: $\exists x \exists y \neg T_3(x, y)$. In words: There are two distinct people and the first is shorter than or the same height as the second.

24. (For Exercise 19) False; Erin and Garth are distinct persons, but Erin is not taller than Garth.

27. (For Exercise 19) False; Pat and Sandy are distinct persons, but Pat is not taller than Sandy.

28. $\exists x \forall y L(x, y)$. True (think of a saint).

31. $\forall x \exists y L(x, y)$. True (according to Dean Martin's song, "Everybody Loves Somebody Sometime").

32. (For Exercise 28) Everyone does not love someone. $\forall x \exists y \neg L(x, y)$

33. $\exists y\, A(\text{Brit}, y)$

36. $\forall y \exists x\, A(x, y)$

37. False

40. True

41. (For Exercise 37) $\exists x \exists y \neg P(x, y)$ or $\exists x \exists y (x < y)$

42. False. A counterexample is $x = 2$, $y = 0$.

45. True. Take $x = y = 0$.

48. False. A counterexample is $x = y = 2$.

51. True. Take $x = 1$, $y = \sqrt{8}$.

54. True. Take $x = 0$. Then for all y, $x^2 + y^2 \geq 0$.

57. True. For any x, if we set $y = x - 1$, the conditional proposition, if $x < y$, then $x^2 < y^2$, is true because the hypothesis is false.

60. (For Exercise 42) $\exists x \exists y (x^2 \geq y + 1)$

63. (For Exercise 42) Since both quantifiers are $\forall$, Farley chooses values for both x and y. Since Farley can choose values that make $x^2 < y + 1$ false (e.g., $x = 2$, $y = 0$), Farley can win the game. Therefore, the proposition is false.

66. Since the first two quantifiers are $\forall$, Farley chooses values for both x and y. The last quantifier is $\exists$, so you choose a value for z. Farley can choose values (e.g., $x = 1$, $y = 2$) so that no matter which value you choose for z, the expression

$$(x < y) \to ((z > x) \wedge (z < y))$$

is false. Since Farley can choose values for the variables so that you cannot win, the quantified statement is false.

68. $\forall x \exists y\, P(x, y)$ must be true. Since $\forall x \forall y\, P(x, y)$ is true, regardless of which value of x is selected, $P(x, y)$ is true *for all* y. Thus for any x, $P(x, y)$ is true *for any* particular y.

71. $\forall x \forall y\, P(x, y)$ might be false. Let $P(x, y)$ denote the expression $x \leq y$. If the domain of discourse is $\mathbf{Z}^+ \times \mathbf{Z}^+$, $\exists x \forall y\, P(x, y)$ is true; however, $\forall x \forall y\, P(x, y)$ is false.

74. $\forall x \forall y\, P(x, y)$ might be false. Let $P(x, y)$ denote the expression $x \leq y$. If the domain of discourse is $\mathbf{Z}^+ \times \mathbf{Z}^+$, $\exists x \exists y\, P(x, y)$ is true; however, $\forall x \forall y\, P(x, y)$ is false.

77. $\forall x \exists y\, P(x, y)$ might be true. Let $P(x, y)$ denote the expression $x \leq y$. If the domain of discourse is $\mathbf{Z}^+ \times \mathbf{Z}^+$, $\forall x \exists y\, P(x, y)$ is true; however, $\forall x \forall y\, P(x, y)$ is false.

80. $\forall x \forall y\, P(x, y)$ must be false. Since $\forall x \exists y\, P(x, y)$ is false, there exists x, say $x = x'$, such that for all y, $P(x, y)$ is false. Choose $y = y'$ in the domain of discourse. Then $P(x', y')$ is false. Therefore $\forall x \forall y\, P(x, y)$ is false.

83. $\forall x \forall y\, P(x, y)$ must be false. Since $\exists x \forall y\, P(x, y)$ is false, for every x there exists y such that $P(x, y)$ is false. Choose $x = x'$ in the domain of discourse. For this choice of x, there exists $y = y'$ such that $P(x', y')$ is false. Therefore $\forall x \forall y\, P(x, y)$ is false.

86. $\forall x \forall y\, P(x, y)$ must be false. Since $\exists x \exists y\, P(x, y)$ is false, for every x and for every y, $P(x, y)$ is false. Choose $x = x'$ and $y = y'$ in the domain of discourse. For these choices of x and y, $P(x', y')$ is false. Therefore $\forall x \forall y\, P(x, y)$ is false.

89. $\exists x \neg(\forall y\, P(x, y))$ is not logically equivalent to $\neg(\forall x \exists y\, P(x, y))$. Let $P(x, y)$ denote the expression $x < y$. If the domain of discourse is $\mathbf{Z} \times \mathbf{Z}$, $\exists x \neg(\forall y\, P(x, y))$ is true; however, $\neg(\forall x \exists y\, P(x, y))$ is false.

92. $\exists x \exists y \neg P(x, y)$ is not logically equivalent to $\neg(\forall x \exists y\, P(x, y))$. Let $P(x, y)$ denote the expression $x < y$. If the domain of discourse is $\mathbf{Z} \times \mathbf{Z}$, $\exists x \exists y \neg P(x, y)$ is true; however, $\neg(\forall x \exists y\, P(x, y))$ is false.

93. $\forall \varepsilon > 0\, \exists \delta > 0\, \forall x\, ((0 < |x - a| < \delta) \to (|f(x) - L| < \varepsilon))$

Chapter 1 Self-Test

1. $\varnothing$

2. $A \subseteq B$

3. Yes

4. Since $|A| = 3$ and $|\mathcal{P}(A)| = 2^3 = 8$, $|\mathcal{P}(A) \times A| = 8 \cdot 3 = 24$.

5. False

6.

p	q	r	$\neg(p \wedge q) \vee (p \vee \neg r)$
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	T
F	T	T	T
F	T	F	T
F	F	T	T
F	F	F	T

7. I take hotel management and either I do not take recreation supervision or I take popular culture.

8. $p \vee (q \wedge \neg r)$

9. If Leah gets an A in discrete mathematics, then Leah studies hard.

10. Converse: If Leah studies hard, then Leah gets an A in discrete mathematics. Contrapositive: If Leah does not study hard, then Leah does not get an A in discrete mathematics.

11. True

12. $(\neg r \vee q) \rightarrow \neg q$

13. Hypothetical syllogism

14. Let

p: The Skyscrapers win.
q: I'll eat my hat.
r: I'll be quite full.

Then the argument symbolically is

$$\begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \hline \therefore r \rightarrow p \end{array}$$

The argument is invalid. If p and q are false and r is true, the hypotheses are true, but the conclusion is false.

15. The argument is invalid. If p and r are true and q is false, the hypotheses are true, but the conclusion is false.

16. Let

p: The Council approves the funds.
q: New Atlantic gets the Olympic Games.
r: New Atlantic builds a new stadium.
s: The Olympic Games are canceled.

Then the argument symbolically is

$$\begin{array}{l} p \rightarrow q \\ q \rightarrow r \\ \neg r \\ \hline \therefore \neg p \vee s \end{array}$$

From $p \rightarrow q$ and $q \rightarrow r$, we may use the hypothetical syllogism to conclude $p \rightarrow r$. From $p \rightarrow r$ and $\neg r$, we may use the modus tollens to conclude $\neg p$. We may then use addition to conclude $\neg p \vee s$.

17. The statement is not a proposition. The truth value cannot be determined without knowing what "the team" refers to.

18. The statement is a propositional function. When we substitute a particular team for the variable "team," the statement becomes a proposition.

19. For all positive integers n, n and $n+2$ are prime. The proposition is false. A counterexample is $n = 7$.

20. For some positive integer n, n and $n+2$ are prime. The proposition is true. For example, if $n = 5$, n and $n+2$ are prime.

21. $\exists x \forall y \neg K(x, y)$

22. $\forall x \exists y K(x, y)$; everybody knows somebody.

23. The statement is true. For every x, there exists y, namely the *cube root* of x, such that $x = y^3$. In words: Every real number has a cube root.

24.
$$\begin{aligned} \neg(\forall x \exists y \forall z P(x, y, z)) &\equiv \exists x \neg(\exists y \forall z P(x, y, z)) \\ &\equiv \exists x \forall y \neg(\forall z P(x, y, z)) \\ &\equiv \exists x \forall y \exists z \neg P(x, y, z) \end{aligned}$$

Section 2.1 Review

1. A mathematical system consists of axioms, definitions, and undefined terms.
2. An axiom is a proposition that is assumed to be true.
3. A definition creates a new concept in terms of existing ones.
4. An undefined term is a term that is not explicitly defined but rather is implicitly defined by the axioms.
5. A theorem is a proposition that has been proved to be true.
6. A proof is an argument that establishes the truth of a theorem.
7. A lemma is a theorem that is usually not too interesting in its own right but is useful in proving another theorem.
8. A direct proof assumes that the hypotheses are true and then, using the hypotheses as well as other axioms, definitions, and previously derived theorems, shows directly that the conclusion is true.
9. An integer n is even if there exists an integer k such that $n = 2k$.
10. An integer n is odd if there exists an integer k such that $n = 2k + 1$.
11. Within a proof, a proof of an auxiliary result is called a subproof.
12. To disprove the universally quantified statement $\forall x P(x)$, we need to find one member x in the domain of discourse that makes $P(x)$ false.

Section 2.1

1. If three points are not collinear, then there is exactly one plane that contains them.

4. If x is a nonnegative real number and n is a positive integer, $x^{1/n}$ is the nonnegative number y satisfying $y^n = x$.

7. Let m and n be even integers. Then there exist k_1 and k_2 such that $m = 2k_1$ and $n = 2k_2$. Now

$$m + n = 2k_1 + 2k_2 = 2(k_1 + k_2).$$

Therefore $m + n$ is even.

10. Let m and n be odd integers. Then there exist k_1 and k_2 such that $m = 2k_1 + 1$ and $n = 2k_2 + 1$. Now

$$mn = (2k_1 + 1)(2k_2 + 1) = 4k_1k_2 + 2k_1 + 2k_2 + 1 = 2(2k_1k_2 + k_1 + k_2) + 1.$$

Therefore mn is odd.

13. Let x and y be rational numbers. Then there exist integers m_1, n_1, m_2, n_2 such that $x = m_1/n_1$ and $y = m_2/n_2$. Now

$$x + y = \frac{m_1}{n_1} + \frac{m_2}{n_2} = \frac{m_1n_2 + m_2n_1}{n_1n_2}.$$

Since $m_1n_2 + m_2n_1$ and n_1n_2 are integers, $x + y$ is a rational number.

16. From the definition of max, it follows that $d \ge d_1$ and $d \ge d_2$. From $x \ge d$ and $d \ge d_1$, we may derive $x \ge d_1$ from a previous theorem (the second theorem of Example 2.1.5). From $x \ge d$ and $d \ge d_2$, we may derive $x \ge d_2$ from the same previous theorem. Therefore, $x \ge d_1$ and $x \ge d_2$.

19. Let $x \in X \cap Y$. From the definition of "intersection," we conclude that $x \in X$. Therefore $X \cap Y \subseteq X$.

22. Let $x \in X \cap Z$. From the definition of "intersection," we conclude that $x \in X$ and $x \in Z$. Since $X \subseteq Y$ and $x \in X$, $x \in Y$. Since $x \in Y$ and $x \in Z$, from the definition of "intersection," we conclude that $x \in Y \cap Z$. Therefore $X \cap Z \subseteq Y \cap Z$.

25. Let $x \in Y$. From the definition of "union," we conclude that $x \in X \cup Y$. Since $X \cup Y = X \cup Z$, $x \in X \cup Z$. From the definition of "union," we conclude that $x \in X$ or $x \in Z$. If $x \in Z$, we conclude that $Y \subseteq Z$. If $x \in X$, from the definition of "intersection," we conclude that $x \in X \cap Y$. Since $X \cap Y = X \cap Z$, $x \in X \cap Z$. Therefore $x \in Z$, and again $Y \subseteq Z$.

The argument that $Z \subseteq Y$ is the same as that for $Y \subseteq Z$ with the roles of Y and Z reversed. Thus $Y = Z$.

28. Since $X \in \mathcal{P}(X)$, $X \in \mathcal{P}(Y)$. Therefore $X \subseteq Y$.

31. False. If $X = \{1, 2\}$ and $Y = \{2, 3\}$, then X is not a subset of Y since $1 \in X$, but $1 \notin Y$. Also, Y is not a subset of X since $3 \in Y$, but $3 \notin X$.

34. False. Let $X = \{1, a\}$, $Y = \{1, 2, 3\}$, and $Z = \{3\}$. Then $Y - Z = \{1, 2\}$ and $(X \cup Y) - (X \cup Z) = \{2\}$.

37. False. Let $X = Y = \{1\}$ and $U = \{1, 2\}$. Then $\overline{X \cap Y} = \{2\}$, which is not a subset of X.

40. False. Let $X = Y = \{1\}$ and $U = \{1, 2\}$. Then $\overline{X \times Y} = \{(1, 2), (2, 1), (2, 2)\}$ and $\overline{X} \times \overline{Y} = \{(2, 2)\}$.

43. False. Let $X = \{1, 2\}$, $Y = \{1\}$, and $Z = \{2\}$. Then $X \cap (Y \times Z) = \varnothing$ and $(X \cap Y) \times (X \cap Z) = \{(1, 2)\}$.

44. We prove only $(A \cup B) \cup C = A \cup (B \cup C)$. Let $x \in (A \cup B) \cup C$. Then $x \in A \cup B$ or $x \in C$. If $x \in A \cup B$, then $x \in A$ or $x \in B$. Thus $x \in A$ or $x \in B$ or $x \in C$. If $x \in A$, then $x \in A \cup (B \cup C)$. If $x \in B$ or $x \in C$, then $x \in B \cup C$. Again, $x \in A \cup (B \cup C)$.

Now suppose that $x \in A \cup (B \cup C)$. Then $x \in A$ or $x \in B \cup C$. If $x \in B \cup C$, then $x \in B$ or $x \in C$. Thus $x \in A$ or $x \in B$ or $x \in C$. If $x \in A$ or $x \in B$, then $x \in A \cup B$. Thus $x \in (A \cup B) \cup C$. If $x \in C$, again $x \in (A \cup B) \cup C$. Therefore $(A \cup B) \cup C = A \cup (B \cup C)$.

47. We prove only $A \cup \varnothing = A$. Let $x \in A \cup \varnothing$. Then $x \in A$ or $x \in \varnothing$. But $x \notin \varnothing$, so $x \in A$.

Now suppose that $x \in A$. Then $x \in A \cup \varnothing$. Therefore $A \cup \varnothing = A$.

50. We prove only $A \cup U = U$. By definition, any set is a subset of the universal set, so $A \cup U \subseteq U$.

If $x \in U$, then $x \in A \cup U$. Thus $U \subseteq A \cup U$. Therefore $A \cup U = U$.

53. We prove only $\overline{\varnothing} = U$. By definition, any set is a subset of the universal set, so $\overline{\varnothing} \subseteq U$.

Now suppose that $x \in U$. Then $x \notin \varnothing$ (by the definition of "empty set"). Thus $x \in \overline{\varnothing}$ and $U \subseteq \overline{\varnothing}$. Therefore $\overline{\varnothing} = U$.

55. Let $x \in A \triangle B$. Then $x \in A \cup B$ and $x \notin A \cap B$. Since $x \in A \cup B$, $x \in A$ or $x \in B$. Since $x \notin A \cap B$, $x \notin A$ or $x \notin B$. If $x \in A$, then $x \notin B$. Thus $x \in A - B$; hence $x \in (A - B) \cup (B - A)$. If $x \in B$, then $x \notin A$. Thus $x \in B - A$ and again $x \in (A - B) \cup (B - A)$. Therefore $A \triangle B \subseteq (A - B) \cup (B - A)$.

Now suppose that $x \in (A - B) \cup (B - A)$. Then $x \in A - B$ or $x \in B - A$. If $x \in A - B$, then $x \in A$ and $x \notin B$. Thus $x \in A \cup B$ and $x \notin A \cap B$. Therefore $x \in (A \cup B) - (A \cap B) = A \triangle B$. If $x \in B - A$, then $x \in B$ and $x \notin A$. Then $x \in A \cup B$ and $x \notin A \cap B$. Again $x \in (A \cup B) - (A \cap B) = A \triangle B$. Therefore $(A - B) \cup (B - A) \subseteq A \triangle B$. We have proved that $(A - B) \cup (B - A) = A \triangle B$.

58. False. Let $A = \{1, 2, 3\}$, $B = \{2, 3, 4\}$, and $C = \{1, 2, 4\}$. Then $A \triangle (B \cup C) = \{4\}$ and $(A \triangle B) \cup (A \triangle C) = \{1, 4\} \cup \{3, 4\} = \{1, 3, 4\}$.

61. True. Using Example 2.1.11, we find that

$$A \cap (B \triangle C) = A \cap [(B \cup C) - (B \cap C)] = [A \cap (B \cup C)] - [A \cap (B \cap C)].$$

Using the distributive law and observing that $(A \cap B) \cap (A \cap C) = A \cap (B \cap C)$, we find that

$$(A \cap B) \triangle (A \cap C) = [(A \cap B) \cup (A \cap C)] - [(A \cap B) \cap (A \cap C)] = [A \cap (B \cup C)] - [A \cap (B \cap C)].$$

Therefore

$$A \cap (B \triangle C) = (A \cap B) \triangle (A \cap C).$$

Section 2.2 Review

1. A proof by contradiction assumes that the hypotheses are true and that the conclusion is false and then, using the hypotheses and the negated conclusion as well as other axioms, definitions, and previously derived theorems, derives a contradiction.

2. Example 2.2.1

3. "Indirect proof" is another name for proof by contradiction.

4. To prove $p \to q$, proof by contrapositive proves the equivalent statement $\neg q \to \neg p$.

5. Example 2.2.4

6. Instead of proving

$$(p_1 \vee p_2 \vee \cdots \vee p_n) \to q,$$

in proof by cases, we prove

$$(p_1 \to q) \wedge (p_2 \to q) \wedge \cdots \wedge (p_n \to q).$$

7. Example 2.2.5
8. Proof of equivalence shows that two or more statements are all true or all false.
9. Example 2.2.9
10. If the statements are p, q, and r, we can show that they are equivalent by proving that $p \to q$, $q \to r$, and $r \to p$ are all true.
11. A proof of $\exists x\ P(x)$ is called an existence proof.
12. An existence proof of $\exists x\ P(x)$ that exhibits an element a of the domain of discourse that makes $P(a)$ true is called a constructive proof.
13. Example 2.2.10
14. A proof of $\exists x\ P(x)$ that does not exhibit an element a of the domain of discourse that makes $P(a)$ true, but rather proves $\exists x\ P(x)$ some other way (e.g., using proof by contradiction), is called a nonconstructive proof.
15. Example 2.2.12

Section 2.2

1. Suppose, by way of contradiction, that x is rational. Then there exist integers p and q such that $x = p/q$. Now $x^2 = p^2/q^2$ is rational, which is a contradiction.
4. Suppose, by way of contradiction, that n is even. Then there exists k such that $n = 2k$. Now $n^2 = 2(2k^2)$; thus n^2 is even, which is a contradiction.
7. Suppose, by way of contradiction, that $\sqrt[3]{2}$ is rational. Then there exist integers p and q such that $\sqrt[3]{2} = p/q$. We assume that the fraction p/q is in lowest terms so that p and q are not both even. Cubing $\sqrt[3]{2} = p/q$ gives $2 = p^3/q^3$, and multiplying by q^3 gives $2q^3 = p^3$. It follows that p^3 is even. An argument like that in Example 2.2.1 shows that p is even. Therefore, there exists an integer k such that $p = 2k$. Substituting $p = 2k$ into $2q^3 = p^3$ gives $2q^3 = (2k)^3 = 8k^3$. Canceling 2 gives $q^3 = 4k^3$. Therefore q^3 is even and, thus, q is even. Thus p and q are both even, which contradicts our assumption that p and q are not both even. Therefore, $\sqrt[3]{2}$ is irrational.
10. Since the integers increase without bound, there exists $n \in \mathbf{Z}$ such that $1/(b-a) < n$. Therefore $1/n < b - a$. Choose $m \in \mathbf{Z}$ as large as possible satisfying $m/n \le a$. Then, by the choice of m, $a < (m+1)/n$. Also

$$\frac{m+1}{n} = \frac{m}{n} + \frac{1}{n} < a + (b-a) = b.$$

Therefore $x = (m+1)/n$ is a rational number satisfying $a < x < b$.

13. True. Let $a = b = 2$. Then a and b are rational numbers and $a^b = 4$ is also rational. This is a constructive existence proof.
16. True. We give a proof by contradiction. Suppose that $(X - Y) \cap (Y - X)$ is nonempty. Then there exists $x \in (X - Y) \cap (Y - X)$. Thus $x \in X - Y$ and $x \in Y - X$. Since $x \in X - Y$, $x \in X$ and $x \notin Y$. Since $x \in Y - X$, $x \in Y$ and $x \notin X$. We now have $x \in X$ and $x \notin X$, which is a contradiction. Therefore $(X - Y) \cap (Y - X) = \varnothing$.
19. Suppose, by way of contradiction, that no two bags contain the same number of coins. Suppose that we arrange the bags in increasing order of the number of coins that they contain. Then the first bag contains at least one coin; the second bag contains at least two coins; and so on. Thus the total number of coins is at least

$$1 + 2 + 3 + \cdots + 9 = 45.$$

This contradicts the hypothesis that there are 40 coins. Thus if 40 coins are distributed among nine bags so that each bag contains at least one coin, at least two bags contain the same number of coins.

22. We use proof by contradiction and assume the negation of the conclusion

$$\neg\exists i(s_i \le A).$$

By the generalized De Morgan's laws for logic, this latter statement is equivalent to

$$\forall i(s_i > A).$$

Thus we assume

$$\begin{aligned} s_1 &> A \\ s_2 &> A \\ &\vdots \\ s_n &> A. \end{aligned}$$

Adding these inequalities yields

$$s_1 + s_2 + \cdots + s_n > nA.$$

Dividing by n gives

$$\frac{s_1 + s_2 + \cdots + s_n}{n} > A,$$

which contradicts the hypothesis. Therefore, there exists i such that $s_i \le A$.

25. Since $s_i \ne s_j$, either $s_i \ne A$ or $s_j \ne A$. By changing the notation, if necessary, we may assume that $s_i \ne A$. Either $s_i < A$ or $s_i > A$. If $s_i < A$, the proof is complete; so assume that $s_i > A$. We show that there exists k such that $s_k < A$. Suppose, by way of contradiction, that $s_m \ge A$ for all m, that is,

$$\begin{aligned} s_1 &\ge A \\ s_2 &\ge A \\ &\vdots \\ s_n &\ge A. \end{aligned}$$

Adding these inequalities yields

$$s_1 + s_2 + \cdots + s_i + \cdots + s_n > nA$$

since $s_i > A$. Dividing by n gives

$$\frac{s_1 + s_2 + \cdots + s_n}{n} > A,$$

which is a contradiction. Therefore there exists k such that $s_k < A$.

27. Notice that if $n \geq 2$ and $m \geq 1$,

$$2m + 5n^2 \geq 2m + 5 \cdot 2^2 > 20,$$

so the only possible solution is if $n = 1$. However, if $n = 1$,

$$2m + 5n^2 = 2m + 5,$$

which is odd being the sum of an even integer and an odd integer. Thus this sum cannot equal 20. Therefore, $2m + 5n^2 = 20$ has no solution in positive integers.

30. We claim that if n and $n + 1$ are consecutive integers, one is odd and one is even. Suppose that n is odd. Then there exists k such that $n = 2k + 1$. Now $n + 1 = 2k + 2 = 2(k + 1)$, which is even. If n is even, there exists k such that $n = 2k$. Now $n + 1 = 2k + 1$, which is odd. Since one of n and $n + 1$ is even and the other is odd, their product is even (see Exercise 11, Section 2.1).

32. We consider four cases: $x \geq 0, y \geq 0$; $x < 0, y \geq 0$; $x \geq 0, y < 0$; and $x < 0, y < 0$.

First assume that $x \geq 0$ and $y \geq 0$. Then $xy \geq 0$ and $|xy| = xy = |x||y|$. Next assume that $x < 0$ and $y \geq 0$. Then $xy \leq 0$ and $|xy| = -xy = (-x)(y) = |x||y|$. Next assume that $x \geq 0$ and $y < 0$. Then $xy \leq 0$ and $|xy| = -xy = (x)(-y) = |x||y|$. Finally assume that $x < 0$ and $y < 0$. Then $xy > 0$ and $|xy| = xy = (-x)(-y) = |x||y|$.

34. We consider three cases: $x > 0$, $x = 0$, and $x < 0$.
If $x > 0$, $|x| = x$ and $\operatorname{sgn}(x) = 1$. Therefore,

$$|x| = x = 1 \cdot x = \operatorname{sgn}(x)x.$$

If $x = 0$, $|x| = 0$ and $\operatorname{sgn}(x) = 0$. Therefore,

$$|x| = 0 = 0 \cdot 0 = \operatorname{sgn}(x)x.$$

If $x < 0$, $|x| = -x$ and $\operatorname{sgn}(x) = -1$. Therefore,

$$|x| = -x = -1 \cdot x = \operatorname{sgn}(x)x.$$

In every case, we have $|x| = \operatorname{sgn}(x)x$.

37. We consider two cases: $x \geq y$ and $x < y$.
If $x \geq y$,

$$\max\{x, y\} = x \qquad \text{and} \qquad \min\{x, y\} = y.$$

Therefore,

$$\max\{x, y\} + \min\{x, y\} = x + y.$$

If $x < y$,

$$\max\{x, y\} = y \qquad \text{and} \qquad \min\{x, y\} = x.$$

Therefore,

$$\max\{x, y\} + \min\{x, y\} = y + x = x + y.$$

In either case,

$$\max\{x, y\} + \min\{x, y\} = x + y.$$

41. We first prove that if n is even, then $n + 2$ is even. Assume that n is even. Then there exists k such that $n = 2k$. Now $n + 2 = 2k + 2 = 2(k + 1)$ is even.

We next prove that if $n + 2$ is even, then n is even. Assume that $n + 2$ is even. Then there exists k such that $n + 2 = 2k$. Now $n = 2k - 2 = 2(k - 1)$ is even.

43. We first prove that if $A \subseteq B$, then $\overline{B} \subseteq \overline{A}$. Assume that $A \subseteq B$. Let $x \in \overline{B}$. Then $x \notin B$. If $x \in A$, then $x \in B$, which is not the case. Therefore $x \notin A$. Therefore $x \in \overline{A}$. Thus $\overline{B} \subseteq \overline{A}$.

We next prove that if $\overline{B} \subseteq \overline{A}$, then $A \subseteq B$. Assume that $\overline{B} \subseteq \overline{A}$. From the first part of the proof, we can deduce $\overline{\overline{A}} \subseteq \overline{\overline{B}}$. Since $\overline{\overline{A}} = A$ and $\overline{\overline{B}} = B$, $A \subseteq B$.

46. We first show that if $(a, b) = (c, d)$, then $a = c$ and $b = d$. Assume that $(a, b) = (c, d)$. Then

$$\{\{a\}, \{a, b\}\} = \{\{c\}, \{c, d\}\}. \tag{$*$}$$

First suppose that $a \neq b$. Then the set on the left-hand side of equation (*) contains two distinct sets: $\{a\}$ and $\{a, b\}$. Thus the set on the right-hand side of equation (*) also contains two distinct sets: $\{c\}$ and $\{c, d\}$. Therefore $c \neq d$. (If $c = d$, $\{c, d\} = \{c, c\} = \{c\}$.) Since $a \neq b$ and $c \neq d$, we must have

$$\{a\} = \{c\} \quad \text{and} \quad \{a, b\} = \{c, d\}.$$

It follows that $a = c$ and $b = d$.

Now suppose that $a = b$. Then

$$\{\{a\}, \{a, b\}\} = \{\{a\}, \{a, a\}\} = \{\{a\}, \{a\}\} = \{\{a\}\}.$$

Thus the set on the left-hand side of equation (*) contains one set. Therefore the set on the right-hand side of equation (*) also contains one set. We must have $c = d$; otherwise the set on the right-hand side of equation (*) would contain two distinct sets. Thus

$$\{\{c\}, \{c, d\}\} = \{\{c\}\}.$$

Equation (*) now becomes

$$\{\{a\}\} = \{\{c\}\}.$$

It follows that $a = c$ and, hence, $b = d$. We have shown that if $(a, b) = (c, d)$, then $a = c$ and $b = d$.

If $a = c$ and $b = d$, then

$$(a, b) = \{\{a\}, \{a, b\}\} = \{\{c\}, \{c, d\}\} = (c, d).$$

The proof is complete.

47. [(a) $\to$ (b)] Assume that n is odd. Then there exists k' such that $n = 2k' + 1$. Since $n = 2(k' + 1) - 1$, taking $k = k' + 1$, we have $n = 2k - 1$.

[(b) $\to$ (c)] Assume that there exists k such that $n = 2k - 1$. Then

$$n^2 + 1 = (2k - 1)^2 + 1 = (4k^2 - 4k + 1) + 1 = 2(2k^2 - 2k + 1).$$

Therefore $n^2 + 1$ is even.

[(c) $\to$ (a)] We prove the contrapositive: If n is even, then $n^2 + 1$ is odd.

Suppose that n is even. Then there exists k such that $n = 2k$. Now

$$n^2 + 1 = (2k)^2 + 1 = 4k^2 + 1 = 2(2k^2) + 1.$$

Therefore $n^2 + 1$ is odd.

Section 2.3 Review

1. $p \vee q,\ \neg p \vee r / \therefore q \vee r$
2. A clause consists of terms separated by *or*'s, where each term is a variable or the negation of a variable.
3. A proof by resolution proceeds by repeatedly applying the rule in Exercise 1 to pairs of statements to derive new statements until the conclusion is derived.

Section 2.3

1.

p	q	r	$p \vee q$	$\neg p \vee r$	$q \vee r$
T	T	T	T	T	T
T	T	F	T	F	T
T	F	T	T	T	T
T	F	F	T	F	F
F	T	T	T	T	T
F	T	F	T	T	T
F	F	T	F	T	T
F	F	F	F	T	F

2. 1. $\neg p \vee q \vee r$
   2. $\neg q$
   3. $\neg r$
   4. $\neg p \vee r$ From 1 and 2
   5. $\neg p$ From 3 and 4
5. First we note that $p \to q$ is logically equivalent to $\neg p \vee q$. We now argue as follows:
   1. $\neg p \vee q$
   2. $p \vee q$
   3. q From 1 and 2
7. (For Exercise 2)
   1. $\neg p \vee q \vee r$ Hypothesis
   2. $\neg q$ Hypothesis
   3. $\neg r$ Hypothesis
   4. p Negation of conclusion
   5. $\neg p \vee r$ From 1 and 2
   6. $\neg p$ From 3 and 5

   Now 4 and 6 combine to give a contradiction.

Section 2.4 Review

1. Suppose that we have propositional function $S(n)$ whose domain of discourse is the set of positive integers. Suppose that $S(1)$ is true and, for all $n \geq 1$, if $S(n)$ is true, then $S(n+1)$ is true. Then $S(n)$ is true for every positive integer n.
2. We first verify that $S(1)$ is true (Basis Step). We then assume that $S(n)$ is true and prove that $S(n+1)$ is true (Inductive Step).
3. $\dfrac{n(n+1)}{2}$
4. The geometric sum is the sum

$$a + ar^1 + ar^2 + \cdots + ar^n.$$

It is equal to

$$\frac{a(r^{n+1} - 1)}{r - 1}.$$

Section 2.4

1. **Basis Step** $1 = 1^2$

   Inductive Step Assume true for n.

$$1 + \cdots + (2n - 1) + (2n + 1) = n^2 + 2n + 1 = (n + 1)^2$$

4. **Basis Step** $1^2 = (1 \cdot 2 \cdot 3)/6$

   Inductive Step Assume true for n.

$$\begin{aligned} 1^2 + \cdots + n^2 + (n+1)^2 &= \frac{n(n+1)(2n+1)}{6} + (n+1)^2 \\ &= \frac{(n+1)(n+2)(2n+3)}{6} \end{aligned}$$

7. **Basis Step** $1/(1 \cdot 3) = 1/3$

   Inductive Step Assume true for n.

$$\begin{aligned} &\frac{1}{1 \cdot 3} + \cdots + \frac{1}{(2n-1)(2n+1)} + \frac{1}{(2n+1)(2n+3)} \\ &= \frac{n}{2n+1} + \frac{1}{(2n+1)(2n+3)} = \frac{n+1}{2n+3} \end{aligned}$$

10. **Basis Step** $\cos x = \dfrac{\cos[(x/2) \cdot 2] \sin(x/2)}{\sin(x/2)}$

    Inductive Step Assume true for n. Then

$$\begin{aligned} &\cos x + \cdots + \cos nx + \cos(n+1)x \\ &= \frac{\cos[(x/2)(n+1)] \sin(nx/2)}{\sin(x/2)} + \cos(n+1)x. \quad (*) \end{aligned}$$

We must show that the right-hand side of $(*)$ is equal to

$$\frac{\cos[(x/2)(n+2)] \sin[(n+1)x/2]}{\sin(x/2)}.$$

This is the same as showing that [after multiplying by the term $\sin(x/2)$]

$$\begin{aligned} &\cos\left[\frac{x}{2}(n+1)\right] \sin\frac{nx}{2} + \cos(n+1)x \sin\frac{x}{2} \\ &= \cos\left[\frac{x}{2}(n+2)\right] \sin\left[\frac{(n+1)x}{2}\right]. \end{aligned}$$

If we let $\alpha = (x/2)(n+1)$ and $\beta = x/2$, we must show that

$$\cos\alpha \sin(\alpha - \beta) + \cos 2\alpha \sin\beta = \cos(\alpha + \beta) \sin\alpha.$$

This last equation can be verified by reducing each side to terms involving α and β.

12. Basis Step $1/2 \leq 1/2$

Inductive Step Assume true for n.

$$\frac{1 \cdot 3 \cdot 5 \cdots (2n-1)(2n+1)}{2 \cdot 4 \cdot 6 \cdots (2n)(2n+2)} \geq \frac{1}{2n} \cdot \frac{2n+1}{2n+2}$$
$$= \frac{2n+1}{2n} \cdot \frac{1}{2n+2} \geq \frac{1}{2n+2}$$

15. Basis Step ($n = 4$) $2^4 = 16 \geq 16 = 4^2$

Inductive Step Assume true for n.

$$(n+1)^2 = n^2 + 2n + 1 \leq 2^n + 2n + 1$$
$$\leq 2^n + 2^n \quad \text{by Exercise 14}$$
$$= 2^{n+1}$$

18. $r^0 + r^1 + \cdots r^n = \dfrac{1 - r^{n+1}}{1 - r} < \dfrac{1}{1 - r}$

21. Basis Step $7^1 - 1 = 6$ is divisible by 6.

Inductive Step Suppose that 6 divides $7^n - 1$. Now

$$7^{n+1} - 1 = 7 \cdot 7^n - 1 = 7^n - 1 + 6 \cdot 7^n.$$

Since 6 divides both $7^n - 1$ and $6 \cdot 7^n$, it divides their sum, which is $7^{n+1} - 1$.

24. Basis Step $3^1 + 7^1 - 2 = 8$ is divisible by 8.

Inductive Step Suppose that 8 divides $3^n + 7^n - 2$. Now

$$3^{n+1} + 7^{n+1} - 2 = 3(3^n + 7^n - 2) + 4(7^n + 1).$$

By the inductive assumption, 8 divides $3^n + 7^n - 2$. We can use mathematical induction to show that 2 divides $7^n + 1$ for all $n \geq 1$ (the argument is similar to that given in the hint for Exercise 21). It then follows that 8 divides $4(7^n + 1)$. Since 8 divides both $3(3^n + 7^n - 2)$ and $4(7^n + 1)$, it divides their sum, which is $3^{n+1} + 7^{n+1} - 2$.

27. We prove the assertion using induction on n. The Basis Step is $n = 1$. In this case, there is one subset of $\{1\}$ with an even number of elements, namely, $\varnothing$. Since $2^{n-1} = 2^0 = 1$, the assertion is true when $n = 1$.

Assume that the number of subsets of $\{1, \ldots, n\}$ containing an even number of elements is 2^{n-1}. We must prove that the number of subsets of $\{1, \ldots, n + 1\}$ containing an even number of elements is 2^n.

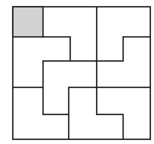
Let $E_1, \ldots, E_{2^{n-1}}$ denote the subsets of $\{1, 2, \ldots, n\}$ containing an even number of elements. Since there are 2^n subsets of $\{1, 2, \ldots, n\}$ altogether and 2^{n-1} contain an even number of elements, there are $2^n - 2^{n-1} = 2^{n-1}$ subsets of $\{1, \ldots, n\}$ that contain an odd number of elements. Denote these as $O_1, \ldots, O_{2^{n-1}}$. Now $E_1, \ldots, E_{2^{n-1}}$ are the subsets of $\{1, \ldots, n+1\}$ containing an even number of elements that do not contain $n + 1$, and

$$O_1 \cup \{n+1\}, \ldots, O_{2^{n-1}} \cup \{n+1\}$$

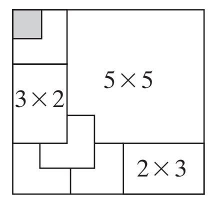
are the subsets of $\{1, \ldots, n+1\}$ containing an even number of elements that contain $n + 1$. Thus there are $2^{n-1} + 2^{n-1} = 2^n$ subsets of $\{1, \ldots, n + 1\}$ that contain an even number of elements. The Inductive Step is complete.

29. At the Inductive Step when the $(n+1)$st line is added, because of the assumptions, the line will intersect each of the other n lines. Now, imagine traveling along the $(n + 1)$st line. Each time we pass through one of the original regions, it is divided into two regions.

32.



35. We denote the square in row i, column j by (i, j). Then, by symmetry, we need only consider 7×7 boards with squares (i, j) removed where $i \leq j \leq 4$. The solution when square $(1, 1)$ is removed is shown in the following figure.



Not all trominoes of the tiling are shown. By Exercise 34, the 3×2 subboards have tilings. By Exercise 32, the 5×5 subboard with a corner square removed has a tiling. Essentially the same figure gives tilings if square (1, 2) or (2, 2) is deleted.

A similar argument gives tilings for the remaining cases.

38. Basis Step ($n = 1$) The board is a tromino.

Inductive Step Assume that any $2^n \times 2^n$ deficient board can be tiled with trominoes. We must prove that any $2^{n+1} \times 2^{n+1}$ deficient board can be tiled with trominoes.

Given a $2^{n+1} \times 2^{n+1}$ deficient board, we divide the board into four $2^n \times 2^n$ subboards as shown in Figure 2.4.6. By the inductive assumption, we can tile the subboard containing the missing square. The three remaining subboards form a $2^n \times 2^n$ L-shape, which may be tiled using Exercise 37. Thus, the $2^{n+1} \times 2^{n+1}$ deficient board is tiled. The inductive step is complete.

39. Number the squares as shown:

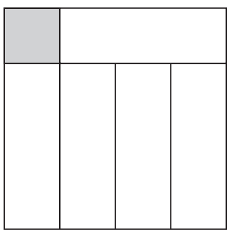
1	2	3	1
2	3	1	2
3	1	2	3
1	2	3	1

Notice that each tromino covers exactly one 1, one 2, and one 3. Therefore, if there is a tiling, five 2's are covered. Since five trominoes are required, the missing square cannot be a 2. Similarly the missing square cannot be a 3.

The same argument applied to

1	3	2	1
2	1	3	2
3	2	1	3
1	3	2	1

shows that the only possibility for the missing square is a corner. Such a board can be tiled:



42. We prove that $pow = a^{i-1}$ is a loop invariant for the while loop. Just before the while loop begins executing, $i = 1$ and $pow = 1$, so $pow = a^{1-1}$. We have proved the Basis Step.

Assume that $pow = a^{i-1}$. If $i \le n$ (so that the loop body executes again), pow becomes

$$pow * a = a^{i-1} * a = a^i,$$

and i becomes $i + 1$. We have proved the Inductive Step. Therefore $pow = a^{i-1}$ is an invariant for the while loop.

The while loop terminates when $i = n + 1$. Because $pow = a^{i-1}$ is an invariant, at this point $pow = a^n$.

46. (a) $S_1 = 0 \neq 2$;

$$\begin{aligned} 2 + \cdots + 2n + 2(n+1) &= S_n + 2n + 2 \\ &= (n+2)(n-1) + 2n + 2 \\ &= (n+3)n = S_{n+1}. \end{aligned}$$

(b) We must have $S'_n = S'_{n-1} + 2n$; thus

$$\begin{aligned} S'_n &= S'_{n-1} + 2n \\ &= [S'_{n-2} + 2(n-1)] + 2n \\ &= S'_{n-2} + 2n + 2(n-1) \\ &= S'_{n-3} + 2n + 2(n-1) + 2(n-2) \\ &\vdots \\ &= S'_1 + 2[n + (n-1) + \cdots + 2] \\ &= C' + 2\left[\frac{n(n+1)}{2} - 1\right] \\ &= n^2 + n + C. \end{aligned}$$

50. If $n = 2$, each person throws a pie at the other and there are no survivors.

53. The statement is false. In

1 and 5 are farthest apart, but neither is a survivor.

55. Let x and y be points in $X \cap Y$. Then x is in X and y is in X. Since X is convex, the line segment from x to y is in X. Similarly, the line segment from x to y is in Y. Therefore, the line segment from x to y is in $X \cap Y$. Thus, $X \cap Y$ is convex.

58. Let $x_1, \ldots, x_n$ denote the n points, and let X_i be the circle of radius 1 centered at x_i. Apply Helly's Theorem to $X_1, \ldots, X_n$.

60. 1

63. **Basis Step ($i = 1$)** Since 2 is eliminated, 1 survives. Thus $J(2) = 1$.

Inductive Step Assume true for i. Now suppose that 2^{i+1} persons are arranged in a circle. We begin by eliminating $2, 4, 6, \ldots, 2^{i+1}$. We then have 2^i persons arranged in a circle, and, beginning with 1, we eliminate the second person, then the fourth person, and so on. By the inductive assumption, 1 survives. Therefore,

$$J(2^{i+1}) = J(2^i) = 1.$$

66. The greatest power of 2 less than or equal to 100,000 is 2^{16}. Thus, in the notation of Exercise 64, $n = 100{,}000$, $i = 16$, and

$$j = n - 2^i = 100{,}000 - 2^{16} = 100{,}000 - 65{,}536 = 34{,}464.$$

By Exercise 64,

$$J(100{,}000) = J(n) = 2j + 1 = 2 \cdot 34{,}464 + 1 = 68{,}929.$$

67.

$$\begin{aligned} b_1 + b_2 + \cdots + b_n &= (a_2 - a_1) + (a_3 - a_2) \\ &\quad + \cdots + (a_{n+1} - a_n) \\ &= -a_1 + a_{n+1} = a_{n+1} - a_1 \end{aligned}$$

since $a_2, \ldots, a_n$ cancel.

70. Let

$$a_n = \frac{1}{n}.$$

Then

$$\Delta a_n = a_{n+1} - a_n = \frac{1}{n+1} - \frac{1}{n} = \frac{-1}{n(n+1)}.$$

Let $b_n = \Delta a_n$. By Exercise 67,

$$\begin{aligned} \frac{-1}{1 \cdot 2} + \cdots + \frac{-1}{n(n+1)} &= \Delta a_1 + \cdots + \Delta a_n \\ &= b_1 + \cdots + b_n \\ &= a_{n+1} - a_1 \\ &= \frac{1}{n+1} - 1 = \frac{-n}{n+1}. \end{aligned}$$

Multiplying by -1 yields the desired formula.

Section 2.5 Review

1. Suppose that we have a propositional function $S(n)$ whose domain of discourse is the set of integers greater than or equal to n_0. Suppose that $S(n_0)$ is true; and for all $n > n_0$, if $S(k)$ is true for all k, $n_0 \le k < n$, then $S(n)$ is true. Then $S(n)$ is true for every integer $n \ge n_0$.
2. Every nonempty set of nonnegative integers has a least element.
3. If d and n are integers, $d > 0$, there exist unique integers q (quotient) and r (remainder) satisfying $n = dq+r$, $0 \le r < d$.

Section 2.5

1. **Basis Steps ($n = 6, 7$)** We can make six cents postage by using three 2-cent stamps. We can make seven cents postage by using one 7-cent stamp.

   Inductive Step We assume that $n \ge 8$ and postage of k cents or more can be achieved by using only 2-cent and 7-cent stamps for $6 \le k < n$. By the inductive assumption, we can make postage of $n - 2$ cents. We may add a 2-cent stamp to make n cents postage.

3. **Basis Step ($n = 4$)** We can make four cents postage by using two 2-cent stamps.

   Inductive Step We assume that we can make n cents postage, and we prove that we can make $n + 1$ cents postage.

   If among the stamps that make n cents postage there is at least one 5-cent stamp, we replace one 5-cent stamp by three 2-cent stamps to make $n + 1$ cents postage. If there are no 5-cent stamps among the stamps that make n cents postage, there are at least two 2-cent stamps (because $n \ge 4$). We replace two 2-cent stamps by one 5-cent stamp to make $n+1$ cents postage.

6. In the Inductive Step, we must have $k = \lfloor n/2 \rfloor \ge 3$. Since this inequality fails for $n = 4, 5$, the Basis Steps are $n = 3, 4, 5$.
9. $c_2 = 4, c_3 = 9, c_4 = 20, c_5 = 29$
11. $c_2 = 2, c_3 = 3, c_4 = 12, c_5 = 13$
14. Notice that

$$\begin{aligned}
c_0 &= 0\\
c_1 &= c_0 + 3 = 3\\
c_2 &= c_1 + 3 = 6\\
c_3 &= c_1 + 3 = 6\\
c_4 &= c_2 + 3 = 9.
\end{aligned}$$

   Thus the assertion $c_n \le 2n$ fails for $n = 4$.

   In the Inductive Step, we must have $k = \lfloor n/2 \rfloor \ge 3$. Since this inequality fails for $n = 4, 5$, the Basis Steps are $n = 3, 4, 5$. In the fallacious proof, only the case $n = 3$ was proved in the Basis Steps. In fact, since the statement is false for $n = 4$, the Basis Steps $n = 3, 4, 5$ cannot be proved.

16. $q = 5,\ r = 2$
19. $q = -1,\ r = 2$
22. $\dfrac{5}{6} = \dfrac{1}{2} + \dfrac{1}{3} = \dfrac{1}{2} + \dfrac{1}{4} + \dfrac{1}{12}$
26. We may assume that $p/q > 1$. Choose the largest integer n satisfying

$$\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n} \le \frac{p}{q}.$$

   (The previous Problem-Solving Corner shows that the sum

$$\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n}$$

   is unbounded; so such an n exists.) If we obtain an equality, p/q is in Egyptian form, so suppose that

$$\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n} < \frac{p}{q}. \qquad (*)$$

   Set

$$D = \frac{p}{q} - \left(\frac{1}{1} + \cdots + \frac{1}{n}\right).$$

   Clearly, $D > 0$. Since n is the largest integer satisfying $(*)$,

$$\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n} + \frac{1}{n+1} \ge \frac{p}{q}.$$

   Thus

$$\begin{aligned}
D &= \frac{p}{q} - \left(\frac{1}{1} + \cdots + \frac{1}{n}\right)\\
&\le \left(\frac{1}{1} + \cdots + \frac{1}{n} + \frac{1}{n+1}\right) - \left(\frac{1}{1} + \cdots + \frac{1}{n}\right)\\
&= \frac{1}{n+1}.
\end{aligned}$$

   In particular, $D < 1$. By Exercise 24, D may be written in Egyptian form:

$$D = \frac{1}{n_1} + \cdots + \frac{1}{n_k},$$

   where the n_i are distinct. Since

$$\frac{1}{n_i} \le D \le \frac{1}{1+n}, \qquad \text{for } i = 1, \ldots, k,$$

   $n < n + 1 \le n_i$ for $i = 1, \ldots, k$. It follows that

$$1, 2, \ldots, n, n_1, \ldots, n_k$$

   are distinct. Thus

$$\frac{p}{q} = D + \frac{1}{1} + \cdots + \frac{1}{n} = \frac{1}{n_1} + \cdots + \frac{1}{n_k} + \frac{1}{1} + \cdots + \frac{1}{n}$$

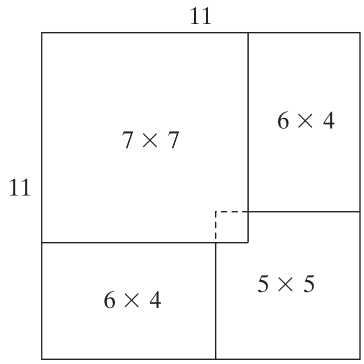
   is represented in Egyptian form.

27. In this solution, the induction is over the set X of odd integers $n > 5$, where 3 divides $n^2 - 1$. Such an induction can be justified by considering the first statement to be about the smallest integer in X, the second statement to be about the second smallest integer in X, and so on.

   Basis Steps ($n = 7, 11$) Exercise 35, Section 2.4, gives a solution if $n = 7$.

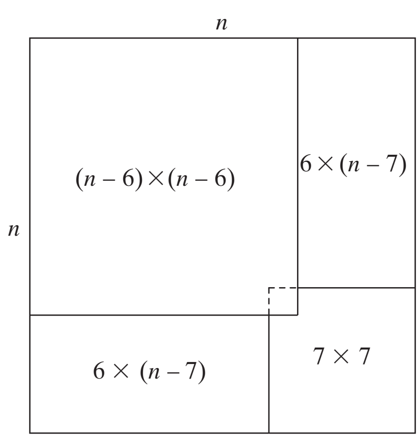
   If $n = 11$, enclose the missing square in a corner 7×7 subboard (see the following figure). Tile this subboard using the result of Exercise 35, Section 2.4. Tile the two 6×4 subboards using the result of Exercise 34, Section 2.4. Tile the

5×5 subboard with a corner square missing using the result of Exercise 32, Section 2.4. Thus the 11×11 board can be tiled.



Inductive Step Suppose that $n > 11$ and assume that if $k < n$, k is odd, $k > 5$, and 3 divides $k^2 - 1$, then a $k \times k$ deficient board can be tiled with trominoes.

Consider an $n \times n$ deficient board. Enclose the missing square in a corner $(n-6) \times (n-6)$ subboard. By the inductive assumption, this board can be tiled with trominoes. Tile the two $6 \times (n - 7)$ subboards using the result of Exercise 34, Section 2.4. Tile the deficient 7×7 subboard using the result of Exercise 35, Section 2.4. The $n \times n$ board is tiled, and the inductive step is complete.



31. Let X be a nonempty set of nonnegative integers. We must prove that X has a least element. Using mathematical induction, we prove that for all $n \geq 0$, if X contains an element less than or equal to n, then X has a least element. Notice that this proves that X has a least element. (Since X is nonempty, X contains an integer n. Now X contains an element less than or equal to n; so it follow that X has a least element.)

Basis Step ($n = 0$) If X contains an element less than or equal to 0, then X contains 0 since X consists of nonnegative integers. In this case, 0 is the least element in X.

Inductive Step Now we assume that if X contains an element less than or equal to n, then X has a least element. We must prove that if X contains an element less than or equal to $n + 1$, then X has a least element.

Suppose that X contains an element less than or equal to $n + 1$. We consider two cases: X contains an element less than or equal to n, and X does not contain an element less than or equal to n. If X contains an element less than or equal to n, by the inductive assumption, X has a least element. If X does not contain an element less than or equal to n, since X contains an element less than or equal to $n + 1$, X must contain $n + 1$, which is the least element in X. The inductive step is complete.

Chapter 2 Self-Test

1. Axioms are statements that are assumed to be true. Definitions are used to create new concepts in terms of existing ones.

2. Suppose that m and $m - n$ are odd. Then there exist integers k_1 and k_2 such that $m = 2k_1 + 1$ and $m - n = 2k_2 + 1$. Now

$$n = m - (m - n) = (2k_1 + 1) - (2k_2 + 1) = 2(k_1 - k_2).$$

Therefore n is even.

3. Since x and y are rational numbers, there exist integers m_1, n_1, m_2, n_2 such that $x = m_1/n_1$ and $y = m_2/n_2$. Since $y \neq 0$, $m_2 \neq 0$. Now

$$\frac{x}{y} = \frac{m_1/n_1}{m_2/n_2} = \frac{m_1 n_2}{n_1 m_2}.$$

Since x/y is the quotient of integers, it is rational.

4. We first prove that $X \subseteq Z$. Let $x \in X$. Since $X \subseteq Y$, $x \in Y$. Since $Y \subset Z$, $x \in Z$. Therefore $X \subseteq Z$.

We now show that X is a *proper* subset of Z. Since $Y \subset Z$, there exists $z \in Z$ such that $z \notin Y$. Now $z \notin X$, because if it were, we would have $z \in Y$. Therefore $X \subset Z$.

5. In a direct proof, the negated conclusion is not assumed, whereas in a proof by contradiction, the negated conclusion is assumed.

6. Suppose that if four teams play seven games, no pair of teams plays at least two times; or, equivalently, if four teams play seven games, each pair of teams plays at most one time. If the teams are A, B, C, and D and each pair of teams plays at most one time, the most games that can be played are:

A and B; A and C; A and D; B and C; B and D; C and D.

Thus at most six games can be played. This is a contradiction. Therefore, if four teams play seven games, some pair of teams plays at least two times.

7. We consider two cases: $a \leq b$ and $a > b$. In each of these two cases, we consider the two cases: $b \leq c$ and $b > c$.

First suppose that $a \leq b$. If $b \leq c$, then

$$\begin{aligned}\min\{\min\{a, b\}, c\} &= \min\{a, c\} = a = \min\{a, b\}\\ &= \min\{a, \min\{b, c\}\}.\end{aligned}$$

If $b > c$, then

$$\min\{\min\{a, b\}, c\} = \min\{a, c\} = \min\{a, \min\{b, c\}\}.$$

In either case,

$$\min\{\min\{a, b\}, c\} = \min\{a, \min\{b, c\}\}.$$

Now suppose that $a > b$. If $b \le c$, then

$$\begin{aligned}\min\{\min\{a,b\},c\} &= \min\{b,c\} = b = \min\{a,b\}\\ &= \min\{a,\min\{b,c\}\}.\end{aligned}$$

If $b > c$, then

$$\begin{aligned}\min\{\min\{a,b\},c\} &= \min\{b,c\} = c = \min\{a,c\}\\ &= \min\{a,\min\{b,c\}\}.\end{aligned}$$

In either case,

$$\min\{\min\{a,b\},c\} = \min\{a,\min\{b,c\}\}.$$

Therefore, for all a, b, c,

$$\min\{\min\{a,b\},c\} = \min\{a,\min\{b,c\}\}.$$

8. [(a) $\to$ (b)] We prove the contrapositive: If $A \cap \overline{B} \ne \varnothing$, then A is not a subset of B. Since $A \cap \overline{B} \ne \varnothing$, there exists x with $x \in A$ and $x \in \overline{B}$. Thus there exists x with $x \in A$ and $x \notin B$. Therefore A is not a subset of B.

[(b) $\to$ (c)] If $x \in B$, then $x \in A \cup B$. Therefore $B \subseteq A \cup B$.

Let $x \in A \cup B$. We must show that $x \in B$. Now $x \in A$ or $x \in B$. If $x \in B$, this part of the proof is complete; so suppose that $x \in A$. Since $A \cap \overline{B} = \varnothing$, $x \notin \overline{B}$. Thus $x \in B$ and $A \cup B \subseteq B$. Therefore $A \cup B = B$.

[(c) $\to$ (a)] Let $x \in A$. Then $x \in A \cup B$. Since $A \cup B = B$, $x \in B$. Therefore $A \subseteq B$.

9.
$$\begin{aligned}(p \vee q) \to r &\equiv \neg(p \vee q) \vee r\\ &\equiv \neg p \neg q \vee r\\ &\equiv (\neg p \vee r)(\neg q \vee r)\end{aligned}$$

10.
$$\begin{aligned}(p \vee \neg q) \to \neg rs &\equiv \neg(p \vee \neg q) \vee \neg rs\\ &\equiv \neg pq \vee \neg rs\\ &\equiv (\neg p \vee \neg r)(\neg p \vee s)(q \vee \neg r)(q \vee s)\end{aligned}$$

11.
1. $\neg p \vee q$
2. $\neg q \vee \neg r$
3. $p \vee \neg r$
4. $\neg p \vee \neg r$ From 1 and 2
5. $\neg r$ From 3 and 4

12.
1. $\neg p \vee q$
2. $\neg q \vee \neg r$
3. $p \vee \neg r$
4. r Negation of conclusion
5. $\neg p \vee \neg r$ From 1 and 2
6. $\neg r$ From 3 and 5

Now 4 and 6 give a contradiction.

In Exercises 13–16, only the Inductive Step is given.

13. $2 + 4 + \cdots + 2n + 2(n+1) = n(n+1) + 2(n+1) = (n+1)(n+2)$

14.
$$2^2 + 4^2 + \cdots + (2n)^2 + [2(n+1)]^2 = \frac{2n(n+1)(2n+1)}{3} + [2(n+1)]^2 = \frac{2(n+1)(n+2)[2(n+1)+1]}{3}$$

15.
$$\frac{1}{2!} + \frac{2}{3!} + \cdots + \frac{n}{(n+1)!} + \frac{n+1}{(n+2)!} = 1 - \frac{1}{(n+1)!} + \frac{n+1}{(n+2)!} = 1 - \frac{1}{(n+2)!}$$

16.
$$\begin{aligned}2^{n+2} &= 2 \cdot 2^{n+1} < 2[1 + (n+1)2^n] = 2 + (n+1)2^{n+1}\\ &= 1 + [1 + (n+1)2^{n+1}]\\ &< 1 + [2^{n+1} + (n+1)2^{n+1}]\\ &= 1 + (n+2)2^{n+1}\end{aligned}$$

17. $q = 9, r = 2$

18. $c_2 = 2, c_3 = 3, c_4 = 8, c_5 = 9.$

19. **Basis Step ($n = 1$)** $c_1 = 0 \le 0 = 1 \lg 1$

Inductive Step

$$\begin{aligned}c_n &= 2c_{\lfloor n/2 \rfloor} + n\\ &\le 2\lfloor n/2 \rfloor \lg \lfloor n/2 \rfloor + n\\ &\le 2(n/2)\lg(n/2) + n\\ &= n(\lg n - 1) + n = n \lg n\end{aligned}$$

20. Let X be a nonempty set of nonnegative integers that has an upper bound. We must show that X contains a largest element.

Let Y be the set of integer upper bounds for X. By assumption, Y is nonempty. Since X consists of nonnegative integers, Y also consists of nonnegative integers. By the Well-Ordering Property, Y has a least element, say n. Since Y consists of upper bounds for X, $k \le n$ for every k in X. Suppose, by way of contradiction, that n is not in X. Then $k \le n - 1$ for every k in X. Thus, $n - 1$ is an upper bound for X, which is a contradiction. Therefore, n is in X. Since $k \le n$ for every k in X, n is the largest element in X.

Section 3.1 Review

1. Let X and Y be sets. A function f from X to Y is a subset of the Cartesian product $X \times Y$ having the property that for each $x \in X$, there is exactly one $y \in Y$ with $(x, y) \in f$.
2. In an arrow diagram of the function f, there is an arrow from i to j if $(i, j) \in f$.
3. The graph of a function f, whose domain and codomain are subsets of the real numbers, consists of the points in the plane that correspond to the elements in f.
4. A set S of points in the plane defines a function when each vertical line intersects at most one point of S.
5. The remainder when x is divided by y
6. A hash function takes a data item to be stored or retrieved and computes the first choice for a location for the item.

7. A collision occurs for a hash function H if $H(x) = H(y)$ but $x \neq y$.

8. When a collision occurs, a collision resolution policy determines an alternative location for one of the data items.

9. Pseudorandom numbers are numbers that appear random even though they are generated by a program.

10. A linear congruential random number generator uses a formula of the form

$$x_n = (ax_{n-1} + c) \bmod m.$$

Given the pseudorandom number x_{n-1}, the next pseudorandom number x_n is given by the formula. A "seed" is used as the first pseudorandom number in the sequence. As an example, the formula

$$x_n = (7x_{n-1} + 5) \bmod 11$$

with seed 3 gives a sequence that begins $3, 4, 0, 5, \ldots$.

11. The floor of x is the greatest integer less than or equal to x. It is denoted $\lfloor x \rfloor$.

12. The ceiling of x is the least integer greater than or equal to x. It is denoted $\lceil x \rceil$.

13. A function f from X to Y is said to be one-to-one if for each $y \in Y$, there is at most one $x \in X$ with $f(x) = y$. The function $\{(a, 1), (b, 3), (c, 0)\}$ is one-to-one. If a function from X to Y is one-to-one, each element in Y in its arrow diagram will have at most one arrow pointing to it.

14. A function f from X to Y is said to be onto Y if the range of f is Y. The function $\{(a, 1), (b, 3), (c, 0)\}$ is onto $\{0, 1, 3\}$. If a function from X to Y is onto Y, each element in Y in its arrow diagram will have at least one arrow pointing to it.

15. A bijection is a function that is one-to-one and onto. The function of Exercises 13 and 14 is one-to-one and onto $\{0, 1, 3\}$.

16. If f is a one-to-one, onto function from X to Y, the inverse function is

$$f^{-1} = \{(y, x) \mid (x, y) \in f\}.$$

If f is the function of Exercises 13 and 14, we have

$$f^{-1} = \{(1, a), (3, b), (0, c)\}.$$

Given the arrow diagram for a one-to-one, onto function f from X to Y, we can obtain the arrow diagram for f^{-1} by reversing the direction of each arrow.

17. Suppose that g is a function from X to Y and f is a function from Y to Z. The composition function from X to Z is defined as

$$f \circ g = \{(x, z) \mid (x, y) \in g \text{ and } (y, z) \in f \text{ for some } y \in Y\}.$$

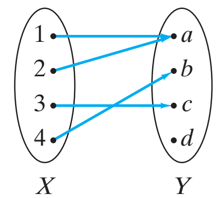
If $g = \{(1, 2), (2, 2)\}$ and $f = \{(2, a)\}$, $f \circ g = \{(1, a), (2, a)\}$. Given the arrow diagrams for functions g from X to Y and f from Y to Z, we can obtain the arrow diagram of $f \circ g$ by drawing an arrow from $x \in X$ to $z \in Z$ provided that there are arrows from x to some $y \in Y$ and from y to z.

18. A binary operator on X is a function from $X \times X$ to X. The addition operator $+$ is a binary operator on the set of integers.

19. A unary operator on X is a function from X to X. The minus operator $-$ is a unary operator on the set of integers.

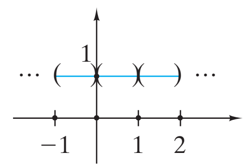
Section 3.1

1. It is a function from X to Y; domain $= X$, range $= \{a, b, c\}$; it is neither one-to-one nor onto. Its arrow diagram is

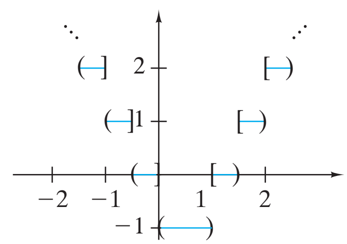


4. It is not a function (from X to Y).

6.



9.



10. The function f is both one-to-one and onto. To prove that f is one-to-one, suppose that $f(n) = f(m)$. Then $n + 1 = m + 1$. Thus $n = m$. Therefore f is one-to-one.

To prove that f is onto, let m be an integer. Then $f(m - 1) = (m - 1) + 1 = m$. Therefore f is onto.

13. The function f is neither one-to-one nor onto. Since $f(-1) = |-1| = 1 = f(1)$, f is not one-to-one. Since $f(n) \geq 0$ for all $n \in \mathbf{Z}$, $f(n) \neq -1$ for all $n \in \mathbf{Z}$. Therefore f is not onto.

16. The function f is not one-to-one, but it is onto. Since $f(2, 1) = 2 - 1 = 1 = 3 - 2 = f(3, 2)$, f is not one-to-one. Suppose that $k \in \mathbf{Z}$. Then $f(k, 0) = k - 0 = k$. Therefore f is onto.

19. The function f is neither one-to-one nor onto. Since $f(2, 1) = 2^2 + 1^2 = 1^2 + 2^2 = f(1, 2)$, f is not one-to-one. Since $f(m, n) \geq 0$ for all $m, n \in \mathbf{Z}$, $f(m, n) \neq -1$ for all $m, n \in \mathbf{Z}$. Therefore f is not onto.

22. Suppose that $f(a, b) = f(c, d)$. Then $2^a 3^b = 2^c 3^d$. We claim that $a = c$. If not, either $a > c$ or $a < c$. We assume that $a > c$. (The argument is the same if $a < c$.) We may then cancel 2^c from both sides of $2^a 3^b = 2^c 3^d$ to obtain $2^{a-c} 3^b = 3^d$. Since $a - c > 0$, $2^{a-c} 3^b$ is even. Since 3^d is odd, we have a contradiction. Therefore $a = c$.

We may now cancel 2^a from both sides of $2^a 3^b = 2^c 3^d$ to obtain $3^b = 3^d$. An argument like that in the preceding paragraph shows that $b = d$. Since $a = c$ and $b = d$, f is one-to-one.

Since $f(m, n) \neq 5$ for all $m, n \in \mathbf{Z}^+$, f is not onto. (Note that $f(m, n) \geq 6$ for all $m, n \in \mathbf{Z}^+$.)

23. f is both one-to-one and onto.

26. f is both one-to-one and onto.

29. Define a function f from $\{1, 2, 3, 4\}$ to $\{a, b, c, d, e\}$ as

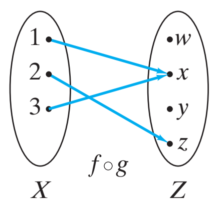
$$f = \{(1, a), (2, c), (3, b), (4, d)\}.$$

Then f is one-to-one, but not onto.

32. $f^{-1}(y) = (y - 2)/4$

35. $f^{-1}(y) = 1/(y - 3)$

38. $f \circ g = \{(1, x), (2, z), (3, x)\}$



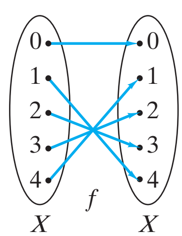
41. $(f \circ f)(x) = 2\lfloor 2x \rfloor$, $(g \circ g)(x) = x^4$, $(f \circ g)(x) = \lfloor 2x^2 \rfloor$, $(g \circ f)(x) = \lfloor 2x \rfloor^2$

42. Let $g(x) = \log_2 x$ and $h(x) = x^2+2$. Then $f(x) = (g \circ h)(x)$.

45. Let $g(x) = 2x$ and $h(x) = \sin x$. Then $f(x) = (g \circ h)(x)$.

48. $f = \{(-5, 25), (-4, 16), (-3, 9), (-2, 4), (-1, 1), (0, 0), (1, 1), (2, 4), (3, 9), (4, 16), (5, 25)\}$. f is neither one-to-one nor onto. We omit the arrow diagram of f.

51. $f = \{(0, 0), (1, 4), (2, 3), (3, 2), (4, 1)\}$; f is one-to-one and onto. The arrow diagram of f is



54. 6

In the solutions to Exercises 55 and 58, a : b means "store item a in cell b."

55. 53 : 9, 13 : 2, 281 : 6, 743 : 7, 377 : 3, 20 : 10, 10 : 0, 796 : 4

58. 714 : 0, 631 : 6, 26 : 5, 373 : 1, 775 : 8, 906 : 13, 509 : 2, 2032 : 7, 42 : 4, 4 : 3, 136 : 9, 1028 : 10

61. During a search, if we stop the search at an empty cell, we may not find the item even if it is present. The cell may be empty because an item was deleted. One solution is to mark deleted cells and consider them nonempty during a search.

62. False. Take $g = \{(1, a), (2, b)\}$ and $f = \{(a, z), (b, z)\}$.

65. True. Let $z \in Z$. Since f is onto, there exists $y \in Y$ such that $f(y) = z$. Since g is onto, there exists $x \in X$ such that $g(x) = y$. Now $f(g(x)) = f(y) = z$. Therefore $f \circ g$ is onto.

68. True. Suppose that $g(x_1) = g(x_2)$. Then $f(g(x_1)) = f(g(x_2))$. Since $f \circ g$ is one-to-one, $x_1 = x_2$. Therefore, g is one-to-one.

70. $g(S) = \{a\}$, $g(T) = \{a, c\}$, $g^{-1}(U) = \{1\}$, $g^{-1}(V) = \{1, 2, 3\}$

75. No. Let $f(x) = x$ and $g(x) = x^2$. Then

$$E_1(f) = f(1) = 1 = g(1) = E_1(g).$$

77. 101

80. Suppose that $S(Y_1) = s_1s_2s_3 = S(Y_2)$. Now $a \in Y_1$ if and only if $s_1 = 1$ if and only if $a \in Y_2$. Also $b \in Y_1$ if and only if $s_2 = 1$ if and only if $b \in Y_2$. Also $c \in Y_1$ if and only if $s_3 = 1$ if and only if $c \in Y_2$. It follows that $Y_1 = Y_2$ and S is one-to-one.

82. If $x \in X \cap Y$, $C_{X \cap Y}(x) = 1 = 1 \cdot 1 = C_X(x)C_Y(x)$. If $x \notin X \cap Y$, then $C_{X \cap Y}(x) = 0$. Since either $x \notin X$ or $x \notin Y$, either $C_X(x) = 0$ or $C_Y(x) = 0$. Thus $C_X(x)C_Y(x) = 0 = C_{X \cap Y}(x)$.

85. If $x \in X - Y$, then

$$C_{X-Y}(x) = 1 = 1 \cdot [1 - 0] = C_X(x)[1 - C_Y(x)].$$

If $x \notin X - Y$, then either $x \notin X$ or $x \in Y$. In case $x \notin X$,

$$C_{X-Y}(x) = 0 = 0 \cdot [1 - C_Y(x)] = C_X(x)[1 - C_Y(x)].$$

In case $x \in Y$,

$$C_{X-Y}(x) = 0 = C_X(x)[1 - 1] = C_X(x)[1 - C_Y(x)].$$

Thus the equation holds for all $x \in U$.

88. f is onto by definition. Suppose that $f(X) = f(Y)$. Then $C_X(x) = C_Y(x)$, for all $x \in U$. Suppose that $x \in X$. Then $C_X(x) = 1$. Thus $C_Y(x) = 1$. Therefore, $x \in Y$. This argument shows that $X \subseteq Y$. Similarly, $Y \subseteq X$. Therefore $X = Y$ and f is one-to-one.

90. f is a commutative, binary operator.

93. f is not a binary operator since $f(x, 0)$ is not defined.

95. $g(x) = -x$

98. The statement is true. The least integer greater than or equal to x is the unique integer k satisfying

$$k - 1 < x \leq k.$$

Now

$$k + 2 < x + 3 \leq k + 3.$$

Thus, $k + 3$ is the least integer greater than or equal to $x + 3$. Therefore, $k + 3 = \lceil x + 3 \rceil$. Since $k = \lceil x \rceil$, we have

$$\lceil x + 3 \rceil = k + 3 = \lceil x \rceil + 3.$$

101. If n is an odd integer, $n = 2k + 1$ for some integer k. Now

$$\frac{n^2}{4} = \frac{(2k + 1)^2}{4} = \frac{4k^2 + 4k + 1}{4} = k^2 + k + \frac{1}{4}.$$

Since $k^2 + k$ is an integer,

$$\left\lfloor \frac{n^2}{4} \right\rfloor = k^2 + k.$$

The result now follows because

$$\begin{aligned}\left(\frac{n-1}{2}\right)\left(\frac{n+1}{2}\right) &= \left[\frac{(2k+1)-1}{2}\right]\left[\frac{(2k+1)+1}{2}\right] \\ &= \frac{2k(2k+2)}{4} \\ &= \frac{4k^2+4k}{4} = k^2+k.\end{aligned}$$

104. Let $k = \lceil x \rceil$. Then $k - 1 < x \le k$ and $2x \le 2k$. Thus $\lceil 2x \rceil \le 2k = 2\lceil x \rceil$. Now $\lceil x \rceil = k < x + 1$. Therefore $2\lceil x \rceil < 2x + 2 \le \lceil 2x \rceil + 2$, so $2\lceil x \rceil - 2 < \lceil 2x \rceil$. Therefore $2\lceil x \rceil - 1 \le \lceil 2x \rceil$.

107. April, July

Section 3.2 Review

1. A sequence is a function in which the domain consists of a set of consecutive integers.

2. If s_n denotes element n of the sequence, we call n the index of the sequence.

3. A sequence s is increasing if $s_n < s_{n+1}$ for all n.

4. A sequence s is decreasing if $s_n > s_{n+1}$ for all n.

5. A sequence s is nonincreasing if $s_n \ge s_{n+1}$ for all n.

6. A sequence s is nondecreasing if $s_n \le s_{n+1}$ for all n.

7. Let $\{s_n\}$ be a sequence defined for $n = m, m+1, \ldots$, and let $n_1, n_2, \ldots$ be an increasing sequence whose values are in the set $\{m, m+1, \ldots\}$. We call the sequence $\{s_{n_k}\}$ a subsequence of $\{s_n\}$.

8. $a_m + a_{m+1} + \cdots + a_n$

9. $a_m a_{m+1} \cdots a_n$

10. A string over X is a finite sequence of elements from X.

11. The null string is the string with no elements.

12. X^* is the set of all strings over X.

13. X^+ is the set of all nonnull strings over X.

14. The length of a string α is the number of elements in α. It is denoted $|\alpha|$.

15. The concatenation of strings α and β is the string consisting of α followed by β. It is denoted $\alpha\beta$.

16. A string β is a substring of the string α if there are strings γ and δ with $\alpha = \gamma\beta\delta$.

Section 3.2

1. c

2. c

3. $cddcdc$

25. 52

26. 52

27. No

28. No

29. No

30. Yes

39. 12

40. 23

41. 7

42. 46

43. 1

44. 3

45. 3

46. 21

47. No

48. No

49. No

50. Yes

67. 15

68. 155

69. $2n + 3(n-1)n/2$

70. Yes

71. No

72. No

73. Yes

83. 1, 3, 5, 7, 9, 11, 13

84. 1, 5, 9, 13, 17, 21, 25

85. $n_k = 2k - 1$

86. $s_{n_k} = 4k - 3$

91. 88

92. 1140

93. 48

94. 3168

111. $b_1 = 1, b_2 = 2, b_3 = 3, b_4 = 4, b_5 = 5, b_6 = 126$

114. Let $s_0 = 0$. Then

$$\begin{aligned}\sum_{k=1}^{n} a_k b_k &= \sum_{k=1}^{n} (s_k - s_{k-1}) b_k \\ &= \sum_{k=1}^{n} s_k b_k - \sum_{k=1}^{n} s_{k-1} b_k \\ &= \sum_{k=1}^{n} s_k b_k - \sum_{k=1}^{n} s_k b_{k+1} + s_n b_{n+1} \\ &= \sum_{k=1}^{n} s_k (b_k - b_{k+1}) + s_n b_{n+1}.\end{aligned}$$

117. 00, 01, 10, 11

120. 000, 010, 001, 011, 100, 110, 101, 111, 00, 01, 11, 10, 0, 1, λ

123. **Basis Step ($n = 1$)** In this case, $\{1\}$ is the only nonempty subset of $\{1\}$, so the sum is

$$\frac{1}{1} = 1 = n.$$

Inductive Step Assume that the statement is true for n. We divide the subsets of

$$\{1, \ldots, n, n+1\}$$

into two classes:

$$\begin{aligned}C_1 &= \text{class of nonempty subsets that do not contain } n+1 \\ C_2 &= \text{class of subsets that contain } n+1.\end{aligned}$$

By the inductive assumption,

$$\sum_{C_1} \frac{1}{n_1 \cdots n_k} = n.$$

Since a set in C_2 consists of $n + 1$ together with a subset (empty or nonempty) of $\{1, \ldots, n\}$,

$$\sum_{C_2} \frac{1}{(n+1)n_1 \cdots n_k} = \frac{1}{n+1} + \frac{1}{n+1} \sum_{C_1} \frac{1}{n_1 \cdots n_k}.$$

[The term $1/(n+1)$ results from the subset $\{n+1\}$.] By the inductive assumption,

$$\frac{1}{n+1} + \frac{1}{n+1}\sum_{C_1}\frac{1}{n_1\cdots n_k} = \frac{1}{n+1} + \frac{1}{n+1}\cdot n = 1.$$

Therefore,

$$\sum_{C_2}\frac{1}{(n+1)n_1\cdots n_k} = 1.$$

Finally,

$$\sum_{C_1\cup C_2}\frac{1}{n_1\cdots n_k} = \sum_{C_1}\frac{1}{n_1\cdots n_k} + \sum_{C_2}\frac{1}{(n+1)n_1\cdots n_k} = n+1.$$

125. Since $x_1 \le x \le x_n$, $|x-x_1| = x-x_1$ and $|x-x_n| = x_n - x$. Thus

$$\begin{aligned}\sum_{i=1}^{n}|x-x_i| &= |x-x_1| + \sum_{i=2}^{n-1}|x-x_i| + |x-x_n|\\ &= (x-x_1) + \sum_{i=2}^{n-1}|x-x_i| + (x_n - x)\\ &= \sum_{i=2}^{n-1}|x-x_i| + (x_n - x_1).\end{aligned}$$

128. Using Exercise 4, Section 2.4, we have

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{n}(i-j)^2 &= \sum_{i=1}^{n}\sum_{j=1}^{n}(i^2 - 2ij + j^2)\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}i^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{n}ij + \sum_{i=1}^{n}\sum_{j=1}^{n}j^2\\ &= \sum_{j=1}^{n}\sum_{i=1}^{n}i^2 - 2\sum_{i=1}^{n}i\sum_{j=1}^{n}j + \sum_{i=1}^{n}\sum_{j=1}^{n}j^2\\ &= \sum_{j=1}^{n}\frac{n(n+1)(2n+1)}{6} - 2\left[\frac{n(n+1)}{2}\right]^2\\ &\quad + \sum_{i=1}^{n}\frac{n(n+1)(2n+1)}{6}\\ &= n\left[\frac{n(n+1)(2n+1)}{6}\right] - \frac{n^2(n+1)^2}{2}\\ &\quad + n\left[\frac{n(n+1)(2n+1)}{6}\right]\\ &= 2n\left[\frac{n(n+1)(2n+1)}{6}\right] - \frac{n^2(n+1)^2}{2}\\ &= \frac{n^2(n+1)[2(2n+1)-3(n+1)]}{6}\\ &= \frac{n^2(n+1)[n-1]}{6} = \frac{n^2(n^2-1)}{6}.\end{aligned}$$

129. The function f is one-to-one. Suppose that $f(\alpha) = f(\beta)$. Then $\alpha ab = \beta ab$. Thus $\alpha = \beta$.

The function f is not onto. Since $|f(\alpha)| \ge 2$ for all $\alpha \in X^*$, $f(\alpha) \ne \lambda$ for all $\alpha \in X^*$.

132. Let $\alpha = \lambda$. Then $\alpha \in L$ and the first rule states that $ab = a\alpha b \in L$. Now $\beta = ab \in L$ and the first rule states that $aabb = a\beta b \in L$. Now $\gamma = aabb \in L$ and the first rule states that $aaabbb = a\gamma b \in L$.

135. We use strong induction on the length n of α to show that if $\alpha \in L$, α has an equal number of a's and b's.

The Basis Step is $n = 0$. In this case, α is the null string, which has an equal number of a's and b's.

We turn now to the Inductive Step. We assume any string in L of length $k < n$ has an equal number of a's and b's. We must show that any string in L of length n has an equal number of a's and b's. Let $\alpha \in L$ and suppose that $|\alpha| = n > 0$. Now α is in L because of either rule 1 or rule 2.

Suppose that α is in L because of rule 1. In this case, $\alpha = a\beta b$ or $\alpha = b\beta a$, where $\beta \in L$. Since $|\beta| < n$, by the inductive hypothesis β has an equal number of a's and b's. Since $\alpha = a\beta b$ or $\alpha = b\beta a$, α also has an equal number of a's and b's.

Suppose that α is in L because of rule 2. In this case, $\alpha = \beta\gamma$, where $\beta \in L$ and $\gamma \in L$. Since $|\beta| < n$ and $|\gamma| < n$, by the inductive hypothesis β and γ each have equal numbers of a's and b's. Since $\alpha = \beta\gamma$, α also has an equal number of a's and b's. The proof by induction is complete.

Section 3.3 Review

1. A binary relation from a set X to a set Y is a subset of the Cartesian product $X \times Y$.
2. In a digraph of a relation on X, vertices represent the elements of X and directed edges from x to y represent the elements (x, y) in the relation.
3. A relation R on a set X is reflexive if $(x, x) \in R$ for every $x \in X$. The relation $\{(1, 1), (2, 2)\}$ is a reflexive relation on $\{1, 2\}$. The relation $\{(1, 1)\}$ is not a reflexive relation on $\{1, 2\}$.
4. A relation R on a set X is symmetric if for all $x, y \in X$, if $(x, y) \in R$, then $(y, x) \in R$. The relation $\{(1, 2), (2, 1)\}$ is a symmetric relation on $\{1, 2\}$. The relation $\{(1, 2)\}$ is not a symmetric relation on $\{1, 2\}$.
5. A relation R on a set X is antisymmetric if for all $x, y \in X$, if $(x, y) \in R$ and $(y, x) \in R$, then $x = y$. The relation $\{(1, 2)\}$ is an antisymmetric relation on $\{1, 2\}$. The relation $\{(1, 2), (2, 1)\}$ is not an antisymmetric relation on $\{1, 2\}$.
6. A relation R on a set X is transitive if for all $x, y, z \in X$, if (x, y) and $(y, z) \in R$, then $(x, z) \in R$. The relation $\{(1, 2), (2, 3), (1, 3)\}$ is a transitive relation on $\{1, 2, 3\}$. The relation $\{(1, 2), (2, 1)\}$ is not a transitive relation on $\{1, 2\}$.
7. A relation R on a set X is a partial order if R is reflexive, antisymmetric, and transitive. The relation
$$\{(1, 1), (2, 2), (3, 3), (1, 2), (2, 3), (1, 3)\}$$
is a partial order on $\{1, 2, 3\}$.
8. If R is a relation from X to Y, the inverse of R is the relation from Y to X:
$$R^{-1} = \{(y, x) \mid (x, y) \in R\}.$$
The inverse of the relation $\{(1, 2), (1, 3)\}$ is $\{(2, 1), (3, 1)\}$.

9. Let R_1 be a relation from X to Y and R_2 be a relation from Y to Z. The composition of R_1 and R_2 is the relation from X to Z

$$R_2 \circ R_1 = \{(x, z) \mid (x, y) \in R_1 \text{ and } (y, z) \in R_2 \text{ for some } y \in Y\}.$$

The composition of the relations

$$R_1 = \{(1, 2), (1, 3), (2, 2)\}$$

and

$$R_2 = \{(2, 1), (2, 3), (1, 4)\}$$

is

$$R_2 \circ R_1 = \{(1, 1), (1, 3), (2, 1), (2, 3)\}.$$

Section 3.3

1. {(8840, Hammer), (9921, Pliers), (452, Paint), (2207, Carpet)}

4. $\{(a, a), (b, b)\}$

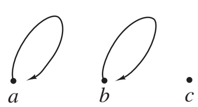
5.

a	6
b	2
a	1
c	1

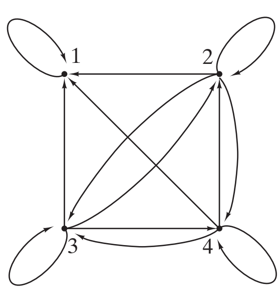
8.

Mercury	1
Venus	2
Earth	3
Mars	4
Jupiter	5
Saturn	6
Uranus	7
Neptune	8

9.



12.



13. $\{(a, b), (a, c), (b, a), (b, d), (c, c), (c, d)\}$

16. $\{(b, c), (c, b), (d, d)\}$

17. (Exercise 1) {(Hammer, 8840), (Pliers, 9921), (Paint, 452), (Carpet, 2207)}

18. $\{(1, 1), (1, 4), (2, 2), (2, 5), (3, 3), (4, 1), (4, 4), (5, 2), (5, 5)\}$

20. $R = R^{-1} = \{(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 1), (2, 2), (2, 3), (2, 4), (3, 1), (3, 2), (3, 3), (4, 1), (4, 2), (5, 1)\}$

23. Antisymmetric

24. Antisymmetric

27. Reflexive, symmetric, antisymmetric, transitive, partial order

30. Antisymmetric

32. Reflexive, antisymmetric, transitive, partial order

35. Reflexive: Suppose that (x_1, x_2) is in $X_1 \times X_2$. Since R_i is reflexive, $x_1 R_1 x_1$ and $x_2 R_2 x_2$. Thus $(x_1, x_2) R (x_1, x_2)$.

Antisymmetric: Suppose that $(x_1, x_2) R (x_1', x_2')$ and $(x_1', x_2') R (x_1, x_2)$. Then $x_1 R_1 x_1'$ and $x_1' R_1 x_1$. Since R_1 is antisymmetric, $x_1 = x_1'$. Similarly, $x_2 = x_2'$. Therefore $(x_1, x_2) = (x_1', x_2')$ and R is antisymmetric.

Transitivity is proved similarly.

37. $\{(1, 1), (2, 2), (3, 3), (4, 4), (1, 2), (2, 3), (2, 1), (3, 2)\}$

40. $\{(1, 1), (1, 2), (2, 1), (2, 2)\}$

42. True. Let $(x, y), (y, z) \in R^{-1}$. Then $(z, y), (y, x) \in R$. Since R is transitive, $(z, x) \in R$. Thus $(x, z) \in R^{-1}$. Therefore, R^{-1} is transitive.

45. True. We must show that $(x, x) \in R \circ S$ for all $x \in X$. Let $x \in X$. Since R and S are reflexive, $(x, x) \in R$ and $(x, x) \in S$. Therefore, $(x, x) \in R \circ S$ and $R \circ S$ is reflexive.

48. True. Let $(x, y) \in R \cap S$. Then $(x, y) \in R$ and $(x, y) \in S$. Since R and S are symmetric, $(y, x) \in R$ and $(y, x) \in S$. Therefore, $(y, x) \in R \cap S$ and $R \cap S$ is symmetric.

51. False. Let $R = \{(1, 2)\}$, $S = \{(2, 1)\}$.

54. True. Suppose that $(x, y), (y, x) \in R^{-1}$. Then $(y, x), (x, y) \in R$. Since R is antisymmetric, $y = x$. Therefore R^{-1} is antisymmetric.

56. R is reflexive and symmetric. R is not antisymmetric, not transitive, and not a partial order.

Section 3.4 Review

1. An equivalence relation is a relation that is reflexive, symmetric, and transitive. The relation

$$\{(1, 1), (2, 2), (3, 3), (1, 2), (2, 1)\}$$

is an equivalence relation on $\{1, 2, 3\}$. The relation

$$\{(1, 1), (3, 3), (1, 2), (2, 1)\}$$

is not an equivalence relation on $\{1, 2, 3\}$.

2. Let R be an equivalence relation on X. The equivalence classes of X given by R are sets of the form

$$\{x \in X \mid x R a\},$$

where $a \in X$.

3. If R is an equivalence relation on X, the equivalence classes partition X. Conversely, if $\mathcal{S}$ is a partition of X and we define $x R y$ to mean that for some $S \in \mathcal{S}$, both x and y belong to S, then R is an equivalence relation.

Section 3.4

1. Equivalence relation: $[1] = [3] = \{1, 3\}$, $[2] = \{2\}$, $[4] = \{4\}$, $[5] = \{5\}$

4. Equivalence relation: $[1] = [3] = [5] = \{1, 3, 5\}$, $[2] = \{2\}$, $[4] = \{4\}$

7. Not an equivalence relation (neither transitive nor reflexive)

9. The relation is an equivalence relation.

12. The relation is not an equivalence relation. It is neither reflexive nor symmetric.

15. $\{(1, 1), (1, 2), (2, 1), (2, 2), (3, 3), (3, 4), (4, 3), (4, 4)\}$,
$[1] = [2] = \{1, 2\}$, $[3] = [4] = \{3, 4\}$

18. $\{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3), (4, 4)\}$,
$[1] = [2] = [3] = \{1, 2, 3\}$, $[4] = \{4\}$

22. $\{1\}, \{1, 3\}, \{1, 4\}, \{1, 3, 4\}$

24. [Part(b)]
{San Francisco, San Diego, Los Angeles}, {Pittsburgh, Philadelphia}, {Chicago}

26. $R = \{(x, x) \mid x \in X\}$

29. Suppose that $x\,R\,y$. Since R is reflexive, $y\,R\,y$. Taking $z = y$ in the given condition, we have $y\,R\,x$. Therefore R is symmetric.

Now suppose that $x\,R\,y$ and $y\,R\,z$. The given condition tells us that $z\,R\,x$. Since R is symmetric, $x\,R\,z$. Therefore R is transitive. Since R is reflexive, symmetric, and transitive, R is an equivalence relation.

31. [Part (b)]
$(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (1, 9), (1, 10), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1), (9, 1), (10, 1)$

34. (a) We show symmetry only. Let $(x, y) \in R_1 \cap R_2$. Then $(x, y) \in R_1$ and $(x, y) \in R_2$. Since R_1 and R_2 are symmetric, $(y, x) \in R_1$ and $(y, x) \in R_2$. Thus $(y, x) \in R_1 \cap R_2$ and, therefore, $R_1 \cap R_2$ is symmetric.

(b) A is an equivalence class of $R_1 \cap R_2$ if and only if there are equivalence classes A_1 of R_1 and A_2 of R_2 such that $A = A_1 \cap A_2$.

37. [Part (b)] Torus

40. If $x \in X$, then $x \in f^{-1}(f(\{x\}))$. Thus $\cup\{S \mid S \in \mathcal{S}\} = X$. Suppose that

$$a \in f^{-1}(\{y\}) \cap f^{-1}(\{z\})$$

for some $y, z \in Y$. Then $f(a) = y$ and $f(a) = z$. Thus $y = z$. Therefore, $\mathcal{S}$ is a partition of X. The equivalence relation that generates this partition is given in Exercise 38.

43. Suppose, by way of contradiction, that $a \in [b]$. Then $(a, b) \in R$. Since R is symmetric, $(b, a) \in R$. Since R is transitive, $(b, b) \in R$, which is a contradiction. Therefore $[b] = \varnothing$.

46. Since R is not transitive, there exist $(a, b), (b, c) \in R$, but $(a, c) \notin R$. Then $a \in [b]$, $b \in [c]$, and $a \notin [c]$. Since R is reflexive, $b \in [b]$. Therefore $[b] \cap [c] \neq \varnothing$, but $[b] \neq [c]$. Thus the collection of pseudo equivalence classes does not partition X.

50. $\rho(R_1) = \{(1, 1), (2, 2), (3, 3), (4, 4), (1, 2), (3, 4), (4, 2)\}$
$\sigma(R_1) = \{(1, 1), (2, 1), (1, 2), (3, 4), (4, 3), (4, 2), (2, 4)\}$
$\tau(R_1) = \{(1, 1), (1, 2), (3, 4), (4, 2), (3, 2)\}$
$\tau(\sigma(\rho(R_1))) = \{(x, y) \mid x, y \in \{1, 2, 3, 4\}\}$

53. Let $(x, y), (x, z) \in \tau(R)$. Then $(x, y) \in R^m$ and $(y, z) \in R^n$. Thus $(x, z) \in R^{m+n}$. Therefore, $(x, z) \in \tau(R)$ and $\tau(R)$ is transitive.

56. Suppose that R is transitive. If $(x, y) \in \tau(R) = \cup\{R^n\}$, then there exist $x = x_0, \ldots, x_n = y \in X$ such that $(x_{i-1}, x_i) \in R$ for $i = 1, \ldots, n$. Since R is transitive, it follows that $(x, y) \in R$. Thus $R \supseteq \tau(R)$. Since we always have $R \subseteq \tau(R)$, it follows that $R = \tau(R)$.

Suppose that $\tau(R) = R$. By Exercise 53, $\tau(R)$ is transitive. Therefore, R is transitive.

57. True. Let $D = \{(x, x) \mid x \in X\}$. Then, by definition, $\rho(R) = R \cup D$, where R is any relation on X. Now

$$\begin{aligned}\rho(R_1 \cup R_2) = (R_1 \cup R_2) \cup D &= (R_1 \cup D) \cup (R_2 \cup D)\\ &= \rho(R_1) \cup \rho(R_2).\end{aligned}$$

60. False. Let $R_1 = \{(1, 2), (2, 3)\}$, $R_2 = \{(1, 3), (3, 4)\}$.

63. True. Using the notation and hint for Exercise 57,

$$\rho(\tau(R_1)) = \tau(R_1) \cup D$$

and

$$\tau(\rho(R_1)) = \tau(R_1 \cup D).$$

So we must show that $\tau(R_1) \cup D = \tau(R_1 \cup D)$.

We first note that if $A \subseteq B$, then $\tau(A) \subseteq \tau(B)$. Now $R_1 \subseteq R_1 \cup D$. Therefore, $\tau(R_1) \subseteq \tau(R_1 \cup D)$. Also, $D \subseteq R_1 \cup D$. Therefore, $D = \tau(D) \subseteq \tau(R_1 \cup D)$. It follows that $\tau(R_1) \cup D \subseteq \tau(R_1 \cup D)$.

Since $R_1 \subseteq \tau(R_1)$, $R_1 \cup D \subseteq \tau(R_1) \cup D$. By the note in the preceding paragraph, we have $\tau(R_1 \cup D) \subseteq \tau(\tau(R_1) \cup D)$. Since $\tau(R_1) \cup D$ is transitive, $\tau(\tau(R_1) \cup D) = \tau(R_1) \cup D$ (Exercise 56). Therefore, $\tau(R_1 \cup D) \subseteq \tau(R_1) \cup D$.

64. A set is equivalent to itself by the identity function.

If X is equivalent to Y, there is a one-to-one, onto function f from X to Y. Now f^{-1} is a one-to-one, onto function from Y to X.

If X is equivalent to Y, there is a one-to-one, onto function f from X to Y. If Y is equivalent to Z, there is a one-to-one, onto function g from Y to Z. Now $g \circ f$ is a one-to-one, onto function from X to Z.

Section 3.5 Review

1. To obtain the matrix of a relation from X to Y, we label the rows with the elements of X and the columns with the elements of Y. We then set the entry in row x and column y to 1 if $x R y$ and to 0 otherwise.

2. A relation is reflexive if and only if its matrix has 1's on the main diagonal.

3. A relation is symmetric if and only if its matrix A satisfies the following: For all i and j, the ijth entry of A is equal to the jith entry of A.

4. See the paragraph following the proof of Theorem 3.5.6.

5. The matrix of the relation $R_2 \circ R_1$ is obtained by replacing each nonzero term in $A_1 A_2$ by 1.

Section 3.5

1.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{cccc} \alpha & \beta & \Sigma & \delta \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

4.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

8. $R = \{(a, w), (a, y), (c, y), (d, w), (d, x), (d, y), (d, z)\}$

11. The test is, whenever the ijth entry is 1, $i \neq j$, then the jith entry is *not* 1.

14. (For Exercise 8)

$$\begin{array}{c} \\ w \\ x \\ y \\ z \end{array}\begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

16. (a) $A_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}$

(b) $A_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$

(c) $A_1A_2 = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$

(d) We change each nonzero entry in part (c) to 1 to obtain

$$A_1A_2 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

(e) $\{(1, b), (1, a), (1, c), (2, b), (3, b)\}$

19. Each column that contains 1 in row x corresponds to an element of the equivalence class containing x.

21. Suppose that the ijth entry of A is 1. Then the ijth entry of either A_1 or A_2 is 1. Thus either $(i, j) \in R_1$ or $(i, j) \in R_2$. Therefore, $(i, j) \in R_1 \cup R_2$. Now suppose that $(i, j) \in R_1 \cup R_2$. Then the ijth entry of either A_1 or A_2 is 1. Therefore, the ijth entry of A is 1. It follows that A is the matrix of $R_1 \cup R_2$.

25. Each row must contain exactly one 1 for the relation to be a function.

Section 3.6 Review

1. An n-ary relation is a set of n-tuples.
2. A database management system is a program that helps users access information in a database.
3. A relational database represents data as tables and provides ways to manipulate the tables.
4. A single attribute or combination of attributes for a relation is a key if the values of the attributes uniquely define an n-tuple.
5. A query is a request for information from a database.
6. The selection operator chooses certain n-tuples from a relation. The choices are made by giving conditions on the attributes (see Example 3.6.3).
7. The project operator chooses specified columns from a relation. In addition, duplicates are eliminated (see Example 3.6.4).
8. The join operation on relations R_1 and R_2 begins by examining all pairs of tuples, one from R_1 and one from R_2. If the join condition is satisfied, the tuples are combined to form a new tuple. The join condition specifies a relationship between an attribute in R_1 and an attribute in R_2 (see Example 3.6.5).

Section 3.6

1. {(1089, Suzuki, Zamora), (5620, Kaminski, Jones), (9354, Jones, Yu), (9551, Ryan, Washington), (3600, Beaulieu, Yu), (0285, Schmidt, Jones), (6684, Manacotti, Jones)}

5. EMPLOYEE [Name]

Suzuki, Kaminski, Jones, Ryan, Beaulieu,
Schmidt, Manacotti

8. BUYER [Name]

United Supplies, ABC Unlimited, JCN Electronics,
Danny's, Underhanded Sales, DePaul University

11. TEMP := BUYER [Part No = 20A8]

TEMP [Name]

Underhanded Sales, Danny's, ABC Unlimited

14. TEMP1 := BUYER [Name = Danny's]

TEMP2 := TEMP1 [Part No = Part No] SUPPLIER

TEMP2 [Dept]

04, 96

17. TEMP1 := BUYER [Name = JCN Electronics]

TEMP2 := TEMP1 [Part No = Part No] SUPPLIER

TEMP3 := TEMP2 [Dept = Dept] DEPARTMENT

TEMP4 := TEMP3 [Manager = Manager] EMPLOYEE

TEMP4 [Name]

Kaminski, Schmidt, Manacotti

22. Let R_1 and R_2 be two n-ary relations. Suppose that the set of elements in the ith column of R_1 and the set of elements in the ith column of R_2 come from a common domain for $i = 1, \ldots, n$. The *union* of R_1 and R_2 is the n-ary relation $R_1 \cup R_2$.

TEMP1 := DEPARTMENT [Dept = 23]

TEMP2 := DEPARTMENT [Dept = 96]

TEMP3 := TEMP1 *union* TEMP2

TEMP4 := TEMP3 [Manager = Manager] EMPLOYEE

TEMP4 [Name]

Kaminski, Schmidt, Manacotti, Suzuki

Chapter 3 Self-Test

1. f is not one-to-one. f is onto.
2. $x = y = 2.3$
3. Define f from $X = \{1, 2\}$ to $\{3\}$ by $f(1) = f(2) = 3$. Define g from $\{1\}$ to X by $g(1) = 1$.

4. (a : b means "store item a in cell b.") 1 : 1, 784 : 4, 18 : 5, 329 : 6, 43 : 7, 281 : 8, 620 : 9, 1141 : 10, 31 : 11, 684 : 12

5. (a) 14

(b) 18

(c) 192

(d) $a_{n_k} = 4k$

6. $\displaystyle\sum_{k=-1}^{n-2}(n-k-2)r^{k+2}$

7. (a) $b_5 = 35,\ b_{10} = 120$

(b) $(n+1)^2 - 1$

(c) Yes

(d) No

8. (a) *ccddccccdd*

(b) *cccddccddc*

(c) 5

(d) 20

9. Reflexive, symmetric, transitive

10. Symmetric

11. $R = \{(1, 1), (2, 2), (3, 3), (4, 4), (1, 2), (2, 1), (2, 3)\}$

12. All counterexample relations are on {1, 2, 3}.

(a) False. $R = \{(1, 1)\}$.

(b) True

(c) True

(d) False. $R = \{(1, 1)\}$.

13. Yes. It is reflexive, symmetric, and transitive.

14. $[3] = \{3, 4\}$. There are two equivalence classes.

15. $\{(a, a), (b, b), (b, d), (b, e), (d, b), (d, d), (d, e), (e, b), (e, d), (e, e), (c, c)\}$

16. (a) R is reflexive because any eight-bit string has the same number of zeros as itself.

R is symmetric because, if s_1 and s_2 have the same number of zeros, then s_2 and s_1 have the same number of zeros.

To see that R is transitive, suppose that s_1 and s_2 have the same number of zeros and that s_2 and s_3 have the same number of zeros. Then s_1 and s_3 have the same number of zeros. Therefore, R is an equivalence relation.

(b) There are nine equivalence classes.

(c) 11111111, 01111111, 00111111, 00011111, 00001111, 00000111, 00000011, 00000001, 00000000

17. $\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix}$

18. $\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$

19. $\begin{pmatrix} 1 & 1 & 0 \\ 2 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$

20. $\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$

21. ASSIGNMENT [Team]

Blue Sox, Mutts, Jackalopes

22. PLAYER [Name, Age]

Johnsonbaugh, 22; Glover, 24; Battey, 18; Cage, 30; Homer, 37; Score, 22; Johnsonbaugh, 30; Singleton, 31

23. TEMP1 := PLAYER [Position = p]

TEMP2 := TEMP1 [ID Number = PID] ASSIGNMENT

TEMP2 [Team]

Mutts, Jackalopes

24. TEMP1 := PLAYER [Age $\geq$ 30]

TEMP2 := TEMP1 [ID Number = PID] ASSIGNMENT

TEMP2 [Team]

Blue Sox, Mutts

Section 4.1 Review

1. An algorithm is a step-by-step method of solving a problem.

2. Input—the algorithm receives input. Output—the algorithm produces output. Precision—the steps are precisely stated. Determinism—the intermediate results of each step of execution are unique and are determined only by the inputs and the results of the preceding steps. Finiteness—the algorithm terminates; that is, it stops after finitely many instructions have been executed. Correctness—the output produced by the algorithm is correct; that is, the algorithm correctly solves the problem. Generality—the algorithm applies to a set of inputs.

3. A trace of an algorithm is a simulation of execution of the algorithm.

4. The advantages of pseudocode over ordinary text are that pseudocode has more precision, structure, and universality. It is often readily converted to computer code.

5. An algorithm is made up of one or more pseudocode functions.

Section 4.1

2. The algorithm does not receive input (but, logically, it needs none). If some even number greater than 2 is not the sum of two prime numbers, the algorithm will stop and output "no". If every even number greater than 2 is the sum of two prime numbers, lines 2 and 3 become an infinite loop—in this case, the algorithm will not terminate and, therefore, will not produce output. The algorithm lacks precision; in order to execute line 2, we need to know how to check whether n is the sum of two primes. The algorithm does have the determinism property. The algorithm *may* lack the finiteness property. We have already noted that if every even number greater than 2 is the sum of two prime numbers (which is currently unsettled), the algorithm will not terminate. The algorithm is not general; that is, it does not apply to a set of inputs. Rather, it applies to *one* set of inputs—namely, the empty set.

5. Input: s, n

Output: *small*, the smallest value in the sequence s

```
min(s, n) {
   small = s_1
   for i = 2 to n
      if (s_i < small) // smaller value found
         small = s_i
   return small
}
```

8. Input: s, n

Output: *small* (smallest), *large* (largest)

```
small_large(s, n, small, large) {
   small = large = s_1
   for i = 2 to n {
      if (s_i < small)
         small = s_i
      if (s_i > large)
         large = s_i
   }
}
```

11. Input: s, n

Output: *sum*

```
seq_sum(s, n) {
   sum = 0
   for i = 1 to n
      sum = sum + s_i
   return sum
}
```

14. Input: s, n

Output: s (in reverse)

```
reverse(s, n) {
   i = 1
   j = n
   while (i < j) {
      swap(s_i, s_j)
      i = i + 1
      j = j - 1
   }
}
```

17. Input: A (an $n \times n$ matrix of a relation R), n

Output: true, if R is reflexive; false, if R is not reflexive

```
is_reflexive(A, n) {
   for i = 1 to n
      if (A_ii == 0)
         return false
   return true
}
```

20. Input: A (an $n \times n$ matrix of a relation R), n

Output: true, if R is antisymmetric; false, if R is not antisymmetric

```
is_reflexive(A, n) {
   for i = 1 to n - 1
      for j = i + 1 to n
         if (A_ij == 1 ∧ A_ji == 1)
            return false
   return true
}
```

23. Input: A (an $m \times k$ matrix of a relation R_1), B (a $k \times n$ matrix of a relation R_2), m, k, n

Output: C (the $m \times n$ matrix of the relation $R_2 \circ R_1$)

```
comp_relation(A, B, m, k, n, C) {
   // first compute the matrix product AB
   for i = 1 to m
      for j = 1 to n {
         C_ij = 0
         for t = 1 to k
            C_ij = C_ij + A_it B_tj
      }
   // replace each nonzero entry in C by 1
   for i = 1 to m
      for j = 1 to n
         if (C_ij > 0)
            C_ij = 1
}
```

Section 4.2 Review

1. Finding web pages containing key words is a searching problem. The programs that perform the search are called *search engines*. Finding medical records in a hospital is another searching problem. Such a search may be carried out by people or by a computer.

2. We are given text t and we want to find the first occurrence of pattern p in t or determine that p does not occur in t.

3. We use the notation in the solution to Exercise 2. Determine whether p is in t starting at index 1 in t. If so, stop. Otherwise, determine whether p is in t starting at index 2 in t. If so, stop. Continue in this way until finding p in t or determining that p cannot be in t. In the latter case, the search can be terminated when the index in t is so large that there are not enough characters remaining in t to accommodate p.

4. Sorting a sequence s means to rearrange the data so that s is in order (nonincreasing order or nondecreasing order).

5. The entries in a book's index are sorted in increasing order, thus making it easy to quickly locate an entry in the index.

6. To sort $s_1, \ldots, s_n$ using insertion sort, first insert s_2 in s_1 so that s_1, s_2 is sorted. Next, insert s_3 in s_1, s_2 so that s_1, s_2, s_3 is sorted. Continue until inserting s_n in $s_1, \ldots, s_{n-1}$ so that the entire sequence $s_1, \ldots, s_n$ is sorted.

7. The time required by an algorithm is the number of steps to termination. The space required by an algorithm is the amount of storage required by the input, local variables, and so on.

8. Knowing or being able to estimate the time and space required by an algorithm gives an indication of how the algorithm will perform for input of various sizes when run on a computer. Knowing or being able to estimate the time and space required by two or more algorithms that solve the same problem makes it possible to compare the algorithms.

9. Many practical problems are too difficult to be solved efficiently, and compromises either in generality or correctness are necessary.

10. When a randomized algorithm executes, at some points it makes random choices.

11. The requirement that the intermediate results of each step of execution be uniquely defined and depend only on the inputs and results of the preceding steps is violated.

12. To shuffle $s_1, \ldots, s_n$, first swap s_1 and a randomly chosen element in $s_1, \ldots, s_n$. Next, swap s_2 and a randomly chosen element in $s_2, \ldots, s_n$. Continue until swapping s_{n-1} and a randomly chosen element in s_{n-1}, s_n.

13. We might generate random arrangements of sequences to use as input to test or time a sorting program.

Section 4.2

1. First i and j are set to 1. The while loop then compares $t_1 \cdots t_4$ = "bala" with p = "bala". Since the comparison succeeds, the algorithm returns $i = 1$ to indicate that p was found in t starting at index 1 in t.

4. First 20 is inserted in

34

Since $20 < 34$, 34 must move one position to the right

	34

Now 20 is inserted

20	34

Since $144 > 34$, it is immediately inserted to 34's right

20	34	144

Since $55 < 144$, 144 must move one position to the right

20	34		144

Since $55 > 34$, 55 is now inserted

20	34	55	144

The sequence is now sorted.

7. Since each element is greater than or equal to the element to its left, the element is always inserted in its original position.

8. We first swap a_i and a_j, where $i = 1$ and $j = rand(1, 5) = 5$. After the swap we have

135	57	72	101	34
↑ i				↑ j

We next swap a_i and a_j, where $i = 2$ and $j = rand(2, 5) = 4$. After the swap we have

135	101	72	57	34
	↑ i		↑ j	

We next swap a_i and a_j, where $i = 3$ and $j = rand(3, 5) = 3$. The sequence is unchanged.

We next swap a_i and a_j, where $i = 4$ and $j = rand(4, 5) = 5$. After the swap we have

135	101	72	34	57
			↑ i	↑ j

11. The while loop tests whether p occurs at index i in t. If p does occur at index i in t, t_{i+j-1} will be equal to p_j for all $j = 1, \ldots, m$. Thus j becomes $m + 1$ and the algorithm returns i. If p does not occur at index i in t, t_{i+j-1} will not be equal to p_j for some j. In this case the while loop terminates (without executing return i).

Now suppose that p occurs in t and its first occurrence is at index i in t. As noted in the previous paragraph, the algorithm correctly returns i, the smallest index in t where p occurs.

If p does not occur in t, then the while loop terminates for every i and i increments in the for loop. Therefore, the for loop runs to completion, and the algorithm correctly returns 0 to indicate that p was not found in t.

14. Input: s (the sequence $s_1, \ldots, s_n$), n, and *key*

Output: i (the index of the last occurrence of *key* in s, or 0 if *key* is not in s

```
reverse_linear_search(s, n, key) {
  i = n
  while (i ≥ 1) {
    if (s_i == key)
      return i
    i = i − 1
  }
  return 0
}
```

17. We measure the time of the algorithm by counting the number of comparisons ($t_{i+j-1} == p_j$) in the while loop.

No comparisons will be made if $n - m + 1 \leq 0$. In the remainder of this solution, we assume that $n - m + 1 > 0$.

If p is in t, m comparisons must be performed to verify that p is, in fact, in t. We can guarantee that exactly m comparisons are performed if p is at index 1 in t.

If p is not in t, at least one comparison must be performed for each i. We can guarantee that exactly one comparison is performed for each i if the first character in p does not occur in t. In this case, $n - m + 1$ comparisons are made.

If $m < n - m + 1$, the best case is that p is at index 1 in t. If $n - m + 1 < m$, the best case is that the first character in p does not occur in t. If $m = n - m + 1$, either situation is the best case.

20. Input: s (the sequence $s_1, \ldots, s_n$) and n

Output: s (sorted in nondecreasing order)

```
selection_sort(s, n) {
   for i = 1 to n − 1 {
      // find smallest in s_i, ..., s_n
      small_index = i
      for j = i + 1 to n
         if (s_j < s_small_index)
            small_index = j
      swap(s_i, s_small_index)
   }
}
```

Section 4.3 Review

1. Analysis of algorithms refers to the process of deriving estimates for the time and space needed to execute algorithms.
2. The worst-case time for input of size n of an algorithm is the maximum time needed to execute the algorithm among all inputs of size n.
3. The best-case time for input of size n of an algorithm is the minimum time needed to execute the algorithm among all inputs of size n.
4. The average-case time for input of size n of an algorithm is the average time needed to execute the algorithm over some finite set of inputs all of size n.
5. $f(n) = O(g(n))$ if there exists a positive constant C_1 such that $|f(n)| \le C_1|g(n)|$ for all but finitely many positive integers n. This notation is called the big oh notation.
6. Except for constants and a finite number of exceptions, f is bounded above by g.
7. $f(n) = \Omega(g(n))$ if there exists a positive constant C_2 such that $|f(n)| \ge C_2|g(n)|$ for all but finitely many positive integers n. This notation is called the omega notation.
8. Except for constants and a finite number of exceptions, f is bounded below by g.
9. $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$. This notation is called the theta notation.
10. Except for constants and a finite number of exceptions, f is bounded above and below by g.

Section 4.3

1. $\Theta(n)$ **4.** $\Theta(n^2)$

7. $\Theta(n^2)$ **10.** $\Theta(n)$

13. $\Theta(n^2)$ **16.** $\Theta(n)$

19. $\Theta(n^2)$ **22.** $\Theta(n^3)$

25. $\Theta(n \lg n)$ **28.** $\Theta(1)$

31. When $n = 1$, we obtain

$$1 = A + B + C.$$

When $n = 2$, we obtain

$$3 = 4A + 2B + C.$$

When $n = 3$, we obtain

$$6 = 9A + 3B + C.$$

Solving this system for A, B, C, we obtain

$$A = B = \frac{1}{2}, \qquad C = 0.$$

We obtain the formula

$$1 + 2 + \cdots + n = \frac{n^2}{2} + \frac{n}{2} + 0 = \frac{n(n+1)}{2},$$

which can be proved using mathematical induction (see Section 2.4).

33. $n! = n(n-1)\cdots 2 \cdot 1 \le n \cdot n \cdots n = n^n$

36. Since $n = 2^{\lg n}$, $n^{n+1} = (2^{\lg n})^{n+1} = 2^{(n+1)\lg n}$. Thus, it suffices to show that $(n+1)\lg n \le n^2$ for all $n \ge 1$. A proof by induction shows that $n \le 2^{n-1}$ for all $n \ge 1$. Thus, $\lg n \le n - 1$ for all $n \ge 1$. Therefore,

$$(n+1)\lg n \le (n+1)(n-1) = n^2 - 1 < n^2 \quad \text{for all } n \ge 1.$$

39. Since $f(n) = O(g(n))$, there exist constants $C' > 0$ and N such that

$$f(n) \le C'g(n) \quad \text{for all } n \ge N.$$

Let

$$C = \max\{C', f(1)/g(1), f(2)/g(2), \ldots, f(N)/g(N)\}.$$

For $n \le N$,

$$f(n)/g(n) \le \max\{f(1)/g(1), f(2)/g(2), \ldots, f(N)/g(N)\} \le C.$$

For $n \ge N$,

$$f(n) \le C'g(n) \le Cg(n).$$

Therefore, $f(n) \le Cg(n)$ for all n.

42. False. If the statement were true, we would have $n^n \le C2^n$ for some constant C and for all sufficiently large n. The preceding inequality may be rewritten as

$$\left(\frac{n}{2}\right)^n \le C$$

for some constant C and for all sufficiently large n. Since $(n/2)^n$ becomes arbitrarily large as n becomes large, we cannot have $n^n \le C2^n$ for some constant C and for all sufficiently large n.

44. True

46. False. A counterexample is $f(n) = n$ and $g(n) = 2n$.

49. True

52. False. A counterexample is $f(n) = 1$ and $g(n) = 1/n$.

53. $f(n) \neq O(g(n))$ means that for every positive constant C, $|f(n)| > C|g(n)|$ for infinitely many positive integers n.

56. We first find nondecreasing positive functions f_0 and g_0 such that for infinitely many n, $f_0(n)=n^2$ and $g_0(n)=n$. This implies that $f_0(n)\neq O(g_0(n))$. Our functions also satisfy $f_0(n)=n$ and $g_0(n)=n^2$ for infinitely many n [obviously different n than those for which $f_0(n)=n^2$ and $g_0(n)=n$]. This implies that $g_0(n)\neq O(f_0(n))$. If we then set $f(n)=f_0(n)+n$ and $g(n)=g_0(n)+n$, we obtain increasing positive functions for which $f(n)\neq O(g(n))$ and $g(n)\neq O(f(n))$.

We begin by setting $f_0(2)=2$ and $g_0(2)=2^2$. Then

$$f_0(n)=n, \quad g_0(n)=n^2, \qquad \text{if } n=2.$$

Because g_0 is nondecreasing, the least n for which we may have $g_0(n)=n$ is $n=2^2$. So we define $f_0(2^2)=2^4$ and $g_0(2^2)=2^2$. Then

$$f_0(n)=n^2, \quad g_0(n)=n, \qquad \text{if } n=2^2.$$

The preceding discussion motivates defining

$$f_0(2^{2^k}) = \begin{cases} 2^{2^k} & \text{if } k \text{ is even} \\ 2^{2^{k+1}} & \text{if } k \text{ is odd} \end{cases}$$

$$g_0(2^{2^k}) = \begin{cases} 2^{2^{k+1}} & \text{if } k \text{ is even} \\ 2^{2^k} & \text{if } k \text{ is odd.} \end{cases}$$

Suppose that $n=2^{2^k}$. If k is odd, $f_0(n)=n^2$ and $g_0(n)=n$; if k is even, $f_0(n)=n$ and $g_0(n)=n^2$. Now f_0 and g_0 are defined only for $n=2^{2^k}$, but they are nondecreasing on this domain. To extend their domains to the set of positive integers, we may simply define $f_0(1)=g_0(1)=1$ and make them constant on sets of the form $\{i \mid 2^{2^k} \leq i < 2^{2^{k+1}}\}$.

60. No

62. (a) The sum of the areas of the rectangles below the curve is equal to

$$\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{n}.$$

This area is less than the area under the curve, which is equal to

$$\int_1^n \frac{1}{x}dx = \log_e n.$$

The given inequality now follows immediately.

(b) The sum of the areas of the rectangles whose bases are on the x-axis and whose tops are above the curve is equal to

$$1+\frac{1}{2}+\cdots+\frac{1}{n-1}.$$

Since this area is greater than the area under the curve, the given inequality follows immediately.

(c) Part (a) shows that

$$1+\frac{1}{2}+\cdots+\frac{1}{n}=O(\log_e n).$$

Since $\log_e n = \Theta(\lg n)$ (see Example 4.3.6),

$$1+\frac{1}{2}+\cdots+\frac{1}{n}=O(\lg n).$$

Similarly, we can conclude from part (b) that

$$1+\frac{1}{2}+\cdots+\frac{1}{n}=\Omega(\lg n).$$

Therefore,

$$1+\frac{1}{2}+\cdots+\frac{1}{n}=\Theta(\lg n).$$

64. Replacing a by b in the sum yields

$$\begin{aligned}\frac{b^{n+1}-a^{n+1}}{b-a} &= \sum_{i=0}^{n} a^i b^{n-i} < \sum_{i=0}^{n} b^i b^{n-i} \\ &= \sum_{i=0}^{n} b^n = (n+1)b^n.\end{aligned}$$

67. By Exercise 65, the sequence $\{(1+1/n)^n\}_{n=1}^{\infty}$ is increasing. Therefore

$$2=\left(1+\frac{1}{1}\right)^1 \leq \left(1+\frac{1}{n}\right)^n$$

for every positive integer n. Exercise 66 shows that

$$\left(1+\frac{1}{n}\right)^n < 4$$

for every positive integer n. Taking logs to the base 2, we obtain

$$1=\lg 2 \leq \lg\left(1+\frac{1}{n}\right)^n < \lg 4 = 2.$$

Since

$$\begin{aligned}\lg\left(1+\frac{1}{n}\right)^n = n\lg\left(1+\frac{1}{n}\right) &= n\lg\left(\frac{n+1}{n}\right) \\ &= n[\lg(n+1)-\lg n],\end{aligned}$$

we have

$$1 \leq n[\lg(n+1)-\lg n] < 2.$$

Dividing by n gives the desired inequality.

70. Replacing b by a in the sum yields

$$\begin{aligned}\frac{b^{n+1}-a^{n+1}}{b-a} &= \sum_{i=0}^{n} a^i b^{n-i} > \sum_{i=0}^{n} a^i a^{n-i} \\ &= \sum_{i=0}^{n} a^n = (n+1)a^n.\end{aligned}$$

73. By Exercise 72, the sequence $\{(1+1/n)^{n+1}\}_{n=1}^{\infty}$ is decreasing. Since $(1+1/n)^{n+1}=4$, when $n=1$,

$$4 \geq \left(1+\frac{1}{n}\right)^{n+1} = \left(\frac{n+1}{n}\right)^{n+1}.$$

Taking logs to the base 2, we obtain

$$2 = \lg 4 \geq \lg\left(\frac{n+1}{n}\right)^{n+1} = (n+1)\lg\left(\frac{n+1}{n}\right)$$
$$= (n+1)[\lg(n+1) - \lg n].$$

Dividing by $n+1$ gives the desired inequality.

75. True. Since $\lim_{n\to\infty} f(n)/g(n) = 0$, taking $\varepsilon = 1$, there exists N such that

$$\left|\frac{f(n)}{g(n)}\right| < 1, \quad \text{for all } n \geq N.$$

Therefore, for all $n \geq N$, $|f(n)| < |g(n)|$ and $f(n) = O(g(n))$.

78. True. Let $d = |c|$. Since $\lim_{n\to\infty} |f(n)|/|g(n)| = d > 0$, taking $\varepsilon = d/2$, there exists N such that

$$\left|\frac{|f(n)|}{|g(n)|} - d\right| < d/2, \quad \text{for all } n \geq N.$$

This last inequality may be written

$$-\frac{d}{2} < \frac{|f(n)|}{|g(n)|} - d < \frac{d}{2}, \quad \text{for all } n \geq N,$$

or

$$\frac{d}{2} < \frac{|f(n)|}{|g(n)|} < \frac{3d}{2}, \quad \text{for all } n \geq N,$$

or

$$\frac{d}{2}|g(n)| < |f(n)| < \frac{3d}{2}|g(n)|, \quad \text{for all } n \geq N.$$

Therefore, $f(n) = \Theta(g(n))$.

83. Multiply both sides of the inequality in Exercise 82 by $\lg e$ and use the change-of-base formula for logarithms.

Section 4.4 Review

1. An algorithm that contains a recursive function

2. A function that invokes itself

3.

```
factorial(n) {
   if (n == 0)
      return 1
   return n * factorial(n − 1)
}
```

4. The original problem is divided into two or more subproblems. Solutions are then found for the subproblems (usually by further subdivision). These solutions are then combined in order to obtain a solution to the original problem.

5. In a base case, a solution is obtained directly, that is, without a recursive call.

6. If a recursive function had no base case, it would continue to call itself and never terminate.

7. $f_1 = 1$, $f_2 = 1$, $f_n = f_{n-1} + f_{n-2}$ for $n \geq 3$

8. $f_1 = 1$, $f_2 = 1$, $f_3 = 2$, $f_4 = 3$

Section 4.4

1. (a) At line 2, since $4 \neq 0$, we proceed to line 4. The algorithm is invoked with input 3.

(b) At line 2, since $3 \neq 0$, we proceed to line 4. The algorithm is invoked with input 2.

(c) At line 2, since $2 \neq 0$, we proceed to line 4. The algorithm is invoked with input 1.

(d) At line 2, since $1 \neq 0$, we proceed to line 4. The algorithm is invoked with input 0.

(e) At lines 2 and 3, since $0 = 0$, we return 1.

Execution resumes in part (d) at line 4 after computing $0!\ (= 1)$. We return $0! \cdot 1 = 1$.

Execution resumes in part (c) at line 4 after computing $1!\ (= 1)$. We return $1! \cdot 2 = 2$.

Execution resumes in part (b) at line 4 after computing $2!\ (= 2)$. We return $2! \cdot 3 = 6$.

Execution resumes in part (a) at line 4 after computing $3!\ (= 6)$. We return $3! \cdot 4 = 24$.

4. We use induction on i, where $n = 2^i$. The Basis Step is $i = 1$. In this case, the board is a tromino T. The algorithm correctly tiles the board with T and returns. Thus the algorithm is correct for $i = 1$.

Now assume that if $n = 2^i$, the algorithm is correct. Let $n = 2^{i+1}$. The algorithm divides the board into four $(n/2) \times (n/2)$ subboards. It then places one right tromino in the center as in Figure 1.7.5. It considers each of the squares covered by the center tromino as missing. It then tiles the four subboards, and, by the inductive assumption, these subboards are correctly tiled. Therefore, the $n \times n$ board is correctly tiled. The Inductive Step is complete. The algorithm is correct.

7. The proof is by strong induction on n. The Basis Steps $(n = 1, 2)$ are readily verified.

Assume that the algorithm is correct for all $k < n$. We must show that the algorithm is correct for $n > 2$. Since $n > 2$, the algorithm executes the return statement

return $walk(n-1) + walk(n-2)$

By the inductive assumption, the values of $walk(n-1)$ and $walk(n-2)$ are correctly computed by the algorithm. Since

$$walk(n) = walk(n-1) + walk(n-2),$$

the algorithm returns the correct value of $walk(n)$.

10. (a) Input: n

Output: $2 + 4 + \cdots + 2n$

```
sum(s, n) {
   if (n == 1)
      return 2
   return sum(n − 1) + 2n
}
```

(b) **Basis Step ($n = 1$)** If n is equal to 1, we correctly return 2.

Inductive Step Assume that the algorithm correctly computes the sum when the input is $n-1$. Now suppose that the input to this algorithm is $n > 1$. At line 2, since $n \neq 1$, we proceed to line 4, where we invoke this algorithm with input $n-1$. By the inductive assumption, the value returned, $sum(n-1)$, is equal to

$$2 + \cdots + 2(n-1).$$

At line 4, we then return

$$sum(n-1) + 2n = 2 + \cdots + 2(n-1) + 2n,$$

which is the correct value.

13. Input: The sequence $s_1, \ldots, s_n$ and the length n of the sequence

Output: The maximum value in the sequence

```
find_max(s, n) {
   if (n == 1)
      return s_1
   x = find_max(s, n − 1)
   if (x > s_n)
      return x
   else
      return s_n
}
```

We prove that the algorithm is correct using induction on n. The base case is $n = 1$. If $n = 1$, the only item in the sequence is s_1 and the algorithm correctly returns it.

Assume that the algorithm computes the maximum for input of size $n-1$, and suppose that the algorithm receives input of size n. By assumption, the recursive call

$$x = find\_max(s, n-1)$$

correctly computes x as the maximum value in the sequence $s_1, \ldots, s_{n-1}$. If x is greater than s_n, the maximum value in the sequence $s_1, \ldots, s_n$ is x—the value returned by the algorithm. If x is not greater than s_n, the maximum value in the sequence $s_1, \ldots, s_n$ is s_n—again, the value returned by the algorithm. In either case, the algorithm correctly computes the maximum value in the sequence. The Inductive Step is complete, and we have proved that the algorithm is correct.

16. To list all of the ways that a robot can walk n meters, set s to the null string and invoke this algorithm.

Input: n, s (a string)

Output: All the ways the robot can walk n meters. Each method of walking n meters includes the extra string s in the list.

```
list_walk1(n, s)
   if (n == 1) {
      println(s + "take one step of length 1")
      return
   }
   if (n == 2) {
      println(s + "take two steps of length 1")
      println(s + "take one step of length 2")
      return
   }
   s' = s + "take one step of length 2" // concatenation
   list_walk1(n − 2, s')
   s' = s + "take one step of length 1" // concatenation
   list_walk1(n − 1, s')
}
```

18. After one month, there is still just one pair because a pair does not become productive until after one month. Therefore, $a_1 = 1$. After two months, the pair alive in the beginning becomes productive and adds one additional pair. Therefore, $a_2 = 2$. The increase in pairs of rabbits $a_n - a_{n-1}$ from month $n-1$ to month n is due to each pair alive in month $n-2$ producing an additional pair. That is, $a_n - a_{n-1} = a_{n-2}$. Since $\{a_n\}$ satisfies the same recurrence relation as $\{f_n\}$, $a_1 = f_2$, and $a_2 = f_3$, $a_n = f_{n+1}$, $n \geq 1$.

21. Basis Step ($n = 2$)

$$f_2^2 = 1 = 1 \cdot 2 - 1 = f_1 f_3 + (-1)^3$$

Inductive Step

$$\begin{aligned} f_n f_{n+2} + (-1)^{n+2} &= f_n(f_{n+1} + f_n) + (-1)^{n+2} \\ &= f_n f_{n+1} + f_n^2 + (-1)^{n+2} \\ &= f_n f_{n+1} + f_{n-1} f_{n+1} + (-1)^{n+1} + (-1)^{n+2} \\ &= f_{n+1}(f_n + f_{n-1}) = f_{n+1}^2 \end{aligned}$$

24. Basis Step ($n = 1$) $f_1^2 = 1^2 = 1 = 1 \cdot 1 = f_1 f_2$

Inductive Step $\displaystyle\sum_{k=1}^{n+1} f_k^2 = \sum_{k=1}^{n} f_k^2 + f_{n+1}^2 = f_n f_{n+1} + f_{n+1}^2$

$$= f_{n+1}(f_n + f_{n+1}) = f_{n+1} f_{n+2}$$

27. We use strong induction.
Basis Steps ($n = 6, 7$) $f_6 = 8 > 7.59 = (3/2)^5$. $f_7 = 13 > 11.39 = (3/2)^6$.
Inductive Step

$$f_n = f_{n-1} + f_{n-2} > \left(\frac{3}{2}\right)^{n-2} + \left(\frac{3}{2}\right)^{n-3}$$

$$= \left(\frac{3}{2}\right)^{n-1}\left[\left(\frac{3}{2}\right)^{-1} + \left(\frac{3}{2}\right)^{-2}\right] = \left(\frac{3}{2}\right)^{n-1}\left[\frac{16}{9}\right] > \left(\frac{3}{2}\right)^{n-1}$$

30. We use strong induction on n.

Basis Step ($n = 1$) $1 = f_1$

Inductive Step Suppose that $n > 2$ and that every positive integer less than n can be expressed as the sum of distinct Fibonacci numbers, no two of which are consecutive. Let f_{k_1} be the largest Fibonacci number satisfying $n \geq f_{k_1}$. If $n = f_{k_1}$, then n is trivially the sum of distinct Fibonacci numbers, no two of which are consecutive. Suppose that $n > f_{k_1}$. By the inductive assumption, $n - f_{k_1}$ can be expressed as the sum of distinct Fibonacci numbers $f_{k_2} > f_{k_3} > \cdots > f_{k_m}$, no two of which are consecutive:

$$n - f_{k_1} = \sum_{i=2}^{m} f_{k_i}.$$

Now n is expressed as the sum of Fibonacci numbers:

$$n = \sum_{i=1}^{m} f_{k_i}. \qquad (*)$$

We next show that $f_{k_1} > f_{k_2}$, so that, in particular, n is the sum of *distinct* Fibonacci numbers.

Notice that $f_{k_2} < n$. Since f_{k_1} is the largest Fibonacci number satisfying $n \geq f_{k_1}$, $f_{k_2} \leq f_{k_1}$. If $f_{k_2} = f_{k_1}$,

$$n \geq f_{k_1} + f_{k_2} > f_{k_1} + f_{k_1-1} = f_{k_1+1}.$$

This last inequality contradicts the choice of f_{k_1} as the largest Fibonacci number satisfying $n \geq f_{k_1}$. Therefore $f_{k_1} > f_{k_2}$.

The only Fibonacci numbers in the sum $(*)$ that might be consecutive are f_{k_1} and f_{k_2}. If they are consecutive, we may also write $(*)$ as

$$\begin{aligned} n &= \sum_{i=1}^{m} f_{k_i} \\ &= f_{k_1} + f_{k_2} + \sum_{i=3}^{m} f_{k_i} \\ &= f_{k_1} + f_{k_1-1} + \sum_{i=3}^{m} f_{k_i} \\ &= f_{k_1+1} + \sum_{i=3}^{m} f_{k_i}. \end{aligned}$$

Now $f_{k_1+1} \leq n$ and $f_{k_1+1} > f_{k_1}$. This contradicts the choice of f_{k_1} as the largest Fibonacci number satisfying $n \geq f_{k_1}$. The inductive step is complete.

33. Using the formula $f_k f_{k+2} - f_{k+1}^2 = (-1)^{k+1}$ from Exercise 21, we obtain

$$\begin{aligned} 1 + \sum_{k=1}^{n} \frac{(-1)^{k+1}}{f_k f_{k+1}} &= 1 + \sum_{k=1}^{n} \frac{f_k f_{k+2} - f_{k+1}^2}{f_k f_{k+1}} \\ &= 1 + \sum_{k=1}^{n} \left(\frac{f_{k+2}}{f_{k+1}} - \frac{f_{k+1}}{f_k} \right) \\ &= 1 + \left(\frac{f_3}{f_2} - \frac{f_2}{f_1} \right) + \left(\frac{f_4}{f_3} - \frac{f_3}{f_2} \right) \\ &\quad + \cdots + \left(\frac{f_{n+2}}{f_{n+1}} - \frac{f_{n+1}}{f_n} \right) \\ &= 1 + \frac{f_{n+2}}{f_{n+1}} - \frac{f_2}{f_1} = \frac{f_{n+2}}{f_{n+1}}. \end{aligned}$$

35. Basis Step ($n = 1$)

$$\frac{dx}{dx} = 1 = 1x^{1-1}$$

Inductive Step

$$\begin{aligned} \frac{dx^{n+1}}{dx} &= \frac{d(x \cdot x^n)}{dx} = x\frac{dx^n}{dx} + x^n \frac{dx}{dx} \\ &= xnx^{n-1} + x^n \cdot 1 = (n+1)x^n \end{aligned}$$

Chapter 4 Self-Test

1. At line 2, we set *large* to 12. At line 3, since $b > large$ ($3 > 12$) is false, we move to line 5. At line 5, since $c > large$ ($0 > 12$) is false, we move to line 7, where we return *large* (12), the maximum of the given values.

2.

```
sort(a, b, c, x, y, z) {
    x = a
    y = b
    z = c
    if (y < x)
        swap(x, y)
    if (z < x)
        swap(x, z)
    if (z < y)
        swap(y, z)
}
```

3.

```
test_distinct(a, b, c) {
    if (a == b ∨ a == c ∨ b == c)
        return false
    return true
}
```

4. If the set S is an infinite set, the algorithm will not terminate, so it lacks the finiteness and output properties. Line 1 is not precisely stated since *how* to list the subsets of S and their sums is not specified; thus the algorithm lacks the precision property. The order of the subsets listed in line 1 depends on the method used to generate them, so the algorithm lacks the determinism property. Since line 2 depends on the order of the subsets generated in line 1, the determinism property is lacking here as well.

5. The while loop first tests whether "110" occurs in t at index 1. Since "110" does not occur in t at index 1, the algorithm next tests whether "110" occurs in t at index 2. Since "110" does occur in t at index 2, the algorithm returns the value 2.

6. First 64 is inserted in

44

Since $64 > 44$, it is immediately inserted to 44's right

44	64

Next 77 is inserted. Since $77 > 64$, it is immediately inserted to 64's right

44	64	77

Next 15 is inserted. Since $15 < 77$, 77 must move one position to the right

44	64		77

Since $15 < 64$, 64 must move one position to the right

44		64	77

Since $15 < 44$, 44 must move one position to the right

	44	64	77

Now 15 is inserted

15	44	64	77

Finally 3 is inserted. Since $3 < 77$, 77 must move one position to the right

15	44	64		77

Since $3 < 64$, 64 must move one position to the right

15	44		64	77

Since $3 < 44$, 44 must move one position to the right

15		44	64	77

Since $3 < 15$, 15 must move one position to the right

	15	44	64	77

Now 3 is inserted

3	15	44	64	77

The sequence is sorted.

7. We first swap a_i and a_j, where $i = 1$ and $j = rand(1, 5) = 1$. The sequence is unchanged.

We next swap a_i and a_j, where $i = 2$ and $j = rand(2, 5) = 3$. After the swap we have

5	2	51	44	96
	↑ i	↑ j		

We next swap a_i and a_j, where $i = 3$ and $j = rand(3, 5) = 5$. After the swap we have

5	2	96	44	51
		↑ i		↑ j

We next swap a_i and a_j, where $i = 4$ and $j = rand(4, 5) = 5$. After the swap we have

5	2	96	51	44
			↑ i	↑ j

8.

```
repeaters(s, n) {
   i = 1
   while (i < n) {
      if (s_i == s_{i+1})
         println(s_i)
      // skip to next element not equal to s_i
      j = i
      while (i < n ∧ s_i == s_j)
         i = i + 1
   }
}
```

9. $\Theta(n^3)$ **10.** $\Theta(n^4)$ **11.** $\Theta(n^2)$

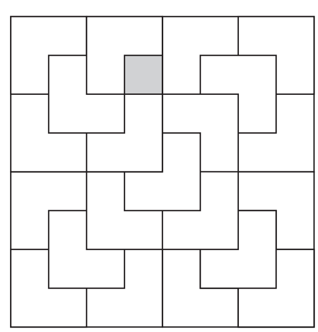
12. Input: A and B ($n \times n$ matrices) and n

Output: true (if $A = B$); false (if $A \neq B$)

```
equal_matrices(A, B, n) {
   for i = 1 to n
      for j = 1 to n
         if (A_ij ¬= B_ij)
            return false
   return true
}
```

The worst-case time is $\Theta(n^2)$.

13. Since $n \neq 2$, we proceed immediately to line 6, where we divide the board into four 4×4 boards. At line 7, we rotate the board so that the missing square is in the upper-left quadrant. At line 8, we place one tromino in the center. We then proceed to lines 9–12, where we call the algorithm to tile the subboards. We obtain the tiling:



14. $t_4 = 3$, $t_5 = 5$

15. Input: n, an integer greater than or equal to 1

Output: t_n

```
   tribonacci(n) {
1.    if (n == 1 ∨ n == 2 ∨ n == 3)
2.       return 1
3.    return tribonacci(n − 1) + tribonacci(n − 2)
         + tribonacci(n − 3)
   }
```

16. **Basis Steps ($n = 1, 2, 3$)** If $n = 1, 2, 3$, at lines 1 and 2 we return the correct value, 1. Therefore, the algorithm is correct in these cases.

Inductive Step Assume that $n > 3$ and that the algorithm correctly computes t_k, if $k < n$. Since $n > 3$, we proceed to line 3. We then call this algorithm to compute t_{n-1}, t_{n-2}, and t_{n-3}. By the inductive assumption, the values computed are correct. The algorithm then computes $t_{n-1} + t_{n-2} + t_{n-3}$. But the formula shows that this value is equal to t_n. Therefore, the algorithm returns the correct value for t_n.

Section 5.1 Review

1. We say that d divides n if there exists an integer q satisfying $n = dq$.
2. If d divides n, we say that d is a divisor of n.
3. If d divides n, $n = dq$, we call q the quotient.
4. An integer greater than 1 whose only positive divisors are itself and 1 is called prime.
5. An integer greater than 1 that is not prime is called composite.
6. If n is composite, it must have a divisor d satisfying $2 \le d \le \lfloor\sqrt{n}\rfloor$ (see Theorem 5.1.7).
7. Algorithm 5.1.8 does not run in time polynomial in the *size* of the input.
8. Any integer greater than 1 can be written as a product of primes. Moreover, if the primes are written in nondecreasing order, the factorization is unique.
9. See the proof of Theorem 5.1.12.
10. A common divisor of m and n, not both zero, is an integer that divides both m and n.
11. The greatest common divisor of m and n, not both zero, is the largest common divisor of m and n.
12. See Theorem 5.1.17.
13. A common multiple of m and n is an integer that is divisible by both m and n.
14. The least common multiple of m and n is the smallest positive common multiple of m and n.
15. See Theorem 5.1.22.
16. $\gcd(m, n) \cdot \operatorname{lcm}(m, n) = mn$

Section 5.1

1. First d is set to 2. Since $n \bmod d = 9 \bmod 2 = 1$ is not equal to 0, d is incremented and becomes 3.

Now $n \bmod d = 9 \bmod 3$ equals 0, so the algorithm returns $d = 3$ to indicate that $n = 9$ is composite and 3 is a divisor of 9.

4. When d is set to $2, \ldots, 6$, $n \bmod d$ is not equal to zero. However, when d becomes 7, $n \bmod d = 637 \bmod 7$ equals 0, so the algorithm returns $d = 7$ to indicate that $n = 637$ is composite and 7 is a divisor of 637.

7. First d is set to 2. Since $n \bmod d = 3738 \bmod 2$ equals 0, the algorithm returns $d = 2$ to indicate that $n = 3738$ is composite and 2 is a divisor of 3738.

9. 47

12. 17

15. 1

18. 20

21. 13

24. $3^2 \cdot 7^3 \cdot 11$

25. (For Exercise 13) 25

28. Since d divides m, there exists q such that $m = dq$. Multiplying by n gives $mn = d(qn)$. Therefore, d divides mn (with quotient qn).

31. Since a divides b, there exists q_1 such that $b = aq_1$. Since b divides c, there exists q_2 such that $c = bq_2$. Now

$$c = bq_2 = (aq_1)q_2 = a(q_1q_2).$$

Therefore, a divides c (with quotient q_1q_2).

Section 5.2 Review

1. $\displaystyle\sum_{i=0}^{n} d_i 10^i$
2. $\displaystyle\sum_{i=0}^{n} b_i 2^i$
3. $\displaystyle\sum_{i=0}^{n} h_i 16^i$
4. $\lfloor 1 + \lg n \rfloor$
5. Perform the computation $\sum_{i=0}^{n} b_i 2^i$ in decimal.
6. Divide the number to be converted to binary by 2. The remainder gives the 1's bit. Divide the quotient by 2. The remainder gives the 2's bit. Continue.
7. Perform the computation $\sum_{i=0}^{n} h_i 16^i$ in decimal.
8. Divide the number to be converted to hexadecimal by 16. The remainder gives the number of 1's. Divide the quotient by 16. The remainder gives the number of 16's. Continue.
9. Use the ordinary algorithm for adding decimal numbers to add binary numbers—except replace the decimal addition table by the binary addition table.
10. Use the ordinary algorithm for adding decimal numbers to add hexadecimal numbers—except replace the decimal addition table by the hexadecimal addition table.
11. Let

$$n = \sum_{i=0}^{m} b_i 2^i$$

be the binary expansion of n. Using repeated squaring, compute $a^1, a^2, a^4, a^8, \ldots, a^{b_m}$. Then

$$a^n = a^{\sum_{i=0}^{m} b_i 2^i} = \prod_{i=0}^{m} a^{b_i 2^i}.$$

12. Proceed as described in the solution to Exercise 11, only use the formula

$$ab \bmod z = [(a \bmod z)(b \bmod z)] \bmod z.$$

Section 5.2

1. 6	**4.** 7	**7.** 1585
8. 9	**11.** 32	**14.** 100010
17. 110010000	**20.** 11000	**23.** 1001000
26. 58	**29.** 2563	

32. (For Exercise 8) 9

35. FE **38.** 3DBF9

40. 2010 cannot represent a number in binary because 2 is an illegal symbol in binary. 2010 could represent a number in either decimal or hexadecimal.

42. 51 **45.** 4570

48. (For Exercise 8) 11 **51.** (For Exercise 42) 33

54. 9450 cannot represent a number in binary because 9, 4, and 5 are illegal symbols in binary. 9450 cannot represent a number in octal because 9 is an illegal symbol in octal. 9450 represents a number in either decimal or hexadecimal.

56. The algorithm begins by setting *result* to 1 and x to a. Since $n = 16 > 0$, the body of the while loop executes. Since $n \bmod 2$ is not equal to 1, *result* is not modified. x becomes a^2, and n becomes 8.

Since $n = 8 > 0$, the body of the while loop executes. Since $n \bmod 2$ is not equal to 1, *result* is not modified. x becomes a^4, and n becomes 4.

Since $n = 4 > 0$, the body of the while loop executes. Since $n \bmod 2$ is not equal to 1, *result* is not modified. x becomes a^8, and n becomes 2.

Since $n = 2 > 0$, the body of the while loop executes. Since $n \bmod 2$ is not equal to 1, *result* is not modified. x becomes a^{16}, and n becomes 1.

Since $n = 1 > 0$, the body of the while loop executes. Since $n \bmod 2$ is equal to 1, *result* becomes $result * x = 1 * a^{16} = a^{16}$. x becomes a^{32}, and n becomes 0.

Since $n = 0$ is not greater than 0, the while loop terminates. The algorithm returns *result*, which is equal to a^{16}.

59. The algorithm begins by setting *result* to 1 and x to $a \bmod z = 5 \bmod 21 = 5$. Since $n = 10 > 0$, the body of the while loop executes. Since $n \bmod 2$ is not equal to 1, *result* is not modified. x is set to $x * x \bmod z = 25 \bmod 21 = 4$, and n is set to 5.

Since $n = 5 > 0$, the body of the while loop executes. Since $n \bmod 2$ is equal to 1, *result* is set to $(result * x) \bmod z = 4 \bmod 21 = 4$. x is set to $x * x \bmod z = 16 \bmod 21 = 16$, and n is set to 2.

Since $n = 2 > 0$, the body of the while loop executes. Since $n \bmod 2$ is not equal to 1, *result* is not modified. x is set to $x * x \bmod z = 256 \bmod 21 = 4$, and n is set to 1.

Since $n = 1 > 0$, the body of the while loop executes. Since $n \bmod 1$ is equal to 1, *result* is set to $(result * x) \bmod z = 16 \bmod 21 = 16$. x is set to $x * x \bmod z = 16 \bmod 21 = 16$, and n is set to 0.

Since $n = 0$ is not greater than 0, the while loop terminates. The algorithm returns *result*, which is equal to $a^n \bmod z = 5^{10} \bmod 21 = 16$.

62. If m_k is the highest power of 2 that divides m, then $m = 2^{m_k} p$, where p is odd. Similarly, if n_k is the highest power of 2 that divides n, then $n = 2^{n_k} q$, where q is odd. Now $mn = 2^{m_k + n_k} pq$. Since pq is odd, $m_k + n_k$ is the highest power of 2 that divides mn, and the result follows.

Section 5.3 Review

1. See Algorithm 5.3.3.

2. If a is a nonnegative integer, b is a positive integer, and $r = a \bmod b$, then $\gcd(a, b) = \gcd(b, r)$.

3. $a \geq f_{n+2}$ and $b \geq f_{n+1}$

4. $\log_{3/2} 2m/3$

5. Write the nonzero remainders as found by the Euclidean algorithm in the form

$$r = n - dq$$

in the order in which the Euclidean algorithm computes them. Substitute the formula for the next-to-last remainder into the last equation. Call the resulting equation E_1. Substitute the second-to-last formula for the remainder into E_1. Call the resulting equation E_2. Substitute the third-to-last formula for the remainder into E_2. Continue until the first formula for the remainder is substituted into the last E_k equation.

6. s is the inverse of $n \bmod z$ if $ns \bmod z = 1$.

7. Find numberss' and t' such that $s'n + t'\phi = 1$. Set $s = s' \bmod \phi$.

Section 5.3

1. $90 \bmod 60 = 30$; $60 \bmod 30 = 0$; so $\gcd(60, 90) = 30$.

4. $825 \bmod 315 = 195$; $315 \bmod 195 = 120$; $195 \bmod 120 = 75$; $120 \bmod 75 = 45$; $75 \bmod 45 = 30$; $45 \bmod 30 = 15$; $30 \bmod 15 = 0$; so $\gcd(825, 315) = 15$.

7. $4807 \bmod 2091 = 625$; $2091 \bmod 625 = 216$; $625 \bmod 216 = 193$; $216 \bmod 193 = 23$; $193 \bmod 23 = 9$; $23 \bmod 9 = 5$; $9 \bmod 5 = 4$; $5 \bmod 4 = 1$; $4 \bmod 1 = 0$; so $\gcd(2091, 4807) = 1$.

10. $490256 \bmod 337 = 258$; $337 \bmod 258 = 79$; $258 \bmod 79 = 21$; $79 \bmod 21 = 16$; $21 \bmod 16 = 5$; $16 \bmod 5 = 1$; $5 \bmod 1 = 0$; so $\gcd(490256, 337) = 1$.

11. (For Exercise 10) The nonzero remainders in the order they are computed by the Euclidean algorithm are

$$490256 \bmod 337 = 258$$
$$337 \bmod 258 = 79$$
$$258 \bmod 79 = 21$$
$$79 \bmod 21 = 16$$
$$21 \bmod 16 = 5$$
$$16 \bmod 5 = 1.$$

Writing these equations in the form $r = n - dq$, where r is the remainder and q is the quotient, yields

$$\begin{aligned}258 &= 490256 - 337 \cdot 1454\\ 79 &= 337 - 258 \cdot 1\\ 21 &= 258 - 79 \cdot 3\\ 16 &= 79 - 21 \cdot 3\\ 5 &= 21 - 16 \cdot 1\\ 1 &= 16 - 5 \cdot 3.\end{aligned}$$

Substituting the next-to-last formula for 5 into the last equation yields

$$1 = 16 - (21 - 16 \cdot 1) \cdot 3 = 16 \cdot 4 - 21 \cdot 3.$$

Substituting the second-to-last formula for 16 into the previous equation yields

$$1 = (79 - 21 \cdot 3)4 - 21 \cdot 3 = 79 \cdot 4 - 21 \cdot 15.$$

Substituting the third formula for 21 into the previous equation yields

$$1 = 79 \cdot 4 - (258 - 79 \cdot 3)15 = 79 \cdot 49 - 258 \cdot 15.$$

Substituting the second formula for 79 into the previous equation yields

$$1 = (337 - 258)49 - 258 \cdot 15 = 337 \cdot 49 - 258 \cdot 64.$$

Finally, substituting the first formula for 258 into the previous equation yields

$$\begin{aligned}1 &= 337 \cdot 49 - (490256 - 337 \cdot 1454)64\\ &= 337 \cdot 93105 - 490256 \cdot 64.\end{aligned}$$

Thus, if we set $s = -64$ and $t = 93105$,

$$s \cdot 490256 + t \cdot 337 = \gcd(490256, 337) = 1.$$

14.
```
gcd_recurs(a, b) {
   make a largest
   if (a < b)
      swap(a, b)
   return gcd_recurs1(a, b)
}

gcd_recurs1(a, b) {
   if (b == 0)
      return a
   r = a mod b
   return gcd_recurs1(b, r)
}
```

17.
```
gcd_subtract(a, b) {
   while (true) {
      // make a largest
      if (a < b)
         swap(a, b)
      if (b == 0)
         return a
      a = a − b
   }
}
```

20. By Theorem 5.3.5, a pair a, b, $a > b$, would require n modulus operations when input to the Euclidean Algorithm only if $a \geq f_{n+2}$ and $b \geq f_{n+1}$. Now $f_{29} = 514229$, $f_{30} = 832040$, and $f_{31} = 1346269$. Thus, no pair can require more than 28 modulus operations in the worst case because 29 modulus operations would require one member of the pair to exceed 1000000. The pair 514229, 832040 itself requires 28 modulus operations.

23. We prove the statement by induction on n.

Basis Step ($n = 1$) $\gcd(f_1, f_2) = \gcd(1, 1) = 1$

Inductive Step Assume that $\gcd(f_n, f_{n+1}) = 1$. Now

$$\gcd(f_{n+1}, f_{n+2}) = \gcd(f_{n+1}, f_{n+1}+f_n) = \gcd(f_{n+1}, f_n) = 1.$$

We use Exercise 16 with $a = f_{n+1} + f_n$ and $b = f_{n+1}$ to justify the second equality.

27. If $m = 1$, the result is immediate, so we assume that $m > 1$.

Suppose that f is one-to-one and onto. Since $m > 1$, there exists x such that $f(x) = nx \bmod m = 1$. Thus there exists q such that

$$nx = mq + 1.$$

Let g be the greatest common divisor of m and n. Then g divides both m and n and also $nx - mq = 1$. Therefore, $g = 1$.

Now suppose that $\gcd(m, n) = 1$. By Theorem 5.3.7, there exist s and t such that

$$1 = sm + tn.$$

Let $k \in X$. Then

$$k = msk + ntk.$$

Therefore,

$$(ntk) \bmod m = (k - msk) \bmod m = k \bmod m = k.$$

We may argue as in the Computing an Inverse Modulo an Integer subsection that if we set $x = tk \bmod m$, then $f(x) = (ntk) \bmod m$. Therefore, f is onto. Since f is a function from X to X, f is also one-to-one.

28. If $a \neq 0$, $a = 1 \cdot a + 0 \cdot b > 0$. In this case, $a \in X$. Similarly, if $b \neq 0$, $b \in X$.

31. Suppose that g does not divide a. Then $a = qg + r$, $0 < r < g$. Since $g \in X$, there exist s and t such that $g = sa + tb$. Now

$$r = a - qg = a - q(sa + tb) = (1 - qs)a + (-qt)b.$$

Therefore $r \in X$. Since g is the least element in X and $0 < r < g$, we have a contradiction. Therefore, g divides a. Similarly, g divides b.

33. $\gcd(3, 2) = \gcd(2, 1) = \gcd(1, 0) = 1$, $s = 2$

36. $\gcd(47, 11) = \gcd(11, 3) = \gcd(3, 2) = \gcd(2, 1) = \gcd(1, 0) = 1$, $s = 30$

39. $\gcd(243, 100) = \gcd(100, 43) = \gcd(43, 14) = \gcd(14, 1) = \gcd(1, 0) = 1$, $s = 226$

40. We argue by contradiction. Suppose that 6 has an inverse modulo 15; that is, suppose that there exists s such that $6s \bmod 15 = 1$. Then there exists q such that

$$15 - 6sq = 1.$$

Since 3 divides 15 and 3 divides $6sq$, 3 divides 1. We have obtained the desired contradiction. Thus, 6 does not have an inverse modulo 15.

That 6 does not have an inverse modulo 15 does not contradict the result preceding Example 5.3.9. In order to guarantee that n has an inverse modulo ϕ, the result preceding Example 5.3.9 requires that $\gcd(n, \phi) = 1$. In this exercise, $\gcd(6, 15) = 3$.

Section 5.4 Review

1. Cryptology is the study of systems for secure communications.
2. A cryptosystem is a system for secure communications.
3. To encrypt a message is to transform the message so that only an authorized recipient can reconstruct it.
4. To decrypt a message is to transform an encrypted message so that it can be read.
5. Compute $c = a^n \bmod z$ and send c.
6. Compute $c^s \bmod z$. z is chosen as the product of primes p and q. s satisfies $ns \bmod (p-1)(q-1) = 1$.
7. The security of the RSA encryption system relies mainly on the fact that at present there is no efficient algorithm known for factoring integers.

Section 5.4

1. FKKGEJAIMWQ
4. BUSHWHACKED
7. $z = pq = 17 \cdot 23 = 391$
10. $c = a^n \bmod z = 101^{31} \bmod 391 = 186$
12. $z = pq = 59 \cdot 101 = 5959$
15. $c = a^n \bmod z = 584^{41} \bmod 5959 = 3237$

Chapter 5 Self-Test

1. For $d = 2, \ldots, 6$, 539 mod d is not equal to zero, so d increments. When $d = 7$, 539 mod d equals zero, so the algorithm returns $d = 7$ to indicate that 539 is composite and 7 is a divisor of 539.
2. $539 = 7^2 \cdot 11$
3. $7^2 \cdot 13^2$
4. $2 \cdot 5^2 \cdot 7^4 \cdot 13^4 \cdot 17$
5. 150
6. 110101110, 1AE
7. The algorithm begins by setting *result* to 1 and x to a. Since $n = 30 > 0$, the body of the while loop executes. Since n mod 2 is not equal to 1, *result* is not modified. x becomes a^2, and n becomes 15.

   Since $n = 15 > 0$, the body of the while loop executes. Since n mod 2 is equal to 1, *result* becomes $result * x = 1 * a^2 = a^2$. x becomes a^4, and n becomes 7.

   Since $n = 7 > 0$, the body of the while loop executes. Since n mod 2 is equal to 1, *result* becomes $result * x = a^2 * a^4 = a^6$. x becomes a^8, and n becomes 3.

   Since $n = 3 > 0$, the body of the while loop executes. Since n mod 2 is equal to 1, *result* becomes $result * x = a^6 * a^8 = a^{14}$. x becomes a^{16}, and n becomes 1.

   Since $n = 1 > 0$, the body of the while loop executes. Since n mod 2 is equal to 1, *result* becomes $result * x = a^{14} * a^{16} = a^{30}$. x becomes a^{32}, and n becomes 0.

   Since $n = 0$ is not greater than 0, the while loop terminates. The algorithm returns *result*, which is equal to a^{30}.
8. The algorithm begins by setting *result* to 1 and x to $a \bmod z = 50 \bmod 11 = 6$. Since $n = 30 > 0$, the body of the while loop executes. Since n mod 2 is not equal to 1, *result* is not modified. x is set to $x * x \bmod z = 36 \bmod 11 = 3$, and n is set to 15.

   Since $n = 15 > 0$, the body of the while loop executes. Since n mod 2 is equal to 1, *result* is set to $(result * x) \bmod z = 3 \bmod 11 = 3$. x is set to $x * x \bmod z = 9 \bmod 11 = 9$, and n is set to 7.

   Since $n = 7 > 0$, the body of the while loop executes. Since n mod 2 is equal to 1, *result* is set to $(result * x) \bmod z = 27 \bmod 11 = 5$. x is set to $x * x \bmod z = 81 \bmod 11 = 4$, and n is set to 3.

   Since $n = 3 > 0$, the body of the while loop executes. Since n mod 2 is equal to 1, *result* is set to $(result * x) \bmod z = 20 \bmod 11 = 9$. x is set to $x * x \bmod z = 16 \bmod 11 = 5$, and n is set to 1.

   Since $n = 1 > 0$, the body of the while loop executes. Since n mod 2 is equal to 1, *result* is set to $(result * x) \bmod z = 45 \bmod 11 = 1$. x is set to $x * x \bmod z = 25 \bmod 11 = 3$, and n is set to 0.

   Since $n = 0$ is not greater than 0, the while loop terminates. The algorithm returns *result*, which is equal to $a^n \bmod z = 50^{30} \bmod 11 = 1$.
9. $\gcd(480, 396) = \gcd(396, 84) = \gcd(84, 60) = \gcd(60, 24) = \gcd(24, 12) = \gcd(12, 0) = 12$
10. Since

$$\begin{aligned}\log_{3/2} \frac{2(100{,}000{,}000)}{3} &= \log_{3/2} 100^4 + \log_{3/2} \frac{2}{3}\\ &= (4 \log_{3/2} 100) - 1\\ &= 4(11.357747) - 1 = 44.430988,\end{aligned}$$

   an upper bound for the number of modulus operations required by the Euclidean algorithm for integers in the range 0 to 100,000,000 is 44.
11. The nonzero remainders in the order they are computed by the Euclidean algorithm are

$$\begin{aligned}480 \bmod 396 &= 84\\ 396 \bmod 84 &= 60\\ 84 \bmod 60 &= 24\\ 60 \bmod 24 &= 12.\end{aligned}$$

   Writing these equations in the form $r = n - dq$, where r is the remainder and q is the quotient, yields

$$\begin{aligned}84 &= 480 - 396 \cdot 1\\ 60 &= 396 - 84 \cdot 4\\ 24 &= 84 - 60 \cdot 1\\ 12 &= 60 - 24 \cdot 2.\end{aligned}$$

Substituting the next-to-last formula for 24 into the last equation yields

$$12 = 60 - 24 \cdot 2 = 60 - (84 - 60) \cdot 2 = 3 \cdot 60 - 2 \cdot 84.$$

Substituting the second formula for 60 into the previous equation yields

$$12 = 3 \cdot (396 - 84 \cdot 4) - 2 \cdot 84 = 3 \cdot 396 - 14 \cdot 84.$$

Finally, substituting the first formula for 84 into the previous equation yields

$$12 = 3 \cdot 396 - 14 \cdot (480 - 396) = 17 \cdot 396 - 14 \cdot 480.$$

Thus, if we set $s = 17$ and $t = -14$,

$$s \cdot 396 + t \cdot 480 = \gcd(396, 480) = 12.$$

12. The nonzero remainders in the order they are computed by the Euclidean algorithm are

$$\begin{aligned} 425 \bmod 196 &= 33 \\ 196 \bmod 33 &= 31 \\ 33 \bmod 31 &= 2 \\ 31 \bmod 2 &= 1. \end{aligned}$$

Writing these equations in the form $r = n - dq$, where r is the remainder and q is the quotient, yields

$$\begin{aligned} 33 &= 425 - 196 \cdot 2 \\ 31 &= 196 - 33 \cdot 5 \\ 2 &= 33 - 31 \cdot 1 \\ 1 &= 31 - 2 \cdot 15. \end{aligned}$$

Substituting the next-to-last formula for 2 into the last equation yields

$$1 = 31 - (33 - 31) \cdot 15 = 16 \cdot 31 - 15 \cdot 33.$$

Substituting the second formula for 31 into the previous equation yields

$$1 = 16 \cdot (196 - 33 \cdot 5) - 15 \cdot 33 = 16 \cdot 196 - 95 \cdot 33.$$

Finally, substituting the first formula for 33 into the previous equation yields

$$1 = 16 \cdot 196 - 95 \cdot (425 - 196 \cdot 2) = 206 \cdot 196 - 95 \cdot 425.$$

Thus, if we set $s' = 206$ and $t' = -95$,

$$s' \cdot 196 + t' \cdot 425 = \gcd(196, 425) = 1.$$

Thus $s = s' \bmod 425 = 206 \bmod 425 = 206$.

13. $z = pq = 13 \cdot 17 = 221$, $\phi = (p-1)(q-1) = 12 \cdot 16 = 192$

14. $s = 91$

15. $c = a^n \bmod z = 144^{19} \bmod 221 = 53$

16. $a = c^s \bmod z = 28^{91} \bmod 221 = 63$

Section 6.1 Review

1. If an activity can be constructed in t successive steps and step 1 can be done in n_1 ways, step 2 can be done in n_2 ways, ..., and step t can be done in n_t ways, then the number of different possible activities is $n_1 \cdot n_2 \cdots n_t$. As an example, if there are two choices for an appetizer and four choices for a main dish, the total number of dinners is $2 \cdot 4 = 8$.

2. Suppose that $X_1, \ldots, X_t$ are sets and that the ith set X_i has n_i elements. If $\{X_1, \ldots, X_t\}$ is a pairwise disjoint family, the number of possible elements that can be selected from X_1 or X_2 or ... or X_t is $n_1 + n_2 + \cdots + n_t$. As an example, suppose that within a set of strings, two start with a and four start with b. Then $2 + 4 = 6$ start with either a or b.

3. $|X \cup Y| = |X| + |Y| - |X \cap Y|$. An example of the use of the Inclusion-Exclusion Principle for two sets is provided by Example 6.1.13.

Section 6.1

1. $2 \cdot 4$ **4.** $8 \cdot 4 \cdot 5$ **7.** 6^2

10. $6 + 12 + 9$ **13.** $m + n$ **16.** $1 + 1$

19. Since there are three kinds of cabs, two kinds of cargo beds, and five kinds of engines, the correct number of ways to personalize the big pickups is $3 \cdot 2 \cdot 5 = 30$, not 32.

20. 3: (1, 3), (2, 2), (3, 1), where (b, r) means the blue die shows b and the red die shows r.

23. 6: (2, 1), (2, 2), (2, 3), (2, 4), (2, 5), (2, 6), where (b, r) means the blue die shows b and the red die shows r.

26. Since each die can show any one of five values, by the Multiplication Principle there are $5 \cdot 5$ outcomes in which neither die shows 2.

28. $10 \cdot 5$ **31.** 2^4 **34.** 8

37. 2^4. (Once the first four bits are assigned values, the last four bits are determined.)

38. $5 \cdot 4 \cdot 3$ **41.** $3 \cdot 4 \cdot 3$ **44.** 5^3

47. $4 \cdot 3$ **50.** $5^3 - 4^3$ **52.** $200 - 5 + 1$

55. 40

58. One one-digit number contains 7. The distinct two-digit numbers that contain 7 are 17, 27, ..., 97 and 70, 71, ..., 76, 78, 79. There are 18 of these. The distinct three-digit numbers that contain 7 are 107 and $1xy$, where xy is one of the two-digit numbers listed above. The answer is $1 + 18 + 19$.

61. $5 + (8 + 7 + \cdots + 1) + (7 + 6 + \cdots + 1)$

64. $10!$ **67.** $(3!)(5!)(2!)(3!)$

71. 2^{10} **74.** $2^{14}(2^{16} - 2)$

77. We count the number of $n \times n$ matrices that represent symmetric relations on an n-element set. Example 6.1.7 showed that there are $n^2 - n$ entries off the main diagonal. Of these, half, $(n^2 - n)/2$, are above the main diagonal. Since there are n entries on the main diagonal, there are

$$\frac{n^2 - n}{2} + n = \frac{n^2 + n}{2}$$

entries on or above the main diagonal. These entries can be assigned values arbitrarily and there are $2^{(n^2+n)/2}$ ways to assign these values. Because the relation is symmetric, once these values are assigned, the values below the main diagonal are determined. (Entry ij is 1 if and only if entry ji is 1.) Therefore there are $2^{(n^2+n)/2}$ symmetric relations on an n-element set.

80. We count the number of $n \times n$ matrices that represent reflexive and antisymmetric relations on an n-element set.

Since the relation is reflexive, the main diagonal must consist of 1's. For i and j satisfying $1 \le i < j \le n$, we can assign the entries in row i, column j and row j, column i in three ways:

Row i, Column j	*Row j, Column i*
0	0
1	0
0	1

Since there are $(n^2 - n)/2$ values of i and j satisfying $1 \le i < j \le n$, we can assign the off-diagonal values in $3^{(n^2-n)/2}$ ways. Therefore there are $3^{(n^2-n)/2}$ reflexive and antisymmetric relations on an n-element set.

83. The truth table of an n-variable function has 2^n rows since each variable can be either T or F. Each row of the table can assign the function the value T or F. Since there are 2^n rows, the function value assignments can be made in 2^{2^n} ways. Therefore there are 2^{2^n} truth tables for an n-variable function.

86. By the Inclusion-Exclusion Principle, the total number of possibilities = number of strings that begin 100 + number of strings that have the fourth bit 1 − number of strings that begin 100 and have the fourth bit 1, so the answer is $2^5 + 2^7 - 2^4$.

89. By the Inclusion-Exclusion Principle, the total number of possibilities = number in which Connie is chairperson + number in which Alice is an officer − number in which Connie is chairperson and Alice is an officer, so the answer is $5 \cdot 4 + 3 \cdot 5 \cdot 4 - 2 \cdot 4$.

93. Let F be the set of students taking French, let B be the set of students taking business, and let M be the set of students taking music. We are given that $|F \cap B \cap M| = 10$, $|F \cap B| = 36$, $|F \cap M| = 20$, $|B \cap M| = 18$, $|F| = 65$, $|B| = 76$, and $|M| = 63$. By Exercise 92,

$$\begin{aligned} |F \cup B \cup M| &= |F| + |B| + |M| - |F \cap B| - |F \cap M| \\ &\quad - |B \cap M| + |F \cap B \cap M| \\ &= 65 + 76 + 63 - 36 - 20 - 18 + 10 = 140. \end{aligned}$$

Thus 140 students are taking French or business or music. Since there are 191 students, $191 - 140 = 51$ are not any of the three courses.

96. Let X be the set of integers between 1 and 10,000 that are multiples of 3, let Y be the set of integers between 1 and 10,000 that are multiples of 5, and let Z be the set of integers between 1 and 10,000 that are multiples of 11.

A multiple of 3 is of the form $3k$ for some integer k, so a multiple of 3 between 1 and 10,000 satisfies

$$1 \le 3k \le 10{,}000.$$

Dividing by 3, we obtain

$$0.333\ldots = \frac{1}{3} \le k \le \frac{10{,}000}{3} = 3333.333\ldots.$$

Thus the multiples of 3 between 1 and 10,000 correspond to the values $k = 1, 2, \ldots, 3333$. Therefore there are 3333 multiples of 3 between 1 and 10,000. Similarly, there are 2000 multiples of 5 between 1 and 10,000 and 909 multiples of 11 between 1 and 10,000. Therefore $|X| = 3333$, $|Y| = 2000$, and $|Z| = 909$.

A number that is multiple of 3 and 5 is a multiple of 15. Arguing as in the previous paragraph, we find that there are 666 multiples of 3 and 5 between 1 and 10,000. Similarly, there are 303 multiples of 3 and 11 between 1 and 10,000, there are 181 multiples of 5 and 11 between 1 and 10,000, and there are 60 multiples of 3, 5, and 11 between 1 and 10,000. Therefore $|X \cap Y| = 666$, $|X \cap Z| = 303$, $|Y \cap Z| = 181$, and $|X \cap Y \cap Z| = 60$. By Exercise 92,

$$\begin{aligned} |X \cup Y \cup Z| &= |X| + |Y| + |Z| - |X \cap Y| - |X \cap Z| \\ &\quad - |Y \cap Z| + |X \cap Y \cap Z| \\ &= 3333 + 2000 + 909 - 666 - 303 - 181 \\ &\quad + 60 = 5152. \end{aligned}$$

Therefore there are 5152 integers between 1 and 10,000 that are multiples of 3 or 5 or 11 or any combination thereof.

Section 6.2 Review

1. An ordering of $x_1, \ldots, x_n$

2. There are $n!$ permutations of an n-element set. There are n ways to choose the first item, $n - 1$ ways to choose the second item, and so on. Therefore, the total number of permutations is

$$n(n-1)\cdots 2 \cdot 1 = n!.$$

3. An ordering of r elements selected from $x_1, \ldots, x_n$

4. There are $n(n-1)\cdots(n-r+1)$ r-permutations of an n-element set. There are n ways to choose the first item, $n-1$ ways to choose the second item, …, and $n-r+1$ ways to choose the rth element. Therefore, the total number of r-permutations is

$$n(n-1)\cdots(n-r+1).$$

5. $P(n, r)$

6. An r-element subset of $\{x_1, \ldots, x_n\}$

7. There are

$$\frac{n!}{(n-r)!r!}$$

r-combinations of an n-element set.

There are $P(n, r)$ ways to select an r-permutation of an n-element set. This r-permutation can also be constructed by first choosing an r-combination [$C(n, r)$ ways] and then ordering it [$r!$ ways]. Therefore, $P(n,r) = C(n,r)r!$. Thus

$$C(n,r) = \frac{P(n,r)}{r!} = \frac{n(n-1)\cdots(n-r+1)}{r!} = \frac{n!}{(n-r)!r!}.$$

8. $C(n,r)$

Section 6.2

1. $4! = 24$

4. $abc, acb, bac, bca, cab, cba, abd, adb, bad, bda, dab, dba, acd, adc, cad, cda, dac, dca, bcd, bdc, cbd, cdb, dbc, dcb$

7. $P(11,3) = 11 \cdot 10 \cdot 9$ **10.** $3!$

13. $4!$ contain the substring AE and $4!$ contain the substring EA; therefore, the total number is $2 \cdot 4!$.

16. We first count the number N of strings that contain either the substring AB or the substring BE. The answer to the exercise will be: Total number of strings $- N$ or $5! - N$.

According to Exercise 65, Section 6.1, the number of strings that contain AB or $BE =$ number of strings that contain $AB +$ number of strings that contain $BE -$ number of strings that contain AB and BE. A string contains AB and BE if and only if it contains ABE and the number of such strings is $3!$. The number of strings that contain $AB =$ number of strings that contain $BE = 4!$. Thus the number of strings that contain AB or BE is $4! + 4! - 3!$. The solution to the exercise is

$$5! - (2 \cdot 4! - 3!).$$

19. $8!P(9,5) = 8!(9 \cdot 8 \cdot 7 \cdot 6 \cdot 5)$

21. $10!$

24. Fix a seat for a Jovian. There are $7!$ arrangements for the remaining Jovians. For each of these arrangements, we can place the Martians in five of the eight in-between positions, which can be done in $P(8,5)$ ways. Thus there are $7!P(8,5)$ such arrangements.

25. $C(4,3) = 4$ **28.** $C(11,3)$

31. $C(17,0) + C(17,1) + C(17,2) + C(17,3) + 4$

33. $C(13,5)$

36. A committee that has at most one man has exactly one man or no men. There are $C(6,1)C(7,3)$ committees with exactly one man. There are $C(7,4)$ committees with no men. Thus the answer is $C(6,1)C(7,3) + C(7,4)$.

39. $C(10,4)C(12,3)C(4,2)$

42. First, we count the number of eight-bit strings with no two 0's in a row. We divide this problem into counting the number of such strings with exactly eight 1's, with exactly seven 1's, and so on.

There is one eight-bit string with no two 0's in a row that has exactly eight 1's. Suppose that an eight-bit string with no two 0's in a row has exactly seven 1's. The 0 can go in any one of eight positions; thus there are eight such strings. Suppose that an eight-bit string with no two 0's in a row has exactly six 1's. The two 0's must go in two of the blanks shown:

$$\_1\_1\_1\_1\_1\_1\_.$$

Thus the two 0's can be placed in $C(7,2)$ ways. Thus there are $C(7,2)$ such strings. Similarly, there are $C(6,3)$ eight-bit strings with no two 0's in a row that have exactly five 1's and there are $C(5,4)$ eight-bit strings with no two 0's in a row that have exactly four 1's in a row. If a string has less than four 1's, it will have two 0's in a row. Therefore, the number of eight-bit strings with no two 0's in a row is

$$1 + 8 + C(7,2) + C(6,3) + C(5,4).$$

Since there are 2^8 eight-bit strings, there are

$$2^8 - [1 + 8 + C(7,2) + C(6,3) + C(5,4)]$$

eight-bit strings that contain at least two 0's in a row.

43. $1 \cdot 48$ (The four aces can be chosen in one way and the fifth card can be chosen in 48 ways.)

46. First, we count the number of hands containing cards in spades and hearts. Since there are 26 spades and hearts, there are $C(26,5)$ ways to select five cards from among these 26. However, $C(13,5)$ contain only spades and $C(13,5)$ contain only hearts. Therefore, there are

$$C(26,5) - 2C(13,5)$$

ways to select five cards containing cards in spades and hearts.

Since there are $C(4,2)$ ways to select two suits, the number of hands containing cards of exactly two suits is

$$C(4,2)[C(26,5) - 2C(13,5)].$$

49. There are nine consecutive patterns: A2345, 23456, 34567, 45678, 56789, 6789T, 789TJ, 89TJQ, 9TJQK. Corresponding to the four possible suits, there are four ways for each pattern to occur. Thus there are $9 \cdot 4$ hands that are consecutive and of the same suit.

52. $C(52,13)$

55. $1 \cdot C(48,9)$ (Select the aces, then select the nine remaining cards.)

58. There are $C(13,4)C(13,4)C(13,4)C(13,1)$ hands that contain four spades, four hearts, four diamonds, and one club. Since there are four ways to select the three suits to have four cards each, there are $4C(13,4)^3C(13,1)$ hands that contain four cards of three suits and one card of the fourth suit.

60. 2^{10} **63.** 2^9 **65.** $C(50,4)$

68. $C(50,4) - C(46,4)$ (Total number $-$ number with no defectives)

72. Order the $2n$ items. The first item can be paired in $2n-1$ ways. The next (not yet selected) item can be paired in $2n-3$ ways, and so on.

73. A list of votes where Wright is never behind Upshaw and each receives r votes is a string of r W's and r U's where, reading the string from left to right, the number of W's is always greater than or equal to the number of U's. Such a string can also be considered a path of the type described in

Example 6.2.23, where W is a move right and U is a move up. Example 6.2.23 proved that there are C_r such paths. Therefore, the number of ways the votes can be counted in which Wright is never behind Upshaw is C_r.

76. By Exercise 75, k vertical steps can occur in $C(k, \lceil k/2 \rceil)$ ways, since, at any point, the number of up steps is greater than or equal to the number of down steps. Then, $n - k$ horizontal steps can be inserted among the k vertical steps in $C(n, k)$ ways. Since each horizontal step can occur in two ways, the number of paths containing exactly k vertical steps that never go strictly below the x-axis is

$$C(k, \lceil k/2 \rceil)C(n, k)2^{n-k}.$$

Summing over all k, we find that the total number of paths is

$$\sum_{k=0}^{n} C(k, \lceil k/2 \rceil)C(n, k)2^{n-k}.$$

82. The solution counts *ordered* hands.

84. Once—when we choose the five slots with 0's and 1's for the remaining slots.

89. Use Theorems 3.4.1 and 3.4.8.

92. Note that

$$\frac{n-i}{k-i} \geq \frac{n}{k},$$

for $i = 0, 1, \ldots, k - 1$. Therefore,

$$\begin{aligned} C(n,k) = \frac{n!}{(n-k)!k!} &= \frac{n(n-1)\cdots(n-k+1)}{k(k-1)\cdots 1} \\ &= \frac{n}{k}\frac{n-1}{k-1}\cdots\frac{n-k+1}{1} \\ &\geq \frac{n}{k}\frac{n}{k}\cdots\frac{n}{k} \\ &= \left(\frac{n}{k}\right)^k. \end{aligned}$$

Also,

$$C(n,k) = \frac{n(n-1)\cdots(n-k+1)}{k!} \leq \frac{nn\cdots n}{k!} = \frac{n^k}{k!}.$$

Section 6.3 Review

1. $n!/(n_1!\cdots n_t!)$. The formula derives from the Multiplication Principle. We first assign positions to the n_1 items of type 1, which can be done in $C(n, n_1)$ ways. Having made these assignments, we next assign positions to the n_2 items of type 2, which can be done in $C(n - n_1, n_2)$ ways, and so on. The number of orderings is then

$$C(n, n_1)C(n - n_1, n_2)\cdots C(n - n_1 - \cdots - n_{t-1}, n_t),$$

which, after applying the formula for $C(n, k)$ and simplification, gives $n!/(n_1!\cdots n_t!)$.

2. $C(k + t - 1, t - 1)$. The formula is obtained by considering $k + t - 1$ slots and $k + t - 1$ symbols consisting of k $\times$'s and $t - 1$ $|$'s. Each placement of these symbols into the slots determines a selection. The number n_1 of $\times$'s between the first and second $|$ represents n_1 copies of the first element in the set. The number n_2 of $\times$'s between the second and third $|$ represents n_2 copies of the second element in the set, and so on. Since there are $C(k + t - 1, t - 1)$ ways to select the positions of the $|$'s, there are also $C(k + t - 1, t - 1)$ selections.

Section 6.3

1. 5!

4. Permute one token with four S's and other tokens with one letter each from among *ALEPERON*, which can be done in $9!/2!$ ways.

7. $C(6 + 6 - 1, 6 - 1)$

10. Each such route can be designated by a string of i X's, j Y's, and k Z's, where an X means move one unit in the x-direction, a Y means move one unit in the y-direction, and a Z means move one unit in the z-direction. There are

$$\frac{(i+j+k)!}{i!j!k!}$$

such strings.

14. $10!/(5! \cdot 3! \cdot 2!)$

15. $C(10 + 3 - 1, 10)$

18. $C(9 + 2 - 1, 9)$

21. Four, since the possibilities are (0, 0), (2, 1), (4, 2), and (6, 3), where the pair (r, g) designates r red and g green balls.

22. $C(15 + 3 - 1, 15)$

25. $C(13 + 2 - 1, 13)$

28. $C(12 + 4 - 1, 12)$
$- [C(7 + 4 - 1, 7) + C(6 + 4 - 1, 6)$
$+ C(3 + 4 - 1, 3) + C(2 + 4 - 1, 2)$
$- C(1 + 4 - 1, 1)]$

33. $52!/(13!)^4$

36. $C(20, 5)$

39. $C(20, 5)^2$

42. $C(15 + 6 - 1, 15)$

45. $C(10 + 12 - 1, 10)$

48. Apply the result of Example 6.3.9 to the inner $k - 1$ nested loops of that example. Next, write out the number of iterations for $i_1 = 1$; then $i_1 = 2$; and so on. By Example 6.3.9, this sum is equal to $C(k + n - 1, k)$.

Section 6.4 Review

1. Let $\alpha = s_1 \cdots s_p$ and $\beta = t_1 \cdots t_q$ be strings over $\{1, 2, \ldots, n\}$. Then α is lexicographically less than β if either $p < q$ and $s_i = t_i$ for all $i = 1, \ldots, p$; or for some i, $s_i \neq t_i$ and for the smallest such i, we have $s_i < t_i$.

2. Given a string $s_1 \ldots s_r$, which represents the r-combination $\{s_1, \ldots, s_r\}$, to find the following string $t_1 \ldots t_r$, find the rightmost element s_m that is not at its maximum value. (s_r's maximum value is n, s_{r-1}'s maximum value is $n - 1$, etc.) Then set $t_i = s_i$ for $i = 1, \ldots, m - 1$; set $t_m = s_m + 1$; and set $t_{m+1} \cdots t_r = (s_m + 2)(s_m + 3)\cdots$. Begin with the string $12 \cdots r$.

3. Given a string s, which represents a permutation, to find the following string, find the rightmost digit d of s whose right neighbor exceeds d. Find the rightmost element r that satisfies $d < r$. Swap d and r. Finally, reverse the substring to the right of d's original position. Begin with the string $12 \cdots n$.

Section 6.4

1. 1357 **4.** 12435

7. (For Exercise 1) At lines 8–12, we find the rightmost s_m not at its maximum value. In this case, $m = 4$. At line 14, we increment s_m. This makes the last digit 7. Since m is the rightmost position, at lines 16 and 17, we do nothing. The next combination is 1357.

9. 123, 124, 125, 126, 134, 135, 136, 145, 146, 156, 234, 235, 236, 245, 246, 256, 345, 346, 356, 456

12. 12, 21

14. Input: r, n

Output: A list of all r-combinations of $\{1, 2, \ldots, n\}$ in increasing lexicographic order

```
r_comb(r, n) {
   s_0 = −1
   for i = 1 to r
      s_i = i
   println(s_1, ..., s_n)
   while (true) {
      m = r
      max_val = n
      while (s_m == max_val) {
         m = m − 1
         max_val = max_val − 1
      }
      if (m == 0)
         return
      s_m = s_m + 1
      for j = m + 1 to r
         s_j = s_{j−1} + 1
      println(s_1, ..., s_n)
   }
}
```

17. Input: $s_1, \ldots, s_r$ (an r-combination of $\{1, \ldots, n\}$), r, and n

Output: $s_1, \ldots, s_r$, the next r-combination (The first r-combination follows the last r-combination.)

```
next_comb(s, r, n) {
   s_0 = n + 1 // dummy value
   m = r
   max_val = n
   // loop test always fails if m = 0
   while (s_m == max_val) {
      // find rightmost element not at its maximum value
      m = m − 1
      max_val = max_val − 1
   }
   if (m == 0) // last r-combination detected
      s_1 = 0
      m = 1
   }
   // increment rightmost element
   s_m = s_m + 1
   // rest of elements are successors of s_m
   for j = m + 1 to r
      s_j = s_{j−1} + 1
}
```

19. Input: $s_1, \ldots, s_r$ (an r-combination of $\{1, \ldots, n\}$), r, and n

Output: $s_1, \ldots, s_r$, the previous r-combination (The last r-combination precedes the first r-combination.)

```
prev_comb(s, r, n) {
   s_0 = n // dummy value
   // find rightmost element at least
   // 2 larger than its left neighbor
   m = r
   // loop test always fails if m = 1
   while (s_m − s_{m−1} == 1)
      m = m − 1
   s_m = s_m − 1
   if (m == 1 ∧ s_1 == 0)
      m = 0
   // set elements to right of index m to max values
   for j = m + 1 to r
      s_j = n + j − r
}
```

21. Input: $r, s_k, s_{k+1}, \ldots, s_n$, a string α, k, and n

Output: A list of all r-combinations of $\{s_k, s_{k+1}, \ldots, s_n\}$ each prefixed by α [To list all r-combinations of $\{s_1, s_2, \ldots, s_n\}$, invoke this function as $r\_comb2(r, s, 1, n, \lambda)$, where λ is the null string.]

```
r_comb2(r, s, k, n, α) {
   if (r == 0) {
      println(α)
      return
   }
   if (k == n) {
      println(α, s_n)
      return
   }
   β = α + " " + s_k
   // print r-combinations containing s_k
   r_comb2(r − 1, s, k + 1, n, β)
   // print r-combinations not containing s_k
   if (r ≤ n − k)
      r_comb2(r, s, k + 1, n, α)
}
```

Section 6.5 Review

1. An experiment is a process that yields an outcome.

2. An event is an outcome or combination of outcomes from an experiment.

3. The sample space is the event consisting of all possible outcomes.

4. The number of outcomes in the event divided by the number of outcomes in the sample space

Section 6.5

1. (H, 1), (H, 2), (H, 3), (H, 4), (H, 5), (H, 6), (T, 1), (T, 2), (T, 3), (T, 4), (T, 5), (T, 6)

4. (H, 1), (H, 2), (H, 3)

5. (1, 1), (1, 3), (1, 5), (2, 2), (2, 4), (2, 6), (3, 1), (3, 3), (3, 5), (4, 2), (4, 4), (4, 6), (5, 1), (5, 3), (5, 5), (6, 2), (6, 4), (6, 6)

8. Three dice are rolled.

11. 1/6

14. 1/52

17. 4/36

20. $C(90, 4)/C(100, 4)$

23. $1/10^3$

26. $1/[C(50, 5) \cdot 36]$

28. $\dfrac{4 \cdot C(13, 5) \cdot 3 \cdot C(13, 4)C(13, 2)^2}{C(52, 13)}$

30. $1/2^{10}$

33. $C(10, 5)/2^{10}$

34. $2^{10}/3^{10}$

37. $1/5!$

38. $10/C(12, 3)$

41. 18/38

44. 2/38

45. 1/3

49. 1/4

52. The possibilities are: A (correct), B (incorrect), C (incorrect); and A (incorrect), B (incorrect), C (incorrect). In the first case, if the student stays with A, the answer will be correct; but if the student switches to B, the answer will be incorrect. In the second case, if the student stays with A, the answer will be incorrect; but if the student switches to B, the answer will be correct. Thus, the probability of a correct answer is 1/2.

55. There are $C(10 + 3 - 1, 3 - 1)$ ways to distribute 10 compact discs to Mary, Ivan, and Juan. If each receives at least two compact discs, we must distribute the remaining six discs, and there are $C(6 + 3 - 1, 3 - 1)$ ways to do this. Thus the probability that each person receives at least two discs is

$$\frac{C(6 + 3 - 1, 3 - 1)}{C(10 + 3 - 1, 3 - 1)}.$$

Section 6.6 Review

1. A probability function P assigns to each outcome x in a sample space S a number $P(x)$ so that

$$0 \le P(x) \le 1, \qquad \text{for all } x \in S$$

and

$$\sum_{x \in S} P(x) = 1.$$

2. $P(x) = 1/n$, where n is the size of the sample space.

3. The probability of E is

$$P(E) = \sum_{x \in E} P(x).$$

4. $P(E) + P(\overline{E}) = 1$

5. E_1 or E_2 (or both)

6. E_1 and E_2

7. $P(E_1 \cup E_2) = P(E_1) + P(E_2) - P(E_1 \cap E_2)$. $P(E_1) + P(E_2)$ equals $P(x)$ for all $x \in E_1$ plus $P(x)$ for all $x \in E_2$, which is equal to $P(E_1) + P(E_2)$, except that $P(x)$, for $x \in E_1 \cap E_2$, is counted twice. The formula now follows.

8. Events E_1 and E_2 are mutually exclusive if $E_1 \cap E_2 = \varnothing$.

9. If we roll two dice, the events "roll doubles" and "the sum is odd" are mutually exclusive.

10. $P(E_1 \cup E_2) = P(E_1) + P(E_2)$. This formula follows from the formula of Exercise 7 because $P(E_1 \cap E_2) = 0$.

11. E *given* F is the event E given that event F occurred.

12. $E \mid F$

13. $P(E \mid F) = P(E \cap F)/P(F)$

14. Events E and F are independent if $P(E \cap F) = P(E)P(F)$.

15. If we roll two dice, the events "get an odd number on the first die" and "get an even number on the second die" are independent.

16. Pattern recognition places items into various classes based on features of the items.

17. Suppose that the possible classes are $C_1, \ldots, C_n$. Suppose further that each pair of classes is mutually exclusive and each item to be classified belongs to one of the classes. For a feature set F, we have

$$P(C_j \mid F) = \frac{P(F \mid C_j)P(C_j)}{\sum_{i=1}^{n} P(F \mid C_i)P(C_i)}.$$

The equation

$$P(C_j \mid F) = \frac{P(C_j \cap F)}{P(F)} = \frac{P(F \mid C_j)P(C_j)}{P(F)}$$

follows from the definition of conditional probability. The proof is completed by showing that

$$P(F) = \sum_{i=1}^{n} P(F \mid C_i)P(C_i),$$

which follows from the fact that each pair of classes is mutually exclusive and each item to be classified belongs to one of the classes.

Section 6.6

1. 1/8

4. $P(2) = P(4) = P(6) = 1/12$. $P(1) = P(3) = P(5) = 3/12$.

7. $1 - (1/4)$

8. $3(1/12)^2 + 3(3/12)^2$

11. Let E denote the event "sum is 6," and let F denote the event "at least one die shows 2." Then

$$P(E \cap F) = P((2, 4)) + P((4, 2)) = 2\left(\frac{1}{12}\right)^2 = \frac{2}{144},$$

and

$$\begin{aligned} P(F) &= P((1,2)) + P((2,1)) + P((2,2)) + P((2,3)) \\ &\quad + P((2,4)) + P((2,5)) + P((2,6)) + P((3,2)) \\ &\quad + P((4,2)) + P((5,2)) + P((6,2)) \\ &= \left(\frac{3}{12}\right)\left(\frac{1}{12}\right) + \left(\frac{1}{12}\right)\left(\frac{3}{12}\right) + \left(\frac{1}{12}\right)^2 \\ &\quad + \left(\frac{1}{12}\right)\left(\frac{3}{12}\right) + \left(\frac{1}{12}\right)^2 + \left(\frac{1}{12}\right)\left(\frac{3}{12}\right) \\ &\quad + \left(\frac{1}{12}\right)^2 + \left(\frac{3}{12}\right)\left(\frac{1}{12}\right) + \left(\frac{1}{12}\right)^2 \\ &\quad + \left(\frac{3}{12}\right)\left(\frac{1}{12}\right) + \left(\frac{1}{12}\right)^2 = \frac{23}{144}. \end{aligned}$$

Therefore,

$$P(E \mid F) = \frac{P(E \cap F)}{P(F)} = \frac{\frac{2}{144}}{\frac{23}{144}} = \frac{2}{23}.$$

14. (T,1), (T,2), (T,3), (T,4), (T,5), (T,6), (H,3)

17. Yes

19. $C(90, 6)/C(100, 6)$

22. $1/2^4$

25. $\dfrac{\frac{1}{2^4}}{\frac{2^4-1}{2^4}} = \dfrac{1}{15}$

28. Let E_1 denote the event "children of both sexes," and let E_2 denote the event "at most one boy." Then

$$P(E_1) = \frac{14}{16}, \quad P(E_2) = \frac{5}{16}, \quad \text{and} \quad P(E_1 \cap E_2) = \frac{4}{16}.$$

Now

$$P(E_1 \cap E_2) = \frac{1}{4} \neq \frac{35}{128} = P(E_1)P(E_2).$$

Therefore, the events E_1 and E_2 are not independent.

31. $1/2^{10}$

34. $1 - (1/2^{10})$

37. Let E denote the event "four or five or six heads," and let F denote the event "at least one head." Then

$$\begin{aligned} P(E \cap F) &= \frac{C(10,4)}{2^{10}} + \frac{C(10,5)}{2^{10}} + \frac{C(10,6)}{2^{10}} \\ &= \frac{210 + 252 + 210}{2^{10}} = 0.65625. \end{aligned}$$

Since $P(F) = 1 - (1/2^{10}) = 0.999023437$,

$$P(E \mid F) = \frac{P(E \cap F)}{P(F)} = \frac{0.65625}{0.999023437} = 0.656891495.$$

40. Let E be the event "at least one person has a birthday on April 1." Then $\overline{E}$ is the event "no one has a birthday on April 1." Now

$$P(E) = 1 - P(\overline{E}) = 1 - \frac{364 \cdot 364 \cdots 364}{365 \cdot 365 \cdots 365} = 1 - \left(\frac{364}{365}\right)^n.$$

44. Let E_1 denote the event "over 350 pounds," and let E_2 denote the event "bad guy." Then

$$\begin{aligned} P(E_1 \cup E_2) &= P(E_1) + P(E_2) - P(E_1 \cap E_2) \\ &= \frac{35}{90} + \frac{20}{90} - \frac{15}{90} = \frac{40}{90}. \end{aligned}$$

46. $P(A) = 0.55$, $P(D) = 0.10$, $P(N) = 0.35$

49. $P(B) = P(B \mid A)P(A) + P(B \mid D)P(D) + P(B \mid N)P(N)$
$= (0.10)(0.55) + (0.30)(0.10) + (0.30)(0.35) = 0.19$

50. We require that

$$P(H \mid Pos) = 0.5 = \frac{(0.95)P(H)}{(0.95)P(H) + (0.02)(1 - P(H))}.$$

Solving for $P(H)$ gives $P(H) = .0206$.

53. Yes. Suppose that E and F are independent, that is, $P(E)P(F) = P(E \cap F)$. Now

$$\begin{aligned} P(\overline{E})P(\overline{F}) &= (1 - P(E))(1 - P(F)) \\ &= 1 - P(E) - P(F) + P(E)P(F) \\ &= 1 - P(E) - P(F) + P(E \cap F). \end{aligned}$$

By De Morgan's law for sets,

$$\overline{E} \cap \overline{F} = \overline{E \cup F};$$

thus,

$$\begin{aligned} P(\overline{E} \cap \overline{F}) &= P(\overline{E \cup F}) \\ &= 1 - P(E \cup F) \\ &= 1 - [P(E) + P(F) - P(E \cap F)] \\ &= 1 - P(E) - P(F) + P(E \cap F). \end{aligned}$$

Therefore,

$$P(\overline{E})P(\overline{F}) = P(\overline{E} \cap \overline{F}),$$

and $\overline{E}$ and $\overline{F}$ are independent.

56. Let E_i be the event "runner completes the marathon on attempt i." The error in the reasoning is assuming that $P(E_2) = 1/3 = P(E_3)$. In fact, $P(E_2) \neq 1/3 \neq P(E_3)$ because, if the runner completes the marathon, it is not run again. Although $P(E_1) = 1/3$,

$$\begin{aligned} P(E_2) &= P(\text{fail on attempt 1 and succeed on attempt 2}) \\ &= P(\text{fail on attempt 1})P(\text{succeed on attempt 2}) \\ &= \frac{2}{3} \cdot \frac{1}{3} = \frac{2}{9}. \end{aligned}$$

Similarly,

$$P(E_3) = \frac{2}{3} \cdot \frac{2}{3} \cdot \frac{1}{3} = \frac{4}{27}.$$

Thus, the probability of completing the marathon is

$$\begin{aligned} P(E_1 \cup E_2 \cup E_3) &= P(E_1) + P(E_2) + P(E_3) \\ &= \frac{1}{3} + \frac{2}{9} + \frac{4}{27} = \frac{19}{27} = 0.704, \end{aligned}$$

which means that there is about a 70 percent chance that the runner will complete the marathon—not exactly a virtual certainty!

Section 6.7 Review

1. If a and b are real numbers and n is a positive integer, then

$$(a+b)^n = \sum_{k=0}^{n} C(n,k)a^{n-k}b^k.$$

2. In the expansion of

$$(a+b)^n = \underbrace{(a+b)(a+b)\cdots(a+b)}_{n \text{ factors}},$$

the term $a^{n-k}b^k$ arises from choosing b from k factors and a from the other $n-k$ factors, which can be done in $C(n,k)$ ways. Summing over all k gives the Binomial Theorem.

3. Pascal's triangle is an arrangement of the binomial coefficients in triangular form. The border consists of 1's, and any interior value is the sum of the two numbers above it:

$$\begin{array}{ccccccccccc}
 & & & & & 1 & & & & & \\
 & & & & 1 & & 1 & & & & \\
 & & & 1 & & 2 & & 1 & & & \\
 & & 1 & & 3 & & 3 & & 1 & & \\
 & 1 & & 4 & & 6 & & 4 & & 1 & \\
1 & & 5 & & 10 & & 10 & & 5 & & 1 \\
 & & & & & \vdots & & & & &
\end{array}$$

4. $C(n,0) = C(n,n) = 1$, for all $n \ge 0$; and $C(n+1,k) = C(n,k-1) + C(n,k)$, for all $1 \le k \le n$

Section 6.7

1. $x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + y^4$

3. $C(11,7)x^4y^7$ **6.** 5,987,520

9. $C(7,3) + C(5,2)$, since

$$\begin{aligned}(a+\sqrt{ax}+x)^2(a+x)^5 &= [(a+x)+\sqrt{ax}]^2(a+x)^5 \\ &= (a+x)^7 + 2\sqrt{ax}(a+x)^6 + ax(a+x)^5.\end{aligned}$$

10. $C(10+3-1,10)$ **13.** 1 8 28 56 70 56 28 8 1

16. [Inductive Step only] Assume that the theorem is true for n.

$$\begin{aligned}
(a+b)^{n+1} &= (a+b)(a+b)^n \\
&= (a+b)\sum_{k=0}^{n} C(n,k)a^{n-k}b^k \\
&= \sum_{k=0}^{n} C(n,k)a^{n+1-k}b^k \\
&\quad + \sum_{k=0}^{n} C(n,k)a^{n-k}b^{k+1} \\
&= \sum_{k=0}^{n} C(n,k)a^{n+1-k}b^k \\
&\quad + \sum_{k=1}^{n+1} C(n,k-1)a^{n+1-k}b^k \\
&= C(n,0)a^{n+1}b^0 + \sum_{k=1}^{n} C(n,k)a^{n+1-k}b^k \\
&\quad + C(n,n)a^0b^{n+1} \\
&\quad + \sum_{k=1}^{n} C(n,k-1)a^{n+1-k}b^k \\
&= C(n+1,0)a^{n+1}b^0 \\
&\quad + \sum_{k=1}^{n}[C(n,k)+C(n,k-1)]a^{n+1-k}b^k \\
&\quad + C(n+1,n+1)a^0b^{n+1} \\
&= C(n+1,0)a^{n+1}b^0 \\
&\quad + \sum_{k=1}^{n} C(n+1,k)a^{n+1-k}b^k \\
&\quad + C(n+1,n+1)a^0b^{n+1} \\
&= \sum_{k=0}^{n+1} C(n+1,k)a^{n+1-k}b^k
\end{aligned}$$

19. The number of solutions in nonnegative integers of

$$x_1 + x_2 + \cdots + x_{k+2} = n-k$$

is $C(k+2+n-k-1, n-k) = C(n+1,k+1)$. The number of solutions is also the number of solutions $C(k+1+n-k-1, n-k) = C(n,k)$ with $x_{k+2} = 0$ plus the number of solutions

$$C(k+1+n-k-1-1, n-k-1) = C(n-1,k)$$

with $x_{k+2} = 1$ plus $\cdots$ plus the number of solutions $C(k+1+0-1,0) = C(k,k)$ with $x_{k+2} = n-k$. The result now follows.

22. Take $a = 1$ and $b = 2$ in the Binomial Theorem.

25. $x^3 + 3x^2y + 3x^2z + 3xy^2 + 6xyz + 3xz^2 + y^3 + 3y^2z + 3yz^2 + z^3$

28. Set $a = 1$ and $b = x$ and replace n by $n-1$ in the Binomial Theorem to obtain

$$(1+x)^{n-1} = \sum_{k=0}^{n-1} C(n-1,k)x^k.$$

Now multiply by n to obtain

$$\begin{aligned}
n(1+x)^{n-1} &= n\sum_{k=0}^{n-1} C(n-1,k)x^k \\
&= n\sum_{k=1}^{n} C(n-1,k-1)x^{k-1} \\
&= \sum_{k=1}^{n} \frac{n(n-1)!}{(n-k)!\,(k-1)!}x^{k-1} \\
&= \sum_{k=1}^{n} \frac{n!}{(n-k)!\,k!}kx^{k-1} \\
&= \sum_{k=1}^{n} C(n,k)kx^{k-1}.
\end{aligned}$$

31. The solution is by induction on k. We omit the Basis Step. Assume that the statement is true for k. After k iterations, we obtain the sequence defined by

$$a'_j = \sum_{i=0}^{k-1} a_{i+j} \frac{B_i}{2^n}.$$

Let $B'_0, \ldots, B'_k$ denote the row after $B_0, \ldots, B_{k-1}$ in Pascal's triangle. Smoothing a' by c to obtain a'' yields

$$\begin{aligned} a''j &= \frac{1}{2}(a'_j + a'_{j+1}) \\ &= \frac{1}{2^{n+1}}\left(\sum_{i=0}^{k-1} a_{i+j}B_i + \sum_{i=0}^{k-2} a_{i+j+1}B_i\right) \\ &= \frac{1}{2^{n+1}}\left(a_jB_0 + \sum_{i=1}^{k-1} a_{i+j}B_i + \sum_{i=0}^{k-2} a_{i+j+1}B_i + a_{k+j}B_{k-1}\right) \\ &= \frac{1}{2^{n+1}}\left(a_jB_0 + \sum_{i=1}^{k-1} a_{i+j}B_i + \sum_{i=1}^{k-1} a_{i+j}B_{i-1} + a_{k+j}B_{k-1}\right) \\ &= \frac{1}{2^{n+1}}\left(a_jB'_0 + \sum_{i=1}^{k-1} a_{i+j}B'_i + a_{k+j}B'_k\right) \\ &= \frac{1}{2^{n+1}} \sum_{i=0}^{k} a_{i+j}B'_i, \end{aligned}$$

and the Inductive Step is complete.

34. [Inductive Step only] Notice that

$$C(n+1,i)^{-1} + C(n+1,i+1)^{-1} = \frac{n+2}{n+1}C(n,i)^{-1}.$$

Now

$$\begin{aligned} &\sum_{i=1}^{n+1} C(n+1,i)^{-1} \\ &= \frac{1}{2}\left(\sum_{i=1}^{n+1} C(n+1,i)^{-1} + \sum_{i=0}^{n} C(n+1,i+1)^{-1}\right) \\ &= \frac{1}{2}\left(C(n+1,1)^{-1} + \frac{n+2}{n+1}\sum_{i=1}^{n} C(n,i)^{-1} + C(n+1,n+1)^{-1}\right) \\ &= \frac{1}{2}\left(\frac{n+2}{n+1} + \frac{n+2}{2^n}\sum_{i=0}^{n-1} \frac{2^i}{i+1}\right) \\ &= \frac{n+2}{2^{n+1}}\sum_{i=0}^{n} \frac{2^i}{i+1}. \end{aligned}$$

Section 6.8 Review

1. *First Form:* If n pigeons fly into k pigeonholes and $k < n$, some pigeonhole contains at least two pigeons.

Second Form: If f is a function from a finite set X to a finite set Y and $|X| > |Y|$, then $f(x_1) = f(x_2)$ for some $x_1, x_2 \in X$, $x_1 \neq x_2$.

Third Form: Let f be a function from a finite set X to a finite set Y. Suppose that $|X| = n$ and $|Y| = m$. Let $k = \lceil n/m \rceil$. Then there are at least k values $a_1, \ldots, a_k \in X$ such that

$$f(a_1) = f(a_2) = \cdots = f(a_k).$$

2. *First Form:* If 20 persons (pigeons) go into six rooms (pigeonholes), then some room contains at least two persons.

Second Form: In the previous example, let X be the set of persons, and let Y be the set of rooms. If p is a person, define a function f by letting $f(p)$ be the room in which person p is located. Then for some distinct persons p_1 and p_2, $f(p_1) = f(p_2)$; that is, the distinct persons p_1 and p_2 are in the same room.

Third Form: Let X, Y, and f be as in the last example. Then there are at least $\lceil 20/6 \rceil = 4$ persons p_1, p_2, p_3, p_4 such that

$$f(p_1) = f(p_2) = f(p_3) = f(p_4);$$

that is, there are at least four persons in the same room.

Section 6.8

1. Let the five cards be the pigeons and the four suits be the pigeonholes. Assign each card (pigeon) to its suit (pigeonhole). By the Pigeonhole Principle, some pigeonhole (suit) will contain at least two pigeons (cards), that is, at least two cards are of the same suit.

4. Let the 35 students be the pigeons and the 24 letters of the alphabet be the pigeonholes. Assign each student (pigeon) the first letter of the first name (pigeonhole). By the Pigeonhole Principle, some pigeonhole (letter) will contain at least two pigeons (students), that is, at least two students have first names that start with the same letter.

7. Let the 13 persons be the pigeons and the $12 = 3 \cdot 4$ possible names be the pigeonholes. Assign each person (pigeon) that person's name (pigeonhole). By the Pigeonhole Principle, some pigeonhole (name) will contain at least two pigeons (persons), that is, at least two persons have the same first and last names.

10. Yes. Connect processors 1 and 2, 2 and 3, 2 and 4, 3 and 4. Processor 5 is not connected to any processors. Now only processors 3 and 4 are directly connected to the same number of processors.

13. Let a_i denote the position of the ith unavailable item. Consider

$$a_1, \ldots, a_{30}; \quad a_1 + 3, \ldots, a_{30} + 3; \quad a_1 + 6, \ldots, a_{30} + 6.$$

These 90 numbers range in value from 1 to 86. By the second form of the Pigeonhole Principle, two of these numbers are the same. If $a_i = a_j + 3$, two are three apart. If $a_i = a_j + 6$, two are six apart. If $a_i + 3 = a_j + 6$, two are three apart.

17. $n + 1$

18. Suppose that $k \le m/2$. Clearly, $k \ge 1$. Since $m \le 2n+1$,

$$k \le \frac{m}{2} \le n + \frac{1}{2} < n+1.$$

Suppose that $k > m/2$. Then

$$m - k < m - \frac{m}{2} = \frac{m}{2} < n+1.$$

Because m is the largest element in X, $k < m$. Thus $k+1 \le m$ and so $1 \le m-k$. Therefore, the range of a is contained in $\{1, \ldots, n\}$.

19. The second form of the Pigeonhole Principle applies.

20. Suppose that $a_i = a_j$. Then either $i \le m/2$ and $j > m/2$ or $j \le m/2$ and $i > m/2$. We may assume that $i \le m/2$ and $j > m/2$. Now

$$i + j = a_i + m - a_j = m.$$

30. When we divide a by b, the possible remainders are $0, 1, \ldots, b-1$. Consider what happens after b divisions.

34. We suppose that the board has three rows and seven columns. We call two squares in one column that are the same color a *colorful pair*. By the Pigeonhole Principle, each column contains at least one colorful pair. Thus the board contains seven colorful pairs, one in each column. Again by the Pigeonhole Principle, at least four of these seven colorful pairs are the same color, say red. Since there are three pairs of rows and four red colorful pairs, a third application of the Pigeonhole Principle shows that at least two columns contain red colorful pairs in the same rows. These colorful pairs determine a rectangle whose four corner squares are red.

37. Suppose that it is possible to mark k squares in the upper-left $k \times k$ subgrid and k squares in the lower-right $k \times k$ subgrid so that no two marked squares are in the same row, column, or diagonal of the $2k \times 2k$ grid. Then the $2k$ marked squares are contained in $2k-1$ diagonals. One diagonal begins at the top left square and runs to the bottom right square; $k-1$ diagonals begin at the $k-1$ squares immediately to the right of the top left square and run parallel to the first diagonal described; and $k-1$ diagonals begin at the $k-1$ squares immediately under the top left square and run parallel to the others described. By the first form of the Pigeonhole Principle, some diagonal contains two marked squares. This contradiction shows that it is impossible to mark k squares in the upper-left $k \times k$ subgrid and k squares in the lower-right $k \times k$ subgrid so that no two marked squares are in the same row, column, or diagonal of the $2k \times 2k$ grid.

Chapter 6 Self-Test

1. 2^4

2. $6 \cdot 9 \cdot 7 + 6 \cdot 9 \cdot 4 + 6 \cdot 7 \cdot 4 + 9 \cdot 7 \cdot 4$

3. $2^n - 2$

4. $6 \cdot 5 \cdot 4 \cdot 3 + 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2$

5. $6!/(3!\,3!) = 20$

6. We construct the strings by a three-step process. First, we choose positions for A, C, and E [$C(6,3)$ ways]. Next, we place A, C, and E in these positions. We can place C one way (last), and we can place A and E two ways (AE or EA). Finally, we place the remaining three letters ($3!$ ways). Therefore, the total number of strings is $C(6,3) \cdot 2 \cdot 3!$.

7. Two suits can be chosen in $C(4,2)$ ways. We can choose three cards of one suit in $C(13,3)$ ways and we can choose three cards of the other suit in $C(13,3)$ ways. Therefore, the total number of hands is $C(4,2)C(13,3)^2$.

8. We must select either three or four defective discs. Thus the total number of selections is $C(5,3)C(95,1) + C(5,4)$.

9. $8!/(3!2!)$

10. We count the number of strings in which no I appears before any L and then subtract from the total number of strings.

We construct strings in which no I appears before any L by a two-step process. First, we choose positions for N, O, and S; then we place the I's and L's. We can choose positions for N, O, and S in $8 \cdot 7 \cdot 6$ ways. The I's and L's can then be placed in only one way because the L's must come first. Thus there are $8 \cdot 7 \cdot 6$ strings in which no I appears before any L.

Exercise 9 shows that there are $8!/(3!\,2!)$ strings formed by ordering the letters *ILLINOIS*. Therefore, there are

$$\frac{8!}{3!\,2!} - 8 \cdot 7 \cdot 6$$

strings formed by ordering the letters *ILLINOIS* in which some I appears before some L.

11. $12!/(3!)^4$

12. $C(11+4-1, 4-1)$

13. 12567

14. 234567

15. 6427153

16. 631245

17. 1/4

18. 5/36

19. $\dfrac{C(7,5)C(31-7,2)}{C(31,7)} = \dfrac{21 \cdot 276}{2629575} = 0.002204158$

20. $\dfrac{4 \cdot C(13,6) \cdot 3 \cdot C(13,5) \cdot 2 \cdot C(13,2)}{C(52,13)}$

21. $P(H) = 5/6$, $P(T) = 1/6$

22. Let S denote the event "children of both sexes," and let G denote the event "at most one girl." Then

$$P(S) = \frac{6}{8} = \frac{3}{4}$$
$$P(G) = \frac{4}{8} = \frac{1}{2}$$
$$P(S \cap G) = \frac{3}{8}.$$

Therefore,

$$P(S)P(G) = \frac{3}{4} \cdot \frac{1}{2} = \frac{3}{8} = P(S \cap G),$$

and S and G are independent.

23. Let J denote the event "Joe passes," and let A denote the event "Alicia passes." Then

$$\begin{aligned}
P(\text{Joe fails}) &= P(\overline{J}) = 1 - P(J) = 0.25 \\
P(\text{both pass}) &= P(J \cap A) = P(J)P(A) \\
&= (0.75)(0.80) = 0.6 \\
P(\text{both fail}) &= P(\overline{J} \cap \overline{A}) = P(\overline{J \cup A}) \\
&= 1 - P(J \cup A) \\
&= 1 - [P(J) + P(A) - P(J \cap A)] \\
&= 1 - [0.75 + 0.80 - 0.6] = 0.05 \\
P(\text{at least one passes}) &= 1 - P(\text{both fail}) \\
&= 1 - 0.05 = 0.95.
\end{aligned}$$

24. Let B denote the event "bug present," and let T, R, and J denote the events "Trisha (respectively, Roosevelt, José) wrote the program." Then

$$\begin{aligned}
P(J \mid B) &= \frac{P(B \mid J)P(J)}{P(B \mid J)P(J) + P(B \mid T)P(T) + P(B \mid R)P(R)} \\
&= \frac{(0.05)(0.25)}{(0.05)(0.25) + (0.03)(0.30) + (0.02)(0.45)} \\
&= 0.409836065.
\end{aligned}$$

25. $$\begin{aligned}(s-r)^4 &= C(4,0)s^4 + C(4,1)s^3(-r) + C(4,2)s^2(-r)^2 \\ &\quad + C(4,3)s(-r)^3 + C(4,4)(-r)^4 \\ &= s^4 - 4s^3r + 6s^2r^2 - 4sr^3 + r^4\end{aligned}$$

26. $2^3 \cdot 8!/(3!\,1!\,4!)$

27. If we set $a = 2$ and $b = -1$ in the Binomial Theorem, we obtain

$$1 = 1^n = [2 + (-1)]^n = \sum_{k=0}^{n} C(n,k)2^{n-k}(-1)^k.$$

28. $C(n, 1) = n$

29. Let the 15 individual socks be the pigeons and let the 14 types of pairs be the pigeonholes. Assign each sock (pigeon) to its type (pigeonhole). By the Pigeonhole Principle, some pigeonhole will contain at least two pigeons (the matched socks).

30. There are $3 \cdot 2 \cdot 3 = 18$ possible names for the 19 persons. We can consider the assignment of names to people to be that of assigning pigeonholes to the pigeons. By the Pigeonhole Principle, some name is assigned to at least two persons.

31. Let a_i denote the position of the ith available item. The 220 numbers

$$a_1, \ldots, a_{110}; \qquad a_1 + 19, \ldots, a_{110} + 19$$

range from 1 to 219. By the Pigeonhole Principle, two are the same.

32. Each point has an x-coordinate that is either even or odd and a y-coordinate that is either even or odd. Since there are four possibilities and there are five points, by the Pigeonhole Principle at least two points, $p_i = (x_i, y_i)$ and $p_j = (x_j, y_j)$ have

- Both x_i and x_j even or both x_i and x_j odd.

and

- Both y_i and y_j even or both y_i and y_j odd.

Therefore, $x_i + x_j$ is even and $y_i + y_j$ is even. In particular, $(x_i + x_j)/2$ and $(y_i + y_j)/2$ are integers. Thus the midpoint of the pair p_i and p_j has integer coordinates.

Section 7.1 Review

1. A recurrence relation defines the nth term of a sequence in terms of certain of its predecessors.

2. An initial condition for a sequence is an explicitly given value for a particular term in the sequence.

3. Compound interest is interest on interest. If a person invests d dollars at p percent compounded annually and we let A_n be the amount of money earned after n years, the recurrence relation

$$A_n = \left(1 + \frac{p}{100}\right) A_{n-1}$$

together with the initial condition $A_0 = d$ defines the sequence $\{A_n\}$.

4. The Tower of Hanoi puzzle consists of three pegs mounted on a board and disks of various sizes with holes in their centers. Only a disk of smaller diameter can be placed on a disk of larger diameter. Given all the disks stacked on one peg, the problem is to transfer the disks to another peg by moving one disk at a time.

5. If there is one disk, move it and stop. If there are $n > 1$ disks, recursively move $n-1$ disks to an empty peg. Move the largest disk to the remaining empty peg. Recursively move $n-1$ disks on top of the largest disk.

6. We assume that at time n, the quantity q_n sold at price p_n is given by the equation $p_n = a - bq_n$, where a and b are positive parameters. We also assume that $p_n = kq_{n+1}$, where k is another positive parameter. If we graph the price and quantity over time, the graph resembles a cobweb (see, e.g., Figure 7.1.5).

7. Ackermann's function $A(m, n)$ is defined by the recurrence relations

$$\begin{aligned}
A(m,0) &= A(m-1,1), & m &\ge 1 \\
A(m,n) &= A(m-1, A(m, n-1)), & m &\ge 1, n \ge 1
\end{aligned}$$

and initial conditions

$$A(0,n) = n+1, \qquad n \ge 0.$$

Section 7.1

1. $a_n = a_{n-1} + 4;\ a_1 = 3$

4. $A_n = (1.14)A_{n-1}$ **5.** $A_0 = 2000$

6. $A_1 = 2280$, $A_2 = 2599.20$, $A_3 = 2963.088$

7. $A_n = (1.14)^n 2000$

8. We must have $A_n = 4000$ or $(1.14)^n 2000 = 4000$ or $(1.14)^n = 2$. Taking the logarithm of both sides, we must have $n \log 1.14 = \log 2$. Thus

$$n = \frac{\log 2}{\log 1.14} = 5.29.$$

18. We count the number of n-bit strings not containing the pattern 000.

- Begin with 1. In this case, if the remaining $(n-1)$-bit string does not contain 000, neither will the n-bit string. There are S_{n-1} such $(n-1)$-bit strings.
- Begin with 0. There are two cases to consider.
  1. Begin with 01. In this case, if the remaining $(n-2)$-bit string does not contain 000, neither will the n-bit string. There are S_{n-2} such $(n-2)$-bit strings.
  2. Begin with 00. Then the third bit must be a 1 and if the remaining $(n-3)$-bit string does not contain 000, neither will the n-bit string. There are S_{n-3} such $(n-3)$-bit strings.

Since the cases are mutually exclusive and cover all n-bit strings $(n > 3)$ not containing 000, we have $S_n = S_{n-1} + S_{n-2} + S_{n-3}$ for $n > 3$. $S_1 = 2$ (there are two 1-bit strings), $S_2 = 4$ (there are four 2-bit strings), and $S_3 = 7$ (there are eight 3-bit strings but one of them is 000).

19. There are S_{n-1} n-bit strings that begin 1 and do not contain the pattern 00 and there are S_{n-2} n-bit strings that begin 0 (since the second bit must be 1) and do not contain the pattern 00. Thus $S_n = S_{n-1} + S_{n-2}$. Initial conditions are $S_1 = 2$, $S_2 = 3$.

22. $S_1 = 2$, $S_2 = 4$, $S_3 = 7$, $S_4 = 12$

25. $C_3 = 5$, $C_4 = 14$, $C_5 = 42$

28. We first prove that if $n \geq 5$, then C_n is not prime. Suppose, by way of contradiction, that C_n is prime for some $n \geq 5$. By Exercise 27, $n + 2 < C_n$. Thus C_n does not divide $n + 2$. By Exercise 26,

$$(n+2)C_{n+1} = (4n+2)C_n.$$

Thus C_n divides $(n+2)C_{n+1}$. By Exercise 25, Section 5.3, C_n divides either $n + 2$ or C_{n+1}. Since C_n does not divide $n+2$, C_n divides C_{n+1}. Therefore there exists an integer $k \geq 1$ satisfying $C_{n+1} = kC_n$. Thus

$$(n+2)kC_n = (4n+2)C_n.$$

Canceling C_n, we obtain

$$(n+2)k = (4n+2).$$

If $k = 1$, the preceding equation becomes $n+2 = 4n+2$, and thus $n = 0$, which contradicts the fact that $n \geq 5$. Similarly, if $k = 2$, then $n = 1$, and if $k = 3$, then $n = 4$, both of which contradict the fact that $n \geq 5$. If $k \geq 4$,

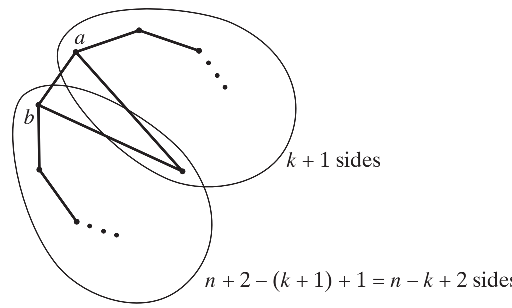
$$4n + 2 = k(n+2) \geq 4(n+2) = 4n + 8.$$

Therefore $0 \geq 6$. Thus k does not exist. This contradiction shows that if $n \geq 5$, C_n is not prime.

Directly checking $n = 0, 1, 2, 3, 4$ shows that only $C_2 = 2$ and $C_3 = 5$ are prime.

31. Let P_n denote the number of ways to divide a convex $(n+2)$-sided polygon, $n \geq 1$, into triangles by drawing $n - 1$ lines through the corners that do not intersect in the interior of the polygon. We note that $P_1 = 1$.

Suppose that $n > 1$ and consider a convex $(n+2)$-sided polygon (see the following figure).



We choose one edge ab and construct a partition of the polygon by a two-step procedure. First we select a triangle to which side ab belongs. This triangle divides the original polygon into two polygons: one having $k + 1$ sides, for some k satisfying $1 \leq k \leq n$; and the other having $n - k + 2$ sides (see the preceding figure). By definition, the $(k+1)$-sided polygon can be partitioned in P_{k-1} ways and the $(n-k+2)$-sided polygon can be partitioned in P_{n-k} ways. (For the degenerate cases $k = 1$ and $k = n$, we set $P_0 = 1$.) Therefore, the total number of ways to partition the $(n+2)$-sided polygon is

$$P_n = \sum_{k=1}^{n} P_{k-1} P_{n-k}.$$

Since the sequence $P_1, P_2, \ldots$ satisfies the same recurrence relation as the Catalan sequence $C_1, C_2, \ldots$ and $P_0 = P_1 = 1 = C_0 = C_1$, it follows that $P_n = C_n$ for all $n \geq 1$.

36. [For $n = 3$]

Step 1—move disk 3 from peg 1 to peg 3.
Step 2—move disk 2 from peg 1 to peg 2.
Step 3—move disk 3 from peg 3 to peg 2.
Step 4—move disk 1 from peg 1 to peg 3.
Step 5—move disk 3 from peg 2 to peg 1.
Step 6—move disk 2 from peg 2 to peg 3.
Step 7—move disk 3 from peg 1 to peg 3.

38. Let α and β be the angles shown in Figure 7.1.6. The geometry of the situation shows that the price tends to stabilize if and only if $\alpha + \beta > 180°$. This last condition holds if and only if $-\tan\beta < \tan\alpha$. Since $b = -\tan\beta$ and $k = \tan\alpha$, we conclude that the price stabilizes if and only if $b < k$.

40. $A(2, 2) = 7$, $A(2, 3) = 9$ **43.** $A(3, n) = 2^{n+3} - 3$

46. If $m = 0$,

$$\begin{aligned} A(m, n+1) &= A(0, n+1) \\ &= n + 2 > n + 1 \\ &= A(0, n) = A(m, n). \end{aligned}$$

The last inequality follows from Exercise 44.

47. Use Exercises 41 and 42.

50. We prove the statement by using induction on x. The inductive step will itself require induction on y.

Exercise 47 shows that the equation is true for $x = 0, 1, 2$ and for all y.

Basis Step ($x = 2$) See Exercise 47.

Inductive Step (Case x implies case $x + 1$) Assume that $x \geq 2$ and

$$A(x, y) = AO(x, 2, y + 3) - 3 \qquad \text{for all } y \geq 0.$$

We must prove that

$$A(x + 1, y) = AO(x + 1, 2, y + 3) - 3 \qquad \text{for all } y \geq 0.$$

We establish this last equation by induction on y.

Basis Step ($y = 0$) We must prove that

$$A(x + 1, 0) = AO(x + 1, 2, 3) - 3.$$

Now

$$\begin{aligned} &AO(x + 1, 2, 3) - 3 \\ &\quad = AO(x, 2, AO(x + 1, 2, 2)) - 3 && \text{by definition} \\ &\quad = AO(x, 2, 4) - 3 && \text{by Exercise 49} \\ &\quad = A(x, 1) && \text{by the inductive assumption on } x \\ &\quad = A(x + 1, 0) && \text{by (7.1.11).} \end{aligned}$$

Inductive Step (Case y implies case $y + 1$) Assume that

$$A(x + 1, y) = AO(x + 1, 2, y + 3) - 3.$$

We must prove that

$$A(x + 1, y + 1) = AO(x + 1, 2, y + 4) - 3.$$

Now

$$\begin{aligned} &AO(x + 1, 2, y + 4) - 3 \\ &\quad = AO(x, 2, AO(x + 1, 2, y + 3)) - 3 && \text{by definition} \\ &\quad = AO(x, 2, A(x + 1, y) + 3) - 3 && \text{by the inductive assumption on } y \\ &\quad = A(x, A(x + 1, y)) && \text{by the inductive assumption on } x \\ &\quad = A(x + 1, y + 1) && \text{by (7.1.12).} \end{aligned}$$

53. Suppose that we have n dollars. If we buy orange juice the first day, we have $n - 1$ dollars left, which may be spent in R_{n-1} ways. Similarly, if the first day we buy milk or beer, there are R_{n-2} ways to spend the remaining dollars. Since these cases are disjoint, $R_n = R_{n-1} + 2R_{n-2}$.

56. $S_3 = 1/2$, $S_4 = 3/4$

58. A function f from $X = \{1, \ldots, n\}$ into X will be denoted $(i_1, i_2, \ldots, i_n)$, which means that $f(k) = i_k$. The problem then is to count the number of ways to select $i_1, \ldots, i_n$ so that if i occurs, so do $1, 2, \ldots, i - 1$.

We shall count the number of such functions having exactly j 1's. Such functions can be constructed in two steps: Pick the positions for the j 1's; then place the other numbers. There are $C(n, j)$ ways to place the 1's. The remaining numbers must be selected so that if i appears, so do $1, \ldots, i - 1$. There are F_{n-j} ways to select the remaining numbers, since the remaining numbers must be selected from $\{2, \ldots, n\}$. Thus there are $C(n, j)F_{n-j}$ functions of the desired type having exactly j 1's. Therefore, the total number of functions from X into X having the property that if i is in the range of f, then so are $1, \ldots, i - 1$, is

$$\sum_{j=1}^{n} C(n, j)F_{n-j} = \sum_{j=1}^{n} C(n, n - j)F_{n-j} = \sum_{j=0}^{n-1} C(n, j)F_j.$$

61. $\{u_n\}$ is not a recurrence relation because, if n is odd and greater than 1, u_n is defined in terms of the *successor* u_{3n+1}. u_i, for $2 \leq i \leq 7$, is equal to one. As examples,

$$u_2 = u_1 = 1$$
$$u_3 = u_{10} = u_5 = u_{16} = u_8 = u_4 = u_2 = 1.$$

64. Use equation (7.7.4) to write

$$S(k, n) = \sum_{i=1}^{n} S(k - 1, i).$$

67. We use the terminology of Exercise 87, Section 6.2. Choose one of $n + 1$ people, say P. There are $s_{n,j-1}$ ways for P to sit alone. (Seat the other n people at the other $k - 1$ tables.) Next we count the number of arrangements in which P is not alone. Seat everyone but P at k tables. This can be done in $s_{n,k}$ ways. Now P can be seated to the right of someone in n ways. Thus there are $ns_{n,k}$ arrangements in which P is not alone. The recurrence relation now follows.

70. Let A_n denote the amount at the end of n years and let i be the interest rate expressed as a decimal. The discussion following Example 7.1.3 shows that

$$A_n = (1 + i)^n A_0.$$

The value of n required to double the amount satisfies

$$2A_0 = (1 + i)^n A_0 \quad \text{or} \quad 2 = (1 + i)^n.$$

If we take the natural logarithm (logarithm to the base e) of both sides of this equation, we obtain

$$\ln 2 = n \ln(1 + i).$$

Thus

$$n = \frac{\ln 2}{\ln(1 + i)}.$$

Since $\ln 2 = 0.6931472\ldots$ and $\ln(1 + i)$ is approximately equal to i for small values of i, n is approximately equal to $0.69\ldots/i$, which, in turn, is approximately equal to $70/r$.

72. 1, 3, 2; 2, 3, 1; $E_3 = 2$

75. We count the number of rise/fall permutations of $1, \ldots, n$ by considering how many have n in the second, fourth, $\ldots$, positions.

Suppose that n is in the second position. Since any of the remaining numbers is less than n, any of them may be placed in the first position. Thus we may select the number to be placed in the first position in $C(n-1, 1)$ ways and, after selecting it, we may arrange it in $E_1 = 1$ way. The last $n-2$ positions can be filled in E_{n-2} ways since any rise/fall permutation of the remaining $n-2$ numbers gives a rise/fall permutation of $1, \ldots, n$. Thus the number of rise/fall permutations of $1, \ldots, n$ with n in the second position is $C(n-1, 1)E_1E_{n-2}$.

Suppose that n is in the fourth position. We may select numbers to be placed in the first three positions in $C(n-1, 3)$ ways. After selecting the three items, we may arrange them in E_3 ways. The last $n-4$ numbers can be arranged in E_{n-4} ways. Thus the number of rise/fall permutations of $1, \ldots, n$ with n in the fourth position is $C(n-1, 3)E_3E_{n-4}$.

In general, the number of rise/fall permutations of $1, \ldots, n$ with n in the $(2j)$th position is

$$C(n-1, 2j-1)E_{2j-1}E_{n-2j}.$$

Summing over all j gives the desired recurrence relation.

Section 7.2 Review

1. Use the recurrence relation to write the nth term in terms of certain of its predecessors. Then successively use the recurrence relation to replace each of the resulting terms by certain of their predecessors. Continue until an explicit formula is obtained.
2. An nth-order, linear homogeneous recurrence relation with constant coefficients is a recurrence relation of the form

$$a_n = c_1a_{n-1} + c_2a_{n-2} + \cdots + c_ka_{n-k}.$$

3. $a_n = 6a_{n-1} - 8a_{n-2}$
4. To solve

$$a_n = c_1a_{n-1} + c_2a_{n-2},$$

first solve the equation

$$t^2 = c_1t + c_2$$

for t. Suppose that the roots are t_1 and t_2 and that $t_1 \neq t_2$. Then the general solution is of the form

$$a_n = bt_1^n + dt_2^n,$$

where b and d are constants. The values of the constants can be obtained from the initial conditions.

If $t_1 = t_2 = t$, the general solution is of the form

$$a_n = bt^n + dnt^n,$$

where again b and d are constants. The values of the constants can again be obtained from the initial conditions.

Section 7.2

1. Yes; order 1
4. No
7. No
10. Yes; order 3
11. $a_n = 2(-3)^n$
15. $a_n = 2^{n+1} - 4^n$
18. $a_n = (2^{2-n} + 3^n)/5$
21. $a_n = 2(-4)^n + 3n(-4)^n$
24. $R_n = [(-1)^n + 2^{n+1}]/3$
28. Let d_n denote the deer population at time n. The initial condition is $d_0 = 0$. The recurrence relation is

$$d_n = 100n + 1.2d_{n-1}, \quad n > 0.$$

$$\begin{aligned}
d_n &= 100n + 1.2d_{n-1} = 100n + 1.2[100(n-1) + 1.2d_{n-2}]\\
&= 100n + 1.2 \cdot 100(n-1) + 1.2^2 d_{n-2}\\
&= 100n + 1.2 \cdot 100(n-1)\\
&\quad + 1.2^2[100(n-2) + 1.2d_{n-3}]\\
&= 100n + 1.2 \cdot 100(n-1)\\
&\quad + 1.2^2 \cdot 100(n-2) + 1.2^3 d_{n-3}\\
&\ \ \vdots\\
&= \sum_{i=0}^{n-1} 1.2^i \cdot 100(n-i) + 1.2^n d_0\\
&= \sum_{i=0}^{n-1} 1.2^i \cdot 100(n-i)\\
&= 100n \sum_{i=0}^{n-1} 1.2^i - 1.2 \cdot 100 \sum_{i=1}^{n-1} i \cdot 1.2^{i-1}\\
&= \frac{100n(1.2^n - 1)}{1.2 - 1}\\
&\quad - 120\frac{(n-1)1.2^n - n1.2^{n-1} + 1}{(1.2-1)^2}, \qquad n > 0.
\end{aligned}$$

29. From $p_{n-1} = \frac{1}{2}p_n + \frac{1}{2}p_{n-2}$, we obtain $p_n = 2p_{n-1} - p_{n-2}$.
32. $p_n = n/(S+T)$
36. Set $b_n = a_n/n!$ to obtain $b_n = -2b_{n-1} + 3b_{n-2}$. Solving gives $a_n = n!\,b_n = (n!/4)[5 - (-3)^n]$.
39. We establish the inequality by using induction on n.

The base cases $n = 1$ and $n = 2$ are left to the reader. Now assume that the inequality is true for values less than $n+1$. Then

$$\begin{aligned}
f_{n+2} &= f_{n+1} + f_n\\
&\geq \left(\frac{1+\sqrt{5}}{2}\right)^{n-1} + \left(\frac{1+\sqrt{5}}{2}\right)^{n-2}\\
&= \left(\frac{1+\sqrt{5}}{2}\right)^{n-2}\left(\frac{1+\sqrt{5}}{2} + 1\right)\\
&= \left(\frac{1+\sqrt{5}}{2}\right)^{n-2}\left(\frac{1+\sqrt{5}}{2}\right)^{2}\\
&= \left(\frac{1+\sqrt{5}}{2}\right)^{n},
\end{aligned}$$

and the Inductive Step is complete.

41. $a_n = b2^n + d4^n + 1$

44. $a_n = b/2^n + d3^n - (4/3)2^n$

47. The argument is identical to that given in Theorem 7.2.11.

50. Recursively invoking this algorithm to move the $n - k_n$ disks at the top of peg 1 to peg 2 takes $T(n - k_n)$ moves. Moving the k_n disks on peg 1 to peg 4 requires $2^{k_n} - 1$ moves (see Example 7.2.4). Recursively invoking this algorithm to move the $n - k_n$ disks on peg 2 to peg 4 again takes $T(n - k_n)$ moves. The recurrence relation now follows.

53. From the inequality

$$\frac{k_n(k_n + 1)}{2} \leq n,$$

we can deduce $k_n \leq \sqrt{2n}$. Since

$$n - k_n \leq \frac{k_n(k_n + 1)}{2},$$

it follows that $r_n \leq k_n$. Therefore,

$$\begin{aligned} T(n) &= (k_n + r_n - 1)2^{k_n} + 1 \\ &< 2k_n 2^{k_n} + 1 \\ &\leq 2\sqrt{2n}2^{\sqrt{2n}} + 1 \\ &= O(4^{\sqrt{n}}). \end{aligned}$$

Section 7.3 Review

1. Let b_n denote the time required for input of size n. Simulate the execution of the algorithm and count the time required by the various steps. Then b_n is equal to the sum of the times required by the various steps.

2. Selection sort selects the largest element, places it last, and then recursively sorts the remaining sequence.

3. $\Theta(n^2)$

4. Binary search examines the middle item in the sequence. If the middle item is the desired item, binary search terminates. Otherwise, binary search compares the middle item with the desired item. If the desired item is less than the middle item, binary search recursively searches in the left half of the sequence. If the desired item is greater than the middle item, binary search recursively searches in the right half of the sequence. The input must be sorted.

5. If a_n is the worst-case time for input of size n, $a_n = 1 + a_{\lfloor n/2 \rfloor}$.

6. $\Theta(\lg n)$

7. Merge maintains two pointers to elements in the two input sequences. Initially the pointers reference the first elements in the sequences. Merge copies the smaller element to the output and moves the pointer to the next element in the sequence that contains the element just copied. It then repeats this process. When a pointer moves off the end of one of the sequences, merge concludes by copying the rest of the other sequence to the output. Both input sequences must be sorted.

8. $\Theta(n)$, where n is the sum of the lengths of the input sequences

9. Merge sort first divides the input into two nearly equal parts. It then recursively sorts each half and merges the halves to produce sorted output.

10. $a_n = a_{\lfloor n/2 \rfloor} + a_{\lfloor (n+1)/2 \rfloor} + n - 1$

11. If the input size is a power of two, the size is always divisible by 2 and the floors vanish.

12. An arbitrary input size falls between two powers of two. Since we know the worst-case time when the input size is a power of two, we may bound the worst-case time for input of arbitrary size by the worst-case times for inputs whose sizes are the powers of two that bound it.

13. $\Theta(n \lg n)$

Section 7.3

1. At line 2, since $i > j$ $(1 > 5)$ is false, we proceed to line 4, where we set k to 3. At line 5, since *key* ('G') is not equal to s_3 ('J'), we proceed to line 7. At line 7, $key < s_k$ ('G' < 'J') is true, so at line 8 we set j to 2. We then invoke this algorithm with $i = 1$, $j = 2$ to search for *key* in

$$s_1 = \text{'}C\text{'}, \qquad s_2 = \text{'}G\text{'}.$$

At line 2, since $i > j$ $(1 > 2)$ is false, we proceed to line 4, where we set k to 1. At line 5, since *key* ('G') is not equal to s_1 ('C'), we proceed to line 7. At line 7, $key < s_k$ ('G' < 'C') is false, so at line 10 we set i to 2. We then invoke this algorithm with $i = j = 2$ to search for *key* in

$$s_2 = \text{'}G\text{'}.$$

At line 2, since $i > j$ $(2 > 2)$ is false, we proceed to line 4, where we set k to 2. At line 5, since *key* ('G') is equal to s_2 ('G'), we return 2, the index of *key* in the sequence s.

4. At line 2, since $i > j$ $(1 > 5)$ is false, we proceed to line 4, where we set k to 3. At line 5, since *key* ('Z') is not equal to s_3 ('J'), we proceed to line 7. At line 7, $key < s_k$ ('Z' < 'J') is false, so at line 10 we set i to 4. We then invoke this algorithm with $i = 4$, $j = 5$ to search for *key* in

$$s_4 = \text{'}M\text{'}, \qquad s_5 = \text{'}X\text{'}.$$

At line 2, since $i > j$ $(4 > 5)$ is false, we proceed to line 4, where we set k to 4. At line 5, since *key* ('Z') is not equal to s_4 ('M') we proceed to line 7. At line 7, $key < s_k$ ('Z' < 'M') is false, so at line 10 we set i to 5. We then invoke this algorithm with $i = j = 5$ to search for *key* in

$$s_5 = \text{'}X\text{'}.$$

At line 2, since $i > j$ $(5 > 5)$ is false, we proceed to line 4, where we set k to 5. At line 5, since *key* ('Z') is not equal to s_5 ('X'), we proceed to line 7. At line 7, $key < s_k$ ('Z' < 'X') is false, so at line 10 we set i to 6. We then invoke this algorithm with $i = 6$, $j = 5$.

At line 2, since $i > j$ $(6 > 5)$ is true, we return 0 to indicate that we failed to find *key*.

7. Consider the input 10, 4, 2 and $key = 10$.

10. The idea is to repeatedly divide the sequence as nearly as possible into two parts and retain the part that might contain the key. Only after obtaining a subsequence of length 1 or

2, do we test whether the subsequence contains the key. The following algorithm implements this design.

```
binary_search_nonrecurs(s, n, key) {
   i = 1
   j = n
   // the body of the loop executes only if the subsequence
   // s_i, ..., s_j has length greater than or equal to 3
   while (i < j − 1) {
      k = ⌊(i + j)/2⌋
      if (s_k < key)
         i = k + 1
      else
         j = k
   }
   for k = i to j
      if (s_k == key)
         return k
   return 0
}
```

We first prove that if a sequence of length n is input to the while loop, where n is a power of 2, say $n = 2^m$, $m \geq 2$, the loop iterates $m - 1$ times. The proof is by induction on m. The Basis Step is $m = 2$. In this case $n = 4$. Assuming that $i = 1$ and $n = 4$, in the while loop, k is first set to 2. Then either i is set to 3 or j is set to 2. Thus the loop does not execute again. Therefore the loop iterates $1 = m - 1$ time. The Basis Step is complete.

Now suppose that if a sequence of length $n = 2^m$ is input to the while loop, the loop iterates $m - 1$ times. Suppose that $n = 2^{m+1}$. Assuming that $i = 1$ and $n = 2^{m+1}$, in the while loop, k is first set to 2^m. Then either i is set to $2^m + 1$ or j is set to 2^m. Thus at the next iteration of the loop, a sequence of length 2^m is processed. By the inductive assumption, the loop iterates an additional $m - 1$ times. Therefore the loop iterates a total of m times. The Inductive Step is complete.

Next we prove that if a sequence of length n, where n satisfies $2^{m-1} < n \leq 2^m$, $m \geq 2$, is input to the while loop, the loop iterates at most $m - 1$ times. The proof is by induction on m. The Basis Step is $m = 2$. In this case we have $2 < n \leq 4$. Thus n is either 3 or 4. In the preceding paragraphs, we proved that if $n = 4$ the loop iterates one time. If $n = 3$, it is easy to check that the loop iterates one time. The Basis Step is complete.

Now assume that if a sequence of length n, where n satisfies $2^{m-1} < n \leq 2^m$, $m \geq 2$, is input to the while loop, the loop iterates at most $m - 1$ times. Suppose that n satisfies $2^m < n \leq 2^{m+1}$. When n is even, the sequence is divided evenly and the next sequence processed by the loop has length $n/2$. Since $n/2$ satisfies $2^{m-1} < n/2 \leq 2^m$, by the inductive assumption the loop iterates at most $m - 1$ more times. When n is odd, the sequence is divided into two parts—one part of length $(n-1)/2$ and the other of length $(n + 1)/2$. Since n is odd, $2^m < n < 2^{m+1}$. Therefore $2^m < n + 1 \leq 2^{m+1}$. Thus $2^{m-1} < (n+1)/2 \leq 2^m$. In this case, the inductive assumption tells us that the loop iterates at most $m - 1$ more times. We also have $2^m \leq n - 1 < 2^{m+1}$ and $2^{m-1} \leq (n - 1)/2 < 2^m$. If $2^{m-1} < (n - 1)/2$, we may use the inductive assumption to conclude that the loop iterates at most $m - 1$ more times. If $2^{m-1} = (n - 1)/2$, we may use the result proved just after the algorithm to conclude that the loop iterates $m - 2$ more times. In every case the loop iterates at most $m - 1$ more times. Together with the first iteration, we conclude that if n satisfies $2^m < n \leq 2^{m+1}$, the while loop iterates at most m times. The Inductive Step is complete.

Suppose that n satisfies $2^{m-1} < n \leq 2^m$. Then the while loop iterates at most $m - 1$ times. This accounts for $m - 1$ tests of the form $s_k < key$. At the for loop, either $i = j$ or $i = j + 1$. Thus there are at most two additional comparisons (of the form $s_k == key$). Thus if n satisfies $2^{m-1} < n \leq 2^m$, the algorithm uses at most $m + 1$ comparisons. Since $2^{m-1} < n \leq 2^m$, $m - 1 < \lg n \leq m$. Therefore $\lceil \lg n \rceil = m$. Thus the algorithm uses at most $1 + m = 1 + \lceil \lg n \rceil$ comparisons.

13. The algorithm is not correct. If s is a sequence of length 1, $s_1 = 9$, and $key = 8$, the algorithm does not terminate.

16. The algorithm is correct. The worst-case time is $\Theta(\log n)$.

18. Algorithm B is superior if $2 \leq n \leq 15$. (For $n = 1$ and $n = 16$, the algorithms require equal numbers of comparisons.)

21. Suppose that the sequences are $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$. (a) $a_1 < b_1 < a_2 < b_2 < \cdots$ (b) $a_n < b_1$

24. 11

28. Algorithm 7.3.11 computes a^n by using the formula $a^n = a^m a^{n-m}$.

29. $b_n = b_{\lfloor n/2 \rfloor} + b_{\lfloor (n+1)/2 \rfloor} + 1$, $b_1 = 0$

30. $b_2 = 1, b_3 = 2, b_4 = 3$ **31.** $b_n = n - 1$

32. We prove the formula by using mathematical induction. The Basis Step, $n = 1$, has already been established.

Assume that $b_k = k - 1$ for all $k < n$. We show that $b_n = n - 1$. Now

$$\begin{aligned} b_n &= b_{\lfloor n/2 \rfloor} + b_{\lfloor (n+1)/2 \rfloor} + 1 \\ &= \left\lfloor \frac{n}{2} \right\rfloor - 1 + \left\lfloor \frac{n+1}{2} \right\rfloor - 1 + 1 \end{aligned}$$

by the inductive assumption

$$= \left\lfloor \frac{n}{2} \right\rfloor + \left\lfloor \frac{n+1}{2} \right\rfloor - 1 = n - 1.$$

45. If $n = 1$, then $i = j$ and we return before reaching line 6b, 10, or 14. Therefore, $b_1 = 0$. If $n = 2$, then $j = i + 1$. There is one comparison at line 6b and we return before reaching line 10 or 14. Therefore, $b_2 = 1$.

46. $b_3 = 3, b_4 = 4$

47. When $n > 2$, $b_{\lfloor (n+1)/2 \rfloor}$ comparisons are required for the first recursive call and $b_{\lfloor n/2 \rfloor}$ comparisons are required for the second recursive call. Two additional comparisons are required at lines 10 and 14. The recurrence relation now follows.

48. Suppose that $n = 2^k$. Then (7.3.12) becomes

$$b_{2^k} = 2b_{2^{k-1}} + 2.$$

Now

$$\begin{aligned} b_{2^k} &= 2b_{2^{k-1}} + 2 \\ &= 2[2b_{2^{k-2}} + 2] + 2 \\ &= 2^2 b_{2^{k-2}} + 2^2 + 2 = \cdots \\ &= 2^{k-1} b_{2^1} + 2^{k-1} + 2^{k-2} + \cdots + 2 \\ &= 2^{k-1} + 2^{k-1} + \cdots + 2 \\ &= 2^{k-1} + 2^k - 2 \\ &= n - 2 + \frac{n}{2} = \frac{3n}{2} - 2. \end{aligned}$$

49. We use the following fact, which can be verified by considering the cases x even and x odd:

$$\left\lceil \frac{3x}{2} - 2 \right\rceil + \left\lceil \frac{3(x+1)}{2} - 2 \right\rceil = 3x - 2 \text{ for } x = 1, 2, \ldots.$$

Let a_n denote the number of comparisons required by the algorithm in the worst case. The cases $n = 1$ and $n = 2$ may be directly verified. (The case $n = 2$ is the Basis Step.)

Inductive Step Assume that $a_k \le \lceil (3k/2) - 2 \rceil$ for $2 \le k < n$. We must show that the inequality holds for $k = n$.

If n is odd, the algorithm partitions the array into subclasses of sizes $(n-1)/2$ and $(n+1)/2$. Now

$$\begin{aligned} a_n &= a_{(n-1)/2} + a_{(n+1)/2} + 2 \\ &\le \left\lceil \frac{(3/2)(n-1)}{2} - 2 \right\rceil \\ &\quad + \left\lceil \frac{(3/2)(n+1)}{2} - 2 \right\rceil + 2 \\ &= \frac{3(n-1)}{2} - 2 + 2 = \frac{3n}{2} - \frac{3}{2} \\ &= \left\lceil \frac{3n}{2} - 2 \right\rceil. \end{aligned}$$

The case n even is treated similarly.

58. $\Theta(n)$

59. If $n = 1$, *sort* just returns; therefore, all of the zeros precede all of the ones. The Basis Step is proved.

Assume that for input of size $n-1$, after *sort* is invoked all of the zeros precede all of the ones. Suppose that *sort* is invoked with input of size n. If the first element is a one, it is swapped with the last element. *sort* is then called recursively on the first $n-1$ elements. By the inductive assumption, within the first $n-1$ elements all of the zeros precede all of the ones. Since the last element is a one, all of the zeros precede all of the ones for all n elements. If the first element is a zero, *sort* is called recursively on the last $n-1$ elements. By the inductive assumption, within the last $n-1$ elements all of the zeros precede all of the ones. Since the first element is a zero, all of the zeros precede all of the ones for all n elements. In either case, *sort* does produce as output a rearranged version of the input sequence in which all of the zeros precede all of the ones, and the Inductive Step is complete.

64. If $n = 2^k$,

$$a_{2^k} = 3a_{2^{k-1}} + 2^k,$$

so

$$\begin{aligned} a_n = a_{2^k} &= 3a_{2^{k-1}} + 2^k \\ &= 3[3a_{2^{k-2}} + 2^{k-1}] + 2^k \\ &= 3^2 a_{2^{k-2}} + 3 \cdot 2^{k-1} + 2^k \\ &\ \ \vdots \\ &= 3^k a_{2^0} + 3^{k-1} \cdot 2^1 + 3^{k-2} \cdot 2^2 + \cdots \\ &\quad + 3 \cdot 2^{k-1} + 2^k \\ &= 3^k + 2(3^k - 2^k) \qquad (*) \\ &= 3 \cdot 3^k - 2 \cdot 2^k \\ &= 3 \cdot 3^{\lg n} - 2n. \end{aligned}$$

Line $(*)$ results from the equation

$$(a-b)(a^{k-1}b^0 + a^{k-2}b^1 + \cdots + a^1 b^{k-2} + a^0 b^{k-1}) = a^k - b^k$$

with $a = 3$ and $b = 2$.

66. $b_n = b_{\lfloor (1+n)/2 \rfloor} + b_{\lfloor n/2 \rfloor} + 3$

69. $b_n = 4n - 3$

72. We will show that $b_n \le b_{n+1}$, $n = 1, 2, \ldots$. We have the recurrence relation

$$b_n = b_{\lfloor (1+n)/2 \rfloor} + b_{\lfloor n/2 \rfloor} + c_{\lfloor (1+n)/2 \rfloor, \lfloor n/2 \rfloor}.$$

Basis Step $b_2 = 2b_1 + c_{1,1} \ge 2b_1 \ge b_1$

Inductive Step Assume that the statement holds for $k < n$. In case n is even, we have $b_n = 2b_{n/2} + c_{n/2,n/2}$; so

$$\begin{aligned} b_{n+1} &= b_{(n+2)/2} + b_{n/2} + c_{(n+2)/2,n/2} \\ &\ge b_{n/2} + b_{n/2} + c_{n/2,n/2} = b_n. \end{aligned}$$

The case n is odd is similar.

74.

```
ex74(s, i, j) {
   if (i == j)
      return
   m = ⌊(i + j)/2⌋
   ex74(s, i, m)
   ex74(s, m + 1, j)
   combine(s, i, m, j)
}
```

77. We prove the inequality by using mathematical induction.

Basis Step $a_1 = 0 \le 0 = b_1$

Inductive Step Assume that $a_k \le b_k$ for $k < n$. Then

$$\begin{aligned} a_n &\le a_{\lfloor n/2 \rfloor} + a_{\lfloor (n+1)/2 \rfloor} + 2 \lg n \\ &\le b_{\lfloor n/2 \rfloor} + b_{\lfloor (n+1)/2 \rfloor} + 2 \lg n = b_n. \end{aligned}$$

80. Let $c = a_1$. If n is a power of m, say $n = m^k$, then

$$\begin{aligned} a_n = a_{m^k} &= a_{m^{k-1}} + d \\ &= [a_{m^{k-2}} + d] + d \\ &= a_{m^{k-2}} + 2d \\ &\ \ \vdots \\ &= a_{m^0} + kd = c + kd. \end{aligned}$$

An arbitrary value of n falls between two powers of m, say

$$m^{k-1} < n \le m^k.$$

This last inequality implies that

$$k - 1 < \log_m n \le k.$$

Since the sequence a is nondecreasing,

$$a_{m^{k-1}} \le a_n \le a_{m^k}.$$

Now

$$\begin{aligned}\Omega(\log_m n) &= c + (-1 + \log_m n)d \le c + (k-1)d \\ &= a_{m^{k-1}} \le a_n\end{aligned}$$

and

$$\begin{aligned}a_n &\le a_{m^k} = c + kd \\ &\le c + (1 + \log_m n)d = O(\log_m n).\end{aligned}$$

Thus $a_n = \Theta(\log_m n)$. By Example 4.3.6, $a_n = \Theta(\lg n)$.

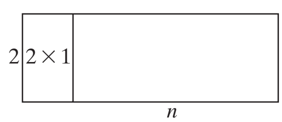
Chapter 7 Self-Test

1. (a) 3, 5, 8, 12 (b) $a_1 = 3$ (c) $a_n = a_{n-1} + n$

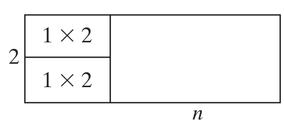
2. $A_n = (1.17)A_{n-1}$, $A_0 = 4000$

3. Let X be an n-element set and choose $x \in X$. Let k be a fixed integer, $0 \le k \le n-1$. We can select a k-element subset Y of $X - \{x\}$ in $C(n-1, k)$ ways. Having done this, we can partition Y in P_k ways. This partition together with $X - Y$ partitions X. Since all partitions of X can be generated in this way, we obtain the desired recurrence relation.

4. If the first domino is placed as shown, there are a_{n-1} ways to cover the $2 \times (n-1)$ board that remains.



If the first two dominoes are placed as shown, there are a_{n-2} ways to cover the $2 \times (n-2)$ board that remains.



It follows that $a_n = a_{n-1} + a_{n-2}$.

By inspection, $a_1 = 1$ and $a_2 = 2$. Since $\{a_n\}$ satisfies the same recurrence relation as the Fibonacci sequence and $a_1 = f_2$ and $a_2 = f_3$, it follows that $a_i = f_{i+1}$ for $i = 1, 2, \ldots$.

5. Yes

6. $a_n = 2(-2)^n - 4n(-2)^n$

7. $a_n = 3 \cdot 5^n + (-2)^n$

8. Consider a string of length n that contains an even number of 1's that begins with 0. The string that follows the 0 may be any string of length $n-1$ that contains an even number of 1's, and there are c_{n-1} such strings. A string of length n that contains an even number of 1's that begins with 2 can be followed by any string of length $n-1$ that contains an even number of 1's, and there are c_{n-1} such strings. A string of length n that contains an even number of 1's that begins with 1 can be followed by any string of length $n-1$ that contains an odd number of 1's. Since there are 3^{n-1} strings altogether of length $n-1$ and c_{n-1} of these contain an even number of 1's, there are $3^{n-1} - c_{n-1}$ strings of length $n-1$ that contain an odd number of 1's. It follows that

$$c_n = 2c_{n-1} + 3^{n-1} - c_{n-1} = c_{n-1} + 3^{n-1}.$$

An initial condition is $c_1 = 2$, since there are two strings (0 and 2) that contain an even number (namely, zero) of 1's.

We may solve the recurrence relation by iteration:

$$\begin{aligned}c_n &= c_{n-1} + 3^{n-1} = c_{n-2} + 3^{n-2} + 3^{n-1} \\ &\ \vdots \\ &= c_1 + 3^1 + 3^2 + \cdots + 3^{n-1} \\ &= 2 + \frac{3^n - 3}{3 - 1} = \frac{3^n + 1}{2}.\end{aligned}$$

9. $b_n = b_{n-1} + 1$, $b_0 = 0$

10. $b_1 = 1$, $b_2 = 2$, $b_3 = 3$

11. $b_n = n$

12. $n(n+1)/2 = O(n^2)$. The given algorithm is faster than the straightforward technique and is, therefore, preferred.

Section 8.1 Review

1. An undirected graph consists of a set V of vertices and a set E of edges such that each edge $e \in E$ is associated with an unordered pair of vertices.

2. Friendship can be modeled by an undirected graph by letting the vertices denote the people and placing an edge between two people if they are friends.

3. A directed graph consists of a set V of vertices and a set E of edges such that each edge $e \in E$ is associated with an ordered pair of vertices.

4. Precedence can be modeled by a directed graph by letting the vertices denote the tasks and placing a directed edge from task t_i to task t_j if t_i must be completed before t_j.

5. If edge e is associated with vertices v and w, e is said to be incident on v and w.

6. If edge e is associated with vertices v and w, v and w are said to be incident on e.

7. If edge e is associated with vertices v and w, v and w are said to be adjacent.

8. Parallel edges are edges that are incident on the same pair of vertices.

9. An edge incident on a single vertex is called a loop.

10. A vertex that is not incident on any edge is called an isolated vertex.

11. A simple graph is a graph with neither loops nor parallel edges.

12. A weighted graph is a graph with numbers assigned to the edges.

13. A map with distances can be modeled as a weighted graph. The vertices are the cities, the edges are the roads between the cities, and the numbers on the edges are the distances between the cities.

14. The length of a path in a weighted graph is the sum of the weights of its edges.

15. A similarity graph has a dissimilarity function s where $s(v, w)$ measures the dissimilarity of vertices v and w.

16. The n-cube has 2^n vertices labeled $0, 1, \ldots, 2^n - 1$. An edge connects two vertices if the binary representation of their labels differs in exactly one bit.

17. A serial computer executes one instruction at a time.

18. A serial algorithm executes one instruction at a time.

19. A parallel computer can execute several instructions at a time.

20. A parallel algorithm can execute several instructions at a time.

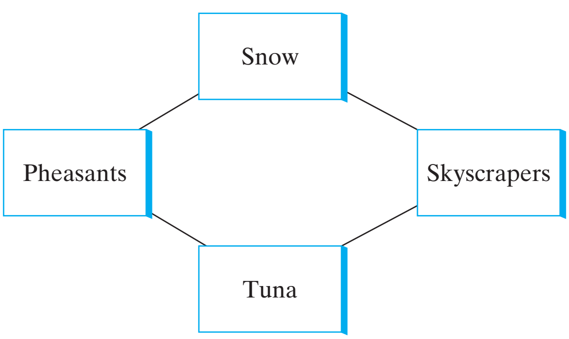
21. The complete graph on n vertices has one edge between each distinct pair of vertices. It is denoted K_n.

22. A graph $G = (V, E)$ is bipartite if there exist subsets V_1 and V_2 (either possibly empty) of V such that $V_1 \cap V_2 = \varnothing$, $V_1 \cup V_2 = V$, and each edge in E is incident on one vertex in V_1 and one vertex in V_2.

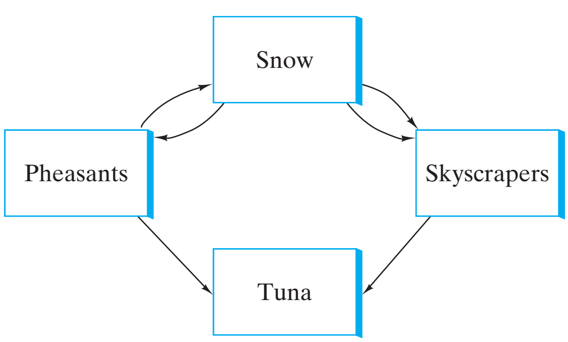
23. The complete bipartite graph on m and n vertices has disjoint vertex sets V_1 with m vertices and V_2 with n vertices in which the edge set consists of all edges of the form (v_1, v_2) with $v_1 \in V_1$ and $v_2 \in V_2$.

Section 8.1

1. The graph is an undirected, simple graph.



4. The graph is a directed, nonsimple graph.

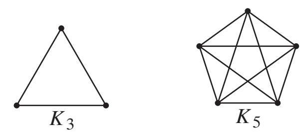


5. Since an odd number of edges touch some vertices (c and d), there is no path from a to a that passes through each edge exactly one time.

8. $(a, c, e, b, c, d, e, f, d, b, a)$

11. $V = \{v_1, v_2, v_3, v_4\}$. $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$. e_1 and e_6 are parallel edges. e_5 is a loop. There are no isolated vertices. G is not a simple graph. e_1 is incident on v_1 and v_2.

14.

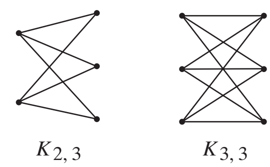


17. Bipartite. $V_1 = \{v_1, v_2, v_5\}$, $V_2 = \{v_3, v_4\}$.

20. Not bipartite

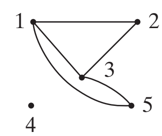
23. Bipartite. $V_1 = \{v_1\}$, $V_2 = \{v_2, v_3\}$.

24.

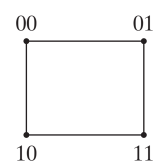


27. (b, c, a, d, e)

32. Two classes

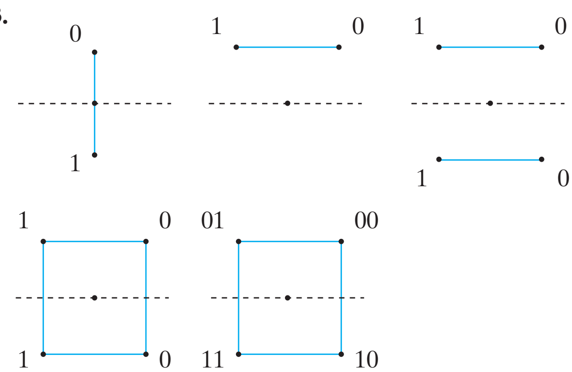


37.

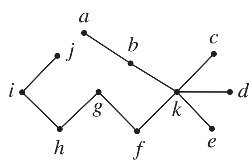


40. n

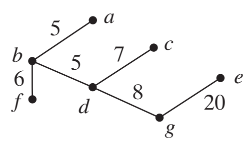
43.



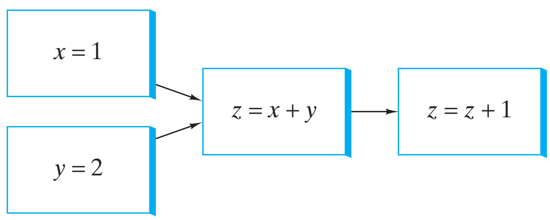
46.



49.



50.



53. f is not one-to-one. Let G_1 be the graph with vertex set $\{1, 2, 3\}$ and edge set $\{(1, 2)\}$, and let G_2 be the graph with vertex set $\{1, 2, 3, 4\}$ and edge set $\{(1, 2)\}$. Then $G_1 \neq G_2$, but $f(G_1) = 1 = f(G_2)$.

f is onto. Let n be a nonnegative integer. If $n = 0$, let G be the graph with vertex set $\{1, 2, 3\}$ and edge set $\varnothing$. Then $f(G) = 0 = n$. If $n > 0$, let G be the graph with vertex set $\{1, 2, \ldots, n, n + 1\}$ and edge set

$$\{(1, 2), (2, 3), \ldots, (n, n + 1)\}.$$

Then $f(G) = n$. Therefore f is onto.

Section 8.2 Review

1. A path is an alternating sequence of vertices and edges

$$(v_0, e_1, v_1, e_2, v_2, \ldots, v_{n-1}, e_n, v_n),$$

in which edge e_i is incident on vertices v_{i-1} and v_i for $i = 1, \ldots, n$.

2. A simple path is a path with no repeated vertices.

3. (1, 2, 3, 1)

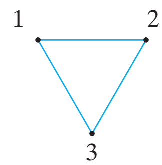
4. A cycle is path of nonzero length from v to v with no repeated edges.

5. A simple cycle is a cycle from v to v in which, except for the beginning and ending vertices that are both equal to v, there are no repeated vertices.

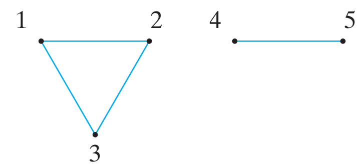
6. (1, 2, 3, 1, 4, 5, 1)

7. A graph is connected, if, given any vertices v and w, there is a path from v to w.

8.



9.



10. Let $G = (V, E)$ be a graph. (V', E') is a subgraph of G if $V' \subseteq V$, $E' \subseteq E$, and, for every edge $e' \in E'$, if e' is incident on v' and w', then $v', w' \in V'$.

11. The graph of Exercise 8 is a subgraph of the graph of Exercise 9.

12. Let G be a graph and let v be a vertex in G. The subgraph G' of G consisting of all edges and vertices in G that are contained in some path beginning at v is called the component of G containing v.

13. The graph of Exercise 8 is a component of the graph of Exercise 9.

14. One

15. The degree of vertex v is the number of edges incident on v.

16. An Euler cycle in a graph G is a cycle that includes all of the edges and all of the vertices of G.

17. A graph G has an Euler cycle if and only if G is connected and the degree of every vertex is even.

18. The graph of Exercise 8 has the Euler cycle (1, 2, 3, 1).

19. The graph of Exercise 9 does not have an Euler cycle because it is not connected.

20. The sum of the degrees of the vertices in a graph equals twice the number of edges in the graph.

21. Yes

22. The graph is connected and v and w are the only vertices having odd degree.

23. Yes

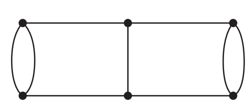
Section 8.2

1. Cycle, simple cycle

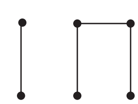
4. Cycle, simple cycle

7. Simple path

10.



13.



16. Suppose that there is such a graph with vertices a, b, c, d, e, f. Suppose that the degrees of a and b are 5. Since the graph is simple, the degrees of c, d, e, and f are each at least 2; thus there is no such graph.

19. (a, a), (b, c, g, b), (b, c, d, f, g, b), (b, c, d, e, f, g, b), (c, g, f, d, c), (c, g, f, e, d, c), (d, f, e, d)

22. Every vertex has degree 4.

24. $G_1 = (\{v_1\}, \varnothing)$

$G_2 = (\{v_2\}, \varnothing)$

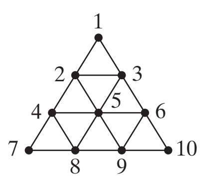
$G_3 = (\{v_1, v_2\}, \varnothing)$

$G_4 = (\{v_1, v_2\}, \{e_1\})$

27. There are 17 subgraphs. **28.** No Euler cycle

31. No Euler cycle

34. For



an Euler cycle is (10, 9, 6, 5, 9, 8, 5, 4, 8, 7, 4, 2, 5, 3, 2, 1, 3, 6, 10). The method generalizes.

37. $m = n = 2$ or $m = n = 1$

39. d and e are the only vertices of odd degree.

42. The argument is similar to that of the proof of Theorem 8.2.23.

45. True. In the path, for all repeated a,

$$(\ldots, a, \ldots, b, a, \ldots)$$

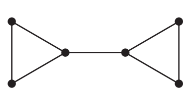
eliminate $a, \ldots, b$.

47. Suppose that $e = (v, w)$ is in a cycle. Then there is a path P from v to w not including e. Let x and y be vertices in $G - \{e\}$. Since G is connected, there is a path P' in G from v to w. Replace any occurrence of e in P' by P. The resulting path from v to w lies in $G - \{e\}$. Therefore, $G - \{e\}$ is connected.

50. The union of all connected subgraphs containing G' is a component.

53. Let G be a simple, disconnected graph with n vertices having the maximum number of edges. Show that G has two components. If one component has i vertices, show that the components are K_i and K_{n-i}. Use Exercise 11, Section 8.1, to find a formula for the number of edges in G as a function of i. Show that the maximum occurs when $i = 1$.

55.



58. Modify the proofs of Theorems 8.2.17 and 8.2.18.

61. Use Exercises 58 and 60.

64. We first count the number of paths

$$(v_0, v_1, \ldots, v_k)$$

of length $k \geq 1$. The first vertex v_0 may be chosen in n ways. Each subsequent vertex may be chosen in $n - 1$ ways (since it must be different from its predecessor). Thus the number of paths of length k is $n(n-1)^k$.

The number of paths of length k, $1 \leq k \leq n$, is

$$\begin{aligned}\sum_{k=1}^{n} n(n-1)^k &= n(n-1)\frac{(n-1)^k - 1}{(n-1) - 1} \\ &= \frac{n(n-1)[(n-1)^k - 1]}{n-2}.\end{aligned}$$

68. If v is a vertex in V, the path consisting of v and no edges is a path from v to v; thus vRv for every vertex v in V. Therefore, R is reflexive.

Suppose that vRw. Then there is a path $(v_0, \ldots, v_n)$, where $v_0 = v$ and $v_n = w$. Now $(v_n, \ldots, v_0)$ is a path from w to v, and thus wRv. Therefore, R is symmetric.

Suppose that vRw and wRx. Then there is a path P_1 from v to w and a path P_2 from w to x. Now P_1 followed by P_2 is a path from v to x, and thus vRx. Therefore, R is transitive.

Since R is reflexive, symmetric, and transitive on V, R is an equivalence relation on V.

70. 2

73. Let s_n denote the number of paths of length n from v_1 to v_1. We show that the sequences $s_1, s_2, \ldots$ and $f_1, f_2, \ldots$ satisfy the same recurrence relation, $s_1 = f_2$, and $s_2 = f_3$, from which it follows that $s_n = f_{n+1}$ for $n \geq 1$.

If $n = 1$, there is one path of length 1 from v_1 to v_1, namely the loop on v_1; thus, $s_1 = f_2$.

If $n = 2$, there are two paths of length 2 from v_1 to v_1: (v_1, v_1, v_1) and (v_1, v_2, v_1); thus, $s_2 = f_3$.

Assume that $n > 2$. Consider a path of length n from v_1 to v_1. The path must begin with the loop (v_1, v_1) or the edge (v_1, v_2).

If the path begins with the loop, the remainder of the path must be a path of length $n - 1$ from v_1 to v_1. Since there are s_{n-1} such paths, there are s_{n-1} paths of length n from v_1 to v_1 that begin $(v_1, v_1, \ldots)$.

If the path begins with the edge (v_1, v_2), the next edge in the path must be (v_2, v_1). The remainder of the path must be a path of length $n - 2$ from v_1 to v_1. Since there are s_{n-2} such paths, there are s_{n-2} paths of length n from v_1 to v_1 that begin $(v_1, v_2, v_1, \ldots)$.

Since any path of length $n > 2$ from v_1 to v_1 begins with the loop (v_1, v_1) or the edge (v_1, v_2), it follows that

$$s_n = s_{n-1} + s_{n-2}.$$

Because the sequences $s_1, s_2, \ldots$ and $f_1, f_2, \ldots$ satisfy the same recurrence relation, $s_1 = f_2$, and $s_2 = f_3$, it follows that $s_n = f_{n+1}$ for $n \geq 1$.

75. Suppose that every vertex has an out edge. Choose a vertex v_0. Follow an edge out of v_0 to a vertex v_1. (By assumption, such an edge exists.) Continue to follow an edge out of v_i to a vertex v_{i+1}. Since there are a finite number of vertices, we will eventually return to a previously visited vertex. At this point, we will have discovered a cycle, which is a contradiction. Therefore, a dag has at least one vertex with no out edges.

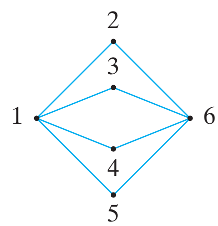
Section 8.3 Review

1. A Hamiltonian cycle in a graph G is a cycle that contains each vertex in G exactly once, except for the starting and ending vertex that appears twice.

2. The graph of Figure 8.3.9 has a Hamiltonian cycle and an Euler cycle. The Hamiltonian and Euler cycles are the graph itself.

3. The graph of Figure 8.3.2 has a Hamiltonian cycle, but not an Euler cycle. The Hamiltonian cycle is shown in Figure 8.3.3. The graph does not have an Euler cycle, because all of the vertices have odd degree.

4. The graph



has an Euler cycle because it is connected and every vertex has even degree. It does not have a Hamiltonian cycle. To prove that it does not have a Hamiltonian cycle, we argue by contradiction. Suppose that the graph has a Hamiltonian cycle. Then, because vertices 2, 3, 4, and 5 all have degree 2, all the edges in the graph would have to be included in a Hamiltonian cycle. Since the graph itself is not a cycle, we have a contradiction.

5. The graph consisting of two vertices and no edges has neither a Hamiltonian cycle nor an Euler cycle because it is not connected.

6. The traveling salesperson problem is: Given a weighted graph G, find a minimum-length Hamiltonian cycle in G. The Hamiltonian cycle problem simply asks for a Hamiltonian cycle—any Hamiltonian cycle will do. The traveling salesperson problem asks not just for a Hamiltonian cycle, but for one of minimum length.

7. A simple cycle

8. A Gray code is a sequence $s_1, s_2, \ldots, s_{2^n}$, where each s_i is a string of n bits, satisfying the following:

- Every n-bit string appears somewhere in the sequence.
- s_i and s_{i+1} differ in exactly one bit, $i = 1, \ldots, 2^n - 1$.
- s_{2^n} and s_1 differ in exactly one bit.

9. See Theorem 8.3.6.

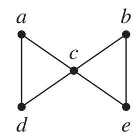
Section 8.3

1. $(d, a, e, b, c, h, g, f, j, i, d)$

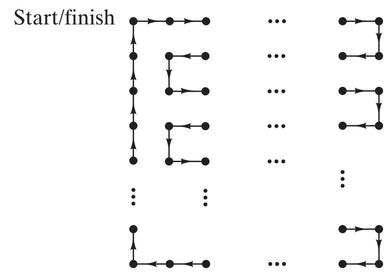
3. We would have to eliminate two edges each at b, d, i, and k, leaving $19 - 8 = 11$ edges. A Hamiltonian cycle would have 12 edges.

6. $(a, b, c, j, i, m, k, d, e, f, l, g, h, a)$

9.



12. If n is even and $m > 1$ or if m is even and $n > 1$, there is a Hamiltonian cycle. The sketch shows the solution in case n is even.



If $n = 1$ or if $m = 1$, there is no cycle and, in particular, there is no Hamiltonian cycle. Suppose that n and m are both odd and that the graph has a Hamiltonian cycle. Since there are nm vertices, this cycle has nm edges; therefore, the Hamiltonian cycle contains an odd number of edges. However, we note that in a Hamiltonian cycle, there must be as many "up" edges as "down" edges and as many "left" edges as "right" edges. Thus a Hamiltonian cycle must have an even number of edges. This contradiction shows that if n and m are both odd, the graph does not have a Hamiltonian cycle.

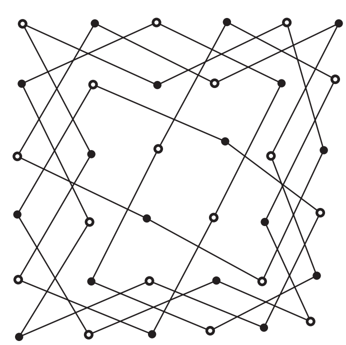
15. When $m = n$ and $n > 1$

18. Any cycle C in the n-cube has even length since the vertices in C alternate between an even and an odd number of 1's.

Suppose that the n-cube has a simple cycle of length m. We just observed that m is even. Now $m > 0$, by definition. Since the n-cube is a simple graph, $m \neq 2$. Therefore, $m \geq 4$.

Now suppose that $m \geq 4$ and m is even. Let G be the first $m/2$ members of the Gray code G_{n-1}. Then $0G$, $1G^R$ describes a simple cycle of length m in the n-cube.

21.



25. Yes. If $(v_1, \ldots, v_{n-1}, v_n)$, $v_1 = v_n$, is a Hamiltonian cycle, $(v_1, \ldots, v_{n-1})$ is a Hamiltonian path.

28. Yes, $(a, b, d, g, m, l, h, i, j, e, f, k, c)$

31. Yes, $(i, j, g, h, e, d, c, b, a, f)$

34. Yes, (a, c, d, f, g, e, b)

Section 8.4 Review

1. Label the start vertex 0 and all other vertices ∞. Let T be the set of all vertices. Choose $v \in T$ with minimum label and remove v from T. For each $x \in T$ adjacent to v, relabel x with the minimum of its current label and the label of $v + w(v, x)$, where $w(v, x)$ is the weight of edge (v, x). Repeat if $z \notin T$.

2. See Example 8.4.2.

3. See the proof of Theorem 8.4.3.

Section 8.4

1. 7; (a, b, c, f)

4. 7; (b, c, f, j)

6. An algorithm can be modeled after Example 8.4.2.

9. Modify Algorithm 8.4.1 so that it begins by assigning the weight ∞ to each nonexistent edge. The algorithm then continues as written. At termination, $L(z)$ will be equal to ∞ if there is no path from a to z.

Section 8.5 Review

1. Order the vertices and label the rows and columns of a matrix with the ordered vertices. The entry in row i, column j, $i \neq j$, is the number of edges incident on i and j. If $i = j$, the entry is twice the number of loops incident on i. The resulting matrix is the adjacency matrix of the graph.

2. The ijth entry in A^n is equal to the number of paths of length n from vertex i to vertex j.

3. Order the vertices and edges and label the rows of a matrix with the vertices and the columns with the edges. The entry in row v and column e is 1 if e is incident on v and 0 otherwise. The resulting matrix is the incidence matrix of the graph.

Section 8.5

1.

$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

4.

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

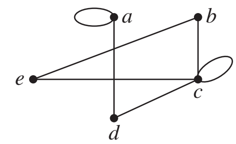
7.

$$\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{cccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

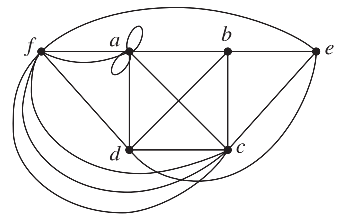
10.

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

13.



16.

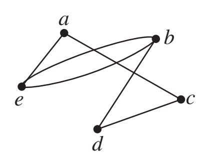


19. [For K_5]

$$\begin{pmatrix} 4 & 3 & 3 & 3 & 3 \\ 3 & 4 & 3 & 3 & 3 \\ 3 & 3 & 4 & 3 & 3 \\ 3 & 3 & 3 & 4 & 3 \\ 3 & 3 & 3 & 3 & 4 \end{pmatrix}$$

22. The graph is not connected.

24.



27. G is not connected.

28. Because of the symmetry of the graph, if v and w are vertices in K_5, there is the same number of paths of length n from v to v as there is from w to w. Thus all the diagonal elements of A^n are equal. Similarly, all the off-diagonal elements of A^n are equal.

31. If $n \geq 2$,

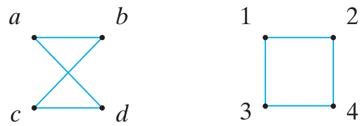
$$\begin{aligned} d_n &= 4a_{n-1} && \text{by Exercise 29} \\ &= 4\left(\frac{1}{5}\right)[4^{n-1} + (-1)^n] && \text{by Exercise 30.} \end{aligned}$$

The formula can be directly verified for $n = 1$.

Section 8.6 Review

1. Graphs G_1 and G_2 are isomorphic if there is a one-to-one, onto function f from the vertices of G_1 to the vertices of G_2 and a one-to-one, onto function g from the edges of G_1 to the edges of G_2, so that an edge e is incident on v and w in G_1 if and only if the edge $g(e)$ is incident on $f(v)$ and $f(w)$ in G_2.

2. The following graphs



are isomorphic. An isomorphism is given by $f(a) = 1$, $f(b) = 2$, $f(c) = 4$, $f(d) = 3$, and $g(a, b) = (1, 2)$, $g(b, c) = (2, 4)$, $g(c, d) = (4, 3)$, $g(d, a) = (3, 1)$.

3. The following graphs

are not isomorphic; the first graph has two vertices, but the second graph has three vertices.

4. A property P is an invariant if, whenever G_1 and G_2 are isomorphic graphs, if G_1 has property P, then G_2 also has property P.

5. To show that two graphs are not isomorphic, find an invariant that one graph has and the other does not have.

6. Two graphs are isomorphic if and only if for some orderings of their vertices, their adjacency matrices are equal.

7. A rectangular array of vertices

Section 8.6

1. Relative to the vertex orderings a, b, c, d, e, f, g for G_1, and $1, 3, 5, 7, 2, 4, 6$ for G_2, the adjacency matrices of G_1 and G_2 are equal.

4. Relative to the vertex orderings $a, b, c, d, e, f, g, h, i, j$ for G_1, and $5, 6, 1, 2, 7, 4, 10, 8, 3, 9$ for G_2, the adjacency matrices of G_1 and G_2 are equal.

7. The graphs are not isomorphic since they do not have the same number of vertices.

10. The graphs are not isomorphic since G_1 has a simple cycle of length 3 and G_2 does not.

13. The graphs are not isomorphic. The edge $(1, 4)$ in G_2 has $\delta(1) = 3$ and $\delta(4) = 3$ but there is no such edge in G_1 (see also Exercise 21).

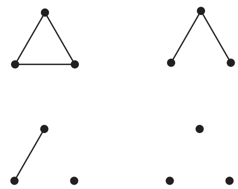
In Exercises 17–23, we use the notation of Definition 8.6.1.

17. If $(v_0, v_1, \ldots, v_k)$ is a simple cycle of length k in G_1, then $(f(v_0), f(v_1), \ldots, f(v_k))$ is a simple cycle of length k in G_2. [The vertices $f(v_i)$, $i = 1, \ldots, k-1$, are distinct, since f is one-to-one.]

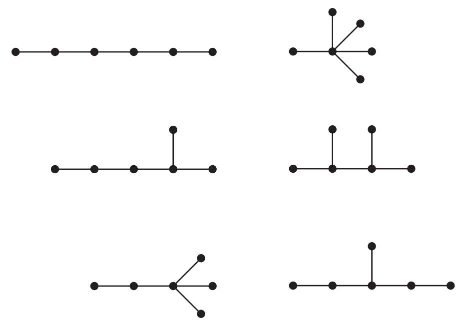
20. In the hint to Exercise 17, we showed that if $C = (v_0, v_1, \ldots, v_k)$ is a simple cycle of length k in G_1, then $(f(v_0), f(v_1), \ldots, f(v_k))$, which here we denote $f(C)$, is a simple cycle of length k in G_1. Let $C_1, C_2, \ldots, C_n$ denote the n simple cycles of length k in G_1. Then $f(C_1)$, $f(C_2), \ldots, f(C_n)$ are n simple cycles of length k in G_2. Moreover, since f is one-to-one, $f(C_1), f(C_2), \ldots, f(C_n)$ are distinct.

23. The property is an invariant. If $(v_0, v_1, \ldots, v_n)$ is an Euler cycle in G_1, then, since g is onto, $(f(v_0), f(v_1), \ldots, f(v_n))$ is an Euler cycle in G_2.

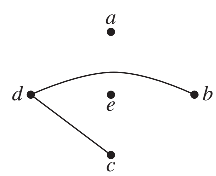
26.



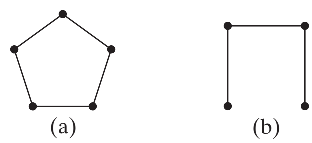
29.



31.



34.



37. Define $g((v, w)) = (f(v), f(w))$.

38. $f(a) = 1$, $f(b) = 2$, $f(c) = 3$, $f(d) = 2$

41. $f(a) = 1$, $f(b) = 2$, $f(c) = 3$, $f(d) = 1$

Section 8.7 Review

1. A graph that can be drawn in the plane without its edges crossing

2. A contiguous region

3. $f = e - v + 2$

4. Edges of the form (v, v_1) and (v, v_2), where v has degree 2 and $v_1 \neq v_2$

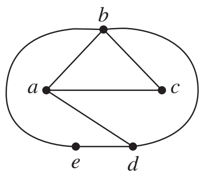
5. Given edges of the form (v, v_1) and (v, v_2), where v has degree 2 and $v_1 \neq v_2$, a series reduction deletes vertex v and replaces (v, v_1) and (v, v_2) by (v_1, v_2).

6. Two graphs are homeomorphic if they can be reduced to isomorphic graphs by performing a sequence of series reductions.

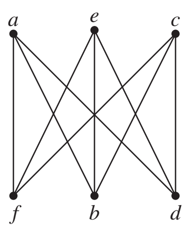
7. A graph is planar if and only if it does not contain a subgraph homeomorphic to K_5 or $K_{3,3}$.

Section 8.7

1.

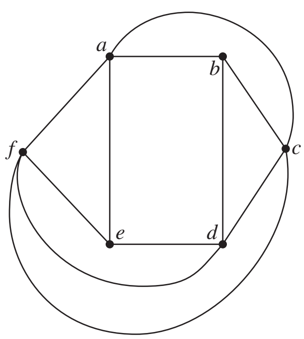


4.



is $K_{3,3}$

6. Planar

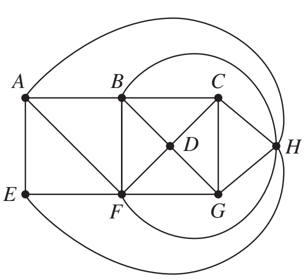


9. $2e = 2+2+2+3+3+3+4+4+5$,
so $e = 14$. $f = e - v + 2 = 14 - 9 + 2 = 7$

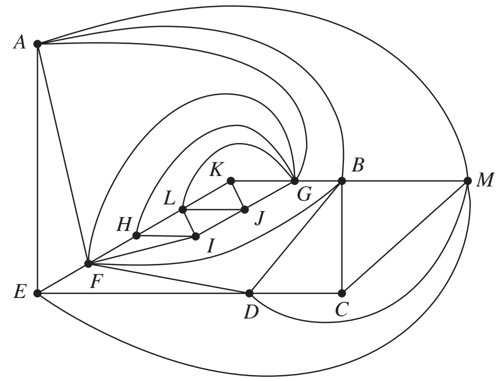
12. A graph with five or fewer vertices and a vertex of degree 2 is homeomorphic to a graph with four or fewer vertices. Such a graph cannot contain a homeomorphic copy of $K_{3,3}$ or K_5.

15. If K_5 is planar, $e \le 3v - 6$ becomes $10 \le 3 \cdot 5 - 6 = 9$.

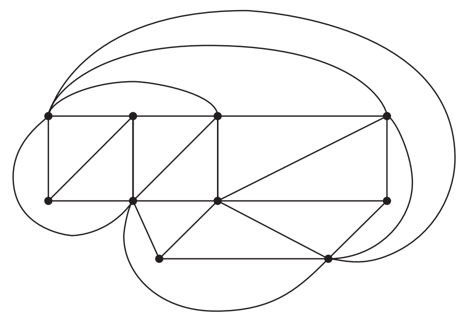
18.



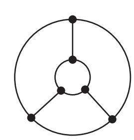
22.



25.



28. It contains



31. Assume that G does not have a vertex of degree 5. Show that $2e \ge 6v$. Now use Exercise 13 to deduce a contradiction.

Section 8.8 Review

1. Instant Insanity consists of four cubes each of whose faces is painted one of the four colors, red, white, blue, or green. The problem is to stack the cubes, one on top of the other, so that whether the stack is viewed from front, back, left, or right, one sees all four colors.

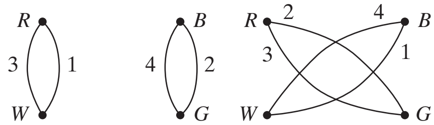
2. Draw a graph G, where the vertices represent the four colors and an edge labeled i connects two vertices if the opposing faces of cube i have those colors. Find two graphs where

- Each vertex has degree 2.
- Each cube represents an edge exactly once in each graph.
- The graphs have no edges in common.

One graph represents the front/back stacking, and the other represents the left/right stacking.

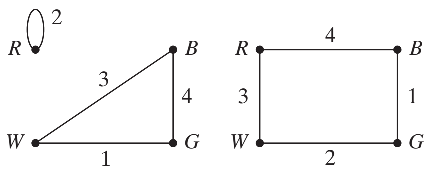
Section 8.8

1.



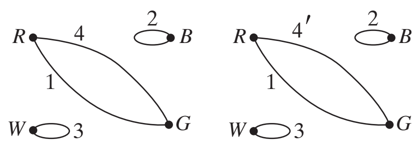
G_1 G_2

4.

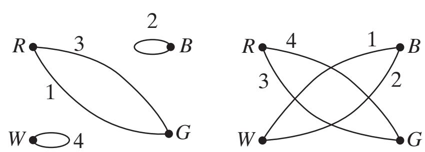


G_1 G_2

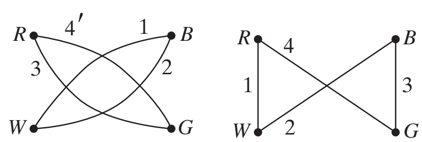
7. (a)



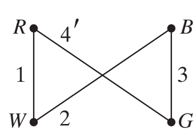
G_1 G_2



G_3 G_4



G_5 G_6

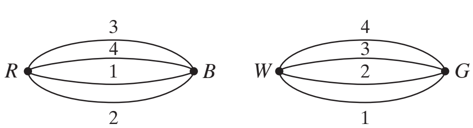


G_7

(b) Solutions are G_1, G_5; G_1, G_7; G_2, G_4; G_2, G_6; G_3, G_6; and G_3, G_7.

13. One edge can be chosen in $C(2+4-1, 2) = 10$ ways. The three edges labeled 1 can be chosen in $C(3+10-1, 3) = 220$ ways. Thus the total number of graphs is 220^4.

15.



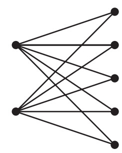
19. According to Exercise 14, not counting loops, every vertex must have degree at least 4. In Figure 8.8.5, not counting loops, vertex W has degree 3 and, therefore, Figure 8.8.5 does not have a solution to the modified version of Instant Insanity. Figure 8.8.3 gives a solution to regular Instant Insanity for Figure 8.8.5.

Chapter 8 Self-Test

1. $V = \{v_1, v_2, v_3, v_4\}$. $E = \{e_1, e_2, e_3\}$. e_1 and e_2 are parallel edges. There are no loops. v_1 is an isolated vertex. G is not a simple graph. e_3 is incident on v_2 and v_4. v_2 is incident on e_1, e_2, and e_3.

2. There are vertices (a and e) of odd degree.

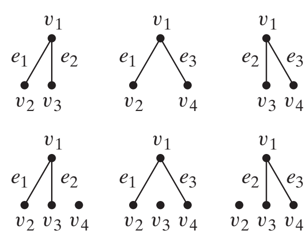
3.



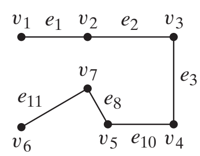
4. If we let V_1 denote the set of vertices containing an even number of 1's and V_2 the set of vertices containing an odd number of 1's, each edge is incident on one vertex in V_1 and one vertex in V_2. Therefore, the n-cube is bipartite.

5. It is a cycle.

6.



7.



8. No. There are vertices of odd degree.

9. $(v_1, v_2, v_3, v_4, v_5, v_7, v_6, v_1)$

10. (000, 001, 011, 010, 110, 111, 101, 100, 000)

11. A Hamiltonian cycle would have seven edges. Suppose that the graph has a Hamiltonian cycle. We would have to eliminate three edges at vertex b and one edge at vertex f. This leaves $10 - 4 = 6$ edges, not enough for a Hamiltonian cycle. Therefore, the graph does not have a Hamiltonian cycle.

12. In a minimum-weight Hamiltonian cycle, every vertex must have degree 2. Therefore, edges (a, b), (a, j), (j, i), (i, h), (g, f), (f, e), and (e, d) must be included. We cannot include edge (b, h) or we will complete a cycle. This implies that we must include edges (h, g) and (b, c). Since vertex g now has degree 2, we cannot include edge (c, g) or (g, d). Thus we must include (c, d). This is a Hamiltonian cycle and the argument shows that it is unique. Therefore, it is minimal.

13. 9

14. 11

15. (a, e, f, i, g, z)

16. 12

17.

$$
\begin{array}{c}
\\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \end{array} \\
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 & 0
\end{pmatrix}
\end{array}
$$

18.

$$
\begin{array}{c}
\\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7
\end{array}
\begin{array}{c}
\begin{array}{ccccccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} & e_{11} \end{array} \\
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1
\end{pmatrix}
\end{array}
$$

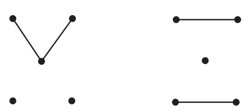
19. The number of paths of length 3 from v_2 to v_3

20. No. Each edge is incident on at least one vertex.

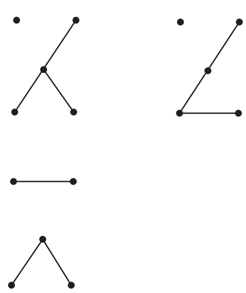
21. The graphs are isomorphic. The orderings v_1, v_2, v_3, v_4, v_5 and w_3, w_1, w_4, w_2, w_5 produce equal adjacency matrices.

22. The graphs are isomorphic. The orderings $v_1, v_2, v_3, v_4, v_5, v_6$ and $w_3, w_6, w_2, w_5, w_1, w_4$ produce equal adjacency matrices.

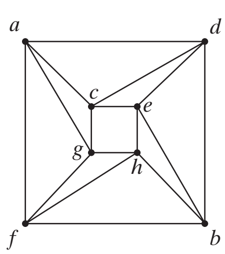
23.



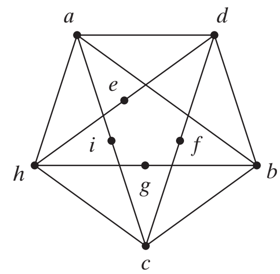
24.



25. The graph is planar:

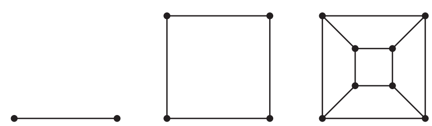


26. The graph is not planar; the following subgraph is homeomorphic to K_5:



27. A simple, planar, connected graph with e edges and v vertices satisfies $e \leq 3v - 6$ (see Exercise 13, Section 8.7). If $e = 31$ and $v = 12$, the inequality is not satisfied, so such a graph cannot be planar.

28. For $n = 1, 2, 3$, it is possible to draw the n-cube in the plane without having any of its edges cross:



$n = 1$ $\quad$ $n = 2$ $\quad$ $n = 3$

We argue by contradiction to show that the 4-cube is not planar. Suppose that the 4-cube is planar. Since every cycle has at least four edges, each face is bounded by at least four edges. Thus the number of edges that bound faces is at least $4f$. In a planar graph, each edge belongs to at most two bounding cycles. Therefore, $2e \geq 4f$. Using Euler's formula for graphs, we find that

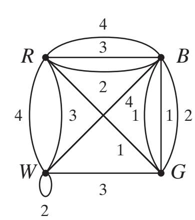
$$2e \geq 4(e - v + 2).$$

For the 4-cube, we have $e = 32$ and $v = 16$, so Euler's formula becomes

$$64 = 2 \cdot 32 \geq 4(32 - 16 + 2) = 72,$$

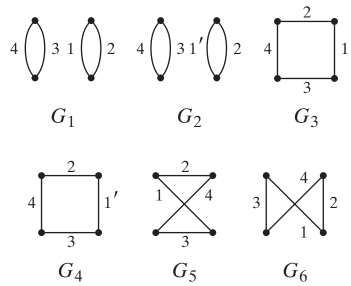
which is a contradiction. Therefore, the 4-cube is not planar. The n-cube, for $n > 4$, is not planar since it contains the 4-cube.

29.



30. See the hints for Exercises 31 and 32.

31.



We denote the two edges incident on B and G labeled 1 in the graph of Exercise 29 as 1 and 1′ here.

32. The puzzle of Exercise 29 has four solutions. Using the notation of Exercise 31, the solutions are G_1, G_5; G_2, G_5; G_3, G_6; and G_4, G_6.

Section 9.1 Review

1. A free tree T is a simple graph satisfying the following: If v and w are vertices in T, there is a unique simple path from v to w.
2. A rooted tree is a tree in which a particular vertex is designated the root.
3. The level of a vertex v is the length of the simple path from the root to v.
4. The height of a rooted tree is the maximum level number that occurs.
5. See Figure 9.1.9.
6. In the rooted tree structure, each vertex represents a file or a folder. Directly under a folder f are the folders and files contained in f.
7. A Huffman code can be defined by a rooted tree. The code for a particular character is obtained by following the simple path from the root to that character. Each edge is labeled with 0 or 1, and the sequence of bits encountered on the simple path is the code for that character.
8. Suppose that there are n frequencies. If $n = 2$, build the tree shown in Figure 9.1.11 and stop. Otherwise, let f_i and f_j denote the smallest frequencies, and replace them in the list by $f_i + f_j$. Recursively construct an optimal Huffman coding tree using the modified list. In the tree that results, add two edges to a vertex labeled $f_i + f_j$, and label the added vertices f_i and f_j.

Section 9.1

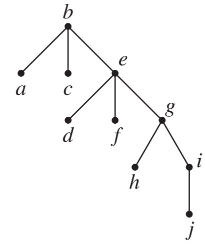
1. The graph is a tree. For any vertices v and w, there is a unique simple path from v to w.

4. The graph is a tree. For any vertices v and w, there is a unique simple path from v to w.

7. $n = 1$

8. a-1; b-1; c-1; d-1; e-2; f-3; g-3; h-4; i-2; j-3; k-0

11. Height = 4



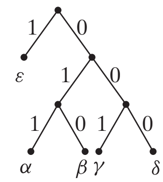
14. PEN

17. SALAD

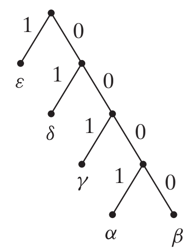
18. 0111100010

21. 0110000100100001111

24.



27. Another tree is shown in the hint for Exercise 24.



32. Let T be a tree. Root T at some arbitrary vertex. Let V be the set of vertices on even levels and let W be the set of vertices on odd levels. Since each edge is incident on a vertex in V and a vertex in W, T is a bipartite graph.

35. e, g

38. The radius is the eccentricity of a center. It is not necessarily true that $2r = d$ (see Figure 9.1.5).

Section 9.2 Review

1. Let $(v_0, \ldots, v_{n-1}, v_n)$ be a path from the root v_0 to v_n. We call v_{n-1} the parent of v_n.
2. Let $(v_0, \ldots, v_n)$ be a path from the root v_0 to v_n. We call $(v_i, \ldots, v_n)$ descendants of v_{i-1}.
3. v and w are siblings if they have the same parent.
4. A terminal vertex is one with no children.
5. If v is not a terminal vertex, it is an internal vertex.
6. An acyclic graph is a graph with no cycles.
7. See Theorem 9.2.3.

Section 9.2

1. Kronos

4. Apollo, Athena, Hermes, Heracles

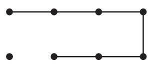
7. b; d

10. e, f, g, j; j

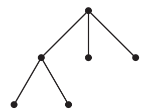
13. a, b, c, d, e

17. They are siblings.

22.



25.

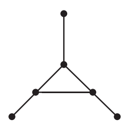


27. A single vertex is a "cycle" of length 0.

30. Each component of a forest is connected and acyclic and, therefore, a tree.

33. Suppose that G is connected. Add parallel edges until the resulting graph G^* has $n-1$ edges. Since G^* is connected and has $n-1$ edges, by Theorem 9.2.3, G^* is acyclic. But adding an edge in parallel introduces a cycle. Contradiction.

36.

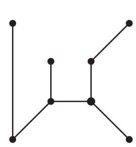


Section 9.3 Review

1. A tree T is a spanning tree of a graph G if T is a subgraph of G that contains all of the vertices of G.
2. A graph G has a spanning tree if and only if G is connected.
3. Select an ordering of the vertices. Select the first vertex and label it the root. Let T consist of this single vertex and no edges. Add to the tree all edges incident on this single vertex that do not produce a cycle when added to the tree. Also add the vertices incident on these edges. Repeat this procedure with the vertices on level 1, then those on level 2, and so on.
4. Select an ordering of the vertices. Select the first vertex and label it the root. Add an edge incident on this vertex to the tree, and add the additional vertex v incident on this edge. Next add an edge incident on v that does not produce a cycle when added to the tree, and add the additional vertex incident on this edge. Repeat this process. If, at any point, we cannot add an edge incident on a vertex w, we backtrack to the parent p of w and try to add an edge incident on p. When we finally backtrack to the root and cannot add more edges, depth-first search concludes.
5. Depth-first search

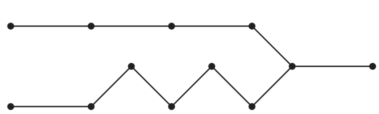
Section 9.3

1.



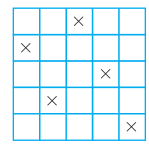
4. The path (h, f, e, g, b, d, c, a)

7.



10. The two-queens problem clearly has no solution. For the three-queens problem, by symmetry, the only possible first column positions are upper left and second from top. If the first move is first column, upper left, the second move must be to the bottom of the second column. Now no move is possible for the third column. If the first move is first column, second from top, there is no move possible in column two. Therefore, there is no solution to the three-queens problem.

13.



17. False. Consider K_4.

20. First, show that the graph T constructed is a tree. Now use induction on the level of T to show that T contains all the vertices of G.

23. Suppose that x is incident on vertices a and b. Removing x from T produces a disconnected graph with two components, U and V. Vertices a and b belong to different components—say, $a \in U$ and $b \in V$. There is a path P from a to b in T'. As we move along P, at some point we encounter an edge $y = (v, w)$ with $v \in U$, $w \in V$. Since adding y to $T - \{x\}$ produces a connected graph, $(T - \{x\}) \cup \{y\}$ is a spanning tree. Clearly, $(T' - \{y\}) \cup \{x\}$ is a spanning tree.

26. Suppose that T has n vertices. If an edge is added to T, the resulting graph T' is connected. If T' were acyclic, T' would be a tree with n edges and n vertices. Thus T' contains a cycle. If T' contains two or more cycles, we would be able to produce a connected graph T'' by deleting two or more edges from T'. But now T'' would be a tree with n vertices and fewer than $n-1$ edges—an impossibility.

27.

$$\begin{array}{c} \\ (abca) \\ (acda) \\ (acdb) \\ (bcdeb) \end{array} \begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_6 & e_5 & e_3 & e_4 & e_7 & e_8 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

30. Input: A graph $G = (V, E)$ with n vertices

Output: true if G is connected
false if G is not connected

```
is_connected(V, E) {
   T = bfs(V, E)
   // T = (V', E') is the spanning tree returned by bfs
   if (|V'| == n)
      return true
   else
      return false
}
```

33.

```
bfs_track_parent(V, E, parent) {
   S = (v1)
   // set v1's parent to 0 to indicate that v1 has no parent
   parent(v1) = 0
   V' = {v1}
```

```
    E' = ∅
    while (true) {
       for each x ∈ S, in order,
          for each y ∈ V − V', in order
             if ((x, y) is an edge) {
                add edge (x, y) to E' and y to V'
                parent(y) = x
             }
       if (no edges were added)
          return T
       S = children of S ordered consistently with the original
             vertex ordering
    }
}
```

34.

```
print_parents(V, parent) {
   for each v ∈ V
      println(v, parent(v))
}
```

37. An algorithm can be obtained by modifying the four-queens algorithm. The array *row* is replaced by the array p, which is the permutation. A conflict for $p(k)$ now means that for some $i < k$, $p(i) = p(k)$, that is, the value $p(k)$ has already been assigned. To obtain *all* of the permutations, when we find a permutation, we print it and continue (whereas in the four-queens algorithm, being content with one solution, we terminated the algorithm).

```
perm(n) {
   k = 1
   p(1) = 0
   while (k > 0) {
      p(k) = p(k) + 1
      while (p(k) ≤ n ∧ p(k) conflicts)
         p(k) = p(k) + 1
      if (p(k) ≤ n)
         if (k == n)
            println(p)
         else {
            k = k + 1
            p(k) = 0
         }
      else
         k = k − 1
   }
}
```

40. The idea of the backtracking algorithm is to scan the grid (we chose to scan top to bottom, left to right), skipping positions where numbers were preassigned, and, at the next available position, we try 1, then 2, then 3, and so on, until we find a legal value (i.e., a value that does not conflict within its 3×3 sub-square, within its row, or within its column). If such a value is found, we continue with the next available position. If no such value can be found, we backtrack to the last position where we assigned a value; if that value was i, we try $i+1$, $i+2$ and so on.

In the following algorithm, the value $s(i, j)$ is the value in row i, column j, or 0 if no value is stored there. We assume that initially all values in s are set to 0, except for those values that are specified in the puzzle. Finally, *show_values* prints the array s.

```
sudoku(s) {
   i = 0
   j = 1
   // advance advances i and j to the next position in which
   // a value is not specified. It proceeds down a column first.
   advance(i, j)
   while (i ≥ 1 ∧ j ≥ 1) {
      // search for a legal value
      s(i, j) = s(i, j) + 1
      // not_valid(i, j) returns true if the value s(i, j)
      // conflicts with the previously chosen and specified
      // values, and false otherwise.
      while (s(i, j) < 10 ∧ not_valid(i, j))
         s(i, j) = s(i, j) + 1
      // if no value found, backtrack
      if (s(i, j) == 10) {
         s(i, j) = 0
         // retreat moves i and j to the previous position in
         // which a value is not specified. It proceeds up a
         // column first.
         retreat(i, j)
      }
      else
         advance(i, j) // sets j to 10 if advanced off board
      if (j == 10) {
         // Solution!
         show_values()
         return
      }
   }
}
```

Section 9.4 Review

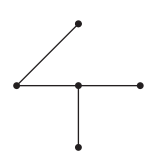
1. A minimal spanning tree is a spanning tree with minimum weight.

2. Prim's Algorithm builds a minimal spanning tree by iteratively adding edges. The algorithm begins with a single vertex. Then at each iteration, it adds to the current tree a minimum-weight edge that does not complete a cycle.

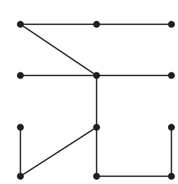
3. A greedy algorithm optimizes the choice at each iteration.

Section 9.4

1.



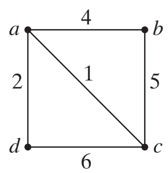
4.



10. If v is the first vertex examined by Prim's Algorithm, the edge will be in the minimal spanning tree constructed by the algorithm.

13. Suppose that G has two minimal spanning trees T_1 and T_2. Then, there exists an edge x in T_1 that is not in T_2. By Exercise 23, Section 9.3, there exists an edge y in T_2 that is not in T_1 such that $T_3 = (T_1 - \{x\}) \cup \{y\}$ and $T_4 = (T_2 - \{y\}) \cup \{x\}$ are spanning trees. Since x and y have different weights, either T_3 or T_4 has weight less than T_1. This is a contradiction.

14. False



16. False. Consider K_5 with the weight of every edge equal to 1.

20. Input: The edges E of an n-vertex, connected, weighted graph. If e is an edge, $w(e)$ is equal to the weight of e; if e is not an edge, $w(e)$ is equal to ∞ (a value greater than any actual weight).

Output: A minimal spanning tree.

```
kruskal(E, w, n) {
   V′ = ∅
   E′ = ∅
   T′ = (V′, E′)
   while (|E′| < n − 1) {
      among all edges that if added to T′ would not
        complete a cycle, choose e = (v_i, v_j) of
        minimum weight
      E′ = E′ ∪ {e}
      V′ = V′ ∪ {v_i, v_j}
      T′ = (V′, E′)
   }
   return T′
}
```

23. Terminate Kruskal's Algorithm after k iterations. This groups the data into $n - k$ classes.

27. We show that $a_1 = 7$ and $a_2 = 3$ provide a solution. We use induction on n to show that the greedy solution gives an optimal solution for $n \geq 1$. The cases $n = 1, 2, \ldots, 8$ may be verified directly.

We first show that if $n \geq 9$, there is an optimal solution containing at least one 7. Let S' be an optimal solution. Suppose that S' contains no 7's. Since S' contains at most two 1's (since S' is optimal), S' contains at least three 3's. We replace three 3's by one 7 and two 1's to obtain a solution S. Since $|S| = |S'|$, S is optimal.

If we remove a 7 from S, we obtain a solution S^* to the $(n-7)$-problem. If S^* were not optimal, S could not be optimal. Thus S^* is optimal. By the inductive assumption, the greedy solution GS^* to the $(n-7)$-problem is optimal, so $|S^*| = |GS^*|$. Notice that 7 together with GS^* is the greedy solution GS to the n-problem. Since $|GS| = |S|$, GS is optimal.

29. Suppose that the greedy algorithm is optimal for all denominations less than $a_{m-1} + a_m$. We use induction on n to show that the greedy algorithm is optimal for all n. We may assume that $n \geq a_{m-1} + a_m$.

Consider an optimal solution $\mathcal{S}$ for n. First suppose that $\mathcal{S}$ uses at least one a_m coin. The solution, $\mathcal{S}$ with one a_m coin removed, is optimal for $n - a_m$. (If there was a solution for $n - a_m$ using fewer coins, we could add one a_m coin to it to obtain a solution for n using fewer coins than $\mathcal{S}$, which is impossible.) By the inductive assumption, the greedy solution for $n - a_m$ is optimal. If we add one a_m coin to the greedy solution for $n - a_m$, we obtain a solution $\mathcal{G}$ for n that uses the same number of coins as $\mathcal{S}$. Therefore, $\mathcal{G}$ is optimal. But $\mathcal{G}$ is also greedy because the greedy solution begins by removing one a_m coin.

Now suppose that $\mathcal{S}$ does not use an a_m coin. Let i be the largest index such that $\mathcal{S}$ uses an a_i coin. The solution, $\mathcal{S}$ with one a_i coin removed, is optimal for $n - a_i$. By the inductive assumption, the greedy solution for $n - a_i$ is optimal. Now

$$n \geq a_{m-1} + a_m \geq a_i + a_m,$$

so $n - a_i \geq a_m$. Therefore, the greedy solution uses at least one a_m coin. Thus there is an optimal solution for $n - a_i$ that uses an a_m coin. If we add one a_i coin to this optimal solution, we obtain an optimal solution for n that uses an a_m coin. The argument in the preceding paragraph can now be repeated to show that the greedy solution is optimal.

Section 9.5 Review

1. A binary tree is a rooted tree in which each vertex has either no children, one child, or two children.

2. A left child of vertex v is a child designated as "left."

3. A right child of vertex v is a child designated as "right."

4. A full binary tree is a binary tree in which each vertex has either two children or zero children.

5. $i + 1$

6. $2i + 1$

7. If a binary tree of height h has t terminal vertices, then $\lg t \leq h$.

8. A binary search tree is a binary tree T in which data are associated with the vertices. The data are arranged so that, for each vertex v in T, each data item in the left subtree of v is less than the data item in v, and each data item in the right subtree of v is greater than the data item in v.

9. See Figures 9.5.4 and 9.5.5.

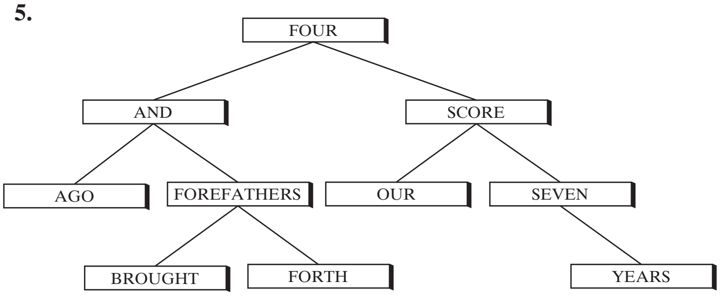
10. Insert the first data item in a vertex and label it the root. Insert the next data items in the tree according to the following steps. Begin at the root. If the data item to be added is less than the data item at the current vertex, move to the left child and repeat; otherwise, move to the right child and repeat. If there is no child, create one, put an edge incident on it and the last vertex visited, and store the data item in the added vertex.

Section 9.5

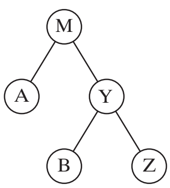
1. Example 9.5.5 showed that $n - 1$ games are played. Since there are two choices for the winner of each game, the tournament can unfold in 2^{n-1} ways.

4. No. Based on past performance, it is likely that certain teams will defeat other teams. Someone knowledgeable about basketball will take this into account. For example, through 2007 a number 16 seed has never defeated a number 1 seed.

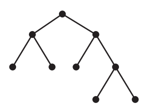
5.



8. False. Consider



9.



12. $mi + 1$, $(m - 1)i + 1$

15. $t - 1$

18. Balanced

21. Balanced

22. A tree of height 0 has one vertex, so $N_0 = 1$. In a balanced binary tree of height 1, the root must have at least one child. If the root has exactly one child, the number of vertices will be minimized. Therefore, $N_1 = 2$. In a balanced binary tree of height 2, there must be a path from the root to a terminal vertex of length 2. This accounts for three vertices. But for the tree to be balanced, the root must have two children. Therefore, $N_2 = 4$.

25. Suppose that there are n vertices in a balanced binary tree of height h. Then

$$n \geq N_h = f_{h+3} - 1 > \left(\frac{3}{2}\right)^{h+2} - 1,$$

for $h \geq 3$. The equality comes from Exercise 24 and the last inequality comes from Exercise 27, Section 4.4. Therefore,

$$n + 1 > \left(\frac{3}{2}\right)^{h+2}.$$

Taking the logarithm to the base 3/2 of each side, we obtain

$$\log_{3/2}(n+1) > h + 2.$$

Therefore,

$$h < [\log_{3/2}(n+1)] - 2 = O(\lg n).$$

Section 9.6 Review

1. Preorder traversal processes the vertices of a binary tree by beginning at the root and recursively processing the current vertex, the vertex's left subtree, and then the vertex's right subtree.

2. Input: PT, the root of a binary tree
   Output: Dependent on how "process" is interpreted

```
preorder(PT) {
   if (PT == null)
      return
   process PT
   l = left child of PT
   preorder(l)
   r = right child of PT
   preorder(r)
}
```

3. Inorder traversal processes the vertices of a binary tree by beginning at the root and recursively processing the vertex's left subtree, the current vertex, and then the vertex's right subtree.

4. Input: PT, the root of a binary tree
   Output: Dependent on how "process" is interpreted

```
inorder(PT) {
   if (PT == null)
      return
   l = left child of PT
   inorder(l)
   process PT
   r = right child of PT
   inorder(r)
}
```

5. Postorder traversal processes the vertices of a binary tree by beginning at the root and recursively processing the vertex's left subtree, the vertex's right subtree, and then the current vertex.

6. Input: PT, the root of a binary tree
   Output: Dependent on how "process" is interpreted

```
postorder(PT) {
   if (PT == null)
      return
   l = left child of PT
   postorder(l)
   r = right child of PT
   postorder(r)
   process PT
}
```

7. In the prefix form of an expression, an operator precedes its operands.

8. Polish notation

9. In the infix form of an expression, an operator is between its operands.

10. In the postfix form of an expression, an operator follows its operands.

11. Reverse Polish notation

12. No parentheses are needed.

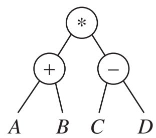
13. In a tree representation of an expression, the internal vertices represent operators, and the operators operate on the subtrees.

Section 9.6

1. *preorder* *inorder* *postorder*
ABDCE *BDAEC* *DBECA*

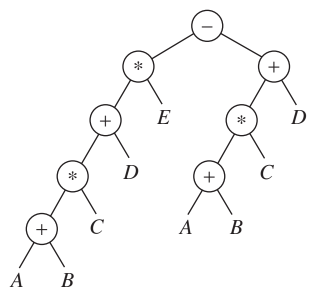
4. *preorder* *inorder* *postorder*
ABCDE *EDCBA* *EDCBA*

6.



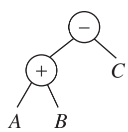
prefix: $* + AB - CD$
postfix: $AB + CD - *$

9.



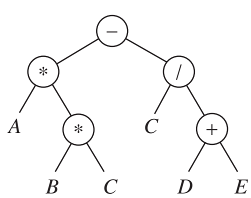
prefix: $- * + * + ABCDE + * + ABCD$
postfix: $AB + C * D + E * AB + C * D + -$

11.



prefix: $- + ABC$
usual infix: $A + B - C$
parened infix: $((A + B) - C)$

14.

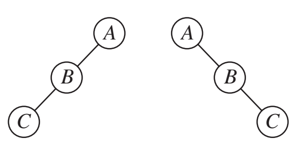


prefix: $- * A * BC/C + DE$
usual infix: $A * B * C - C/(D + E)$
parened infix: $((A * (B * C)) - (C/(D + E)))$

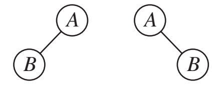
16. -4

19. 0

22.



25.



28. Input: PT, the root of a binary tree
Output: PT, the root of the modified binary tree

```
swap_children (PT) {
   if (PT == null)
      return
   swap the left and right children of PT
   l = left child of PT
   swap_children(l)
   r = right child of PT
   swap_children(r)
}
```

31. If T is a binary tree, we let *post*(T) denote the order in which the vertices of T are visited under postorder traversal. We let *revpost*(T) denote the reverse of *post*(T). We prove by induction on the number of nodes in a tree T that the order in which *funnyorder* visits the nodes of T is *revpost*(T).

The assertion is evident if T has no nodes. Thus the basis step is proved.

Now assume that the order in which *funnyorder* visits the nodes of a tree T' having fewer than n nodes is *revpost*(T'). Let T be an n-node tree. We must prove that the order in which *funnyorder* visits the nodes of T is *revpost*(T).

Let T_1 be the left subtree of T, let T_2 be the right subtree of T, and let r be the root of T. By the inductive assumption, the order in which *funnyorder* visits the nodes of T_1 is *revpost*(T_1), and the order in which *funnyorder* visits the nodes of T_2 is *revpost*(T_2). The pseudocode shows that the order in which *funnyorder* visits the nodes of T is

$$r,\ revpost(T_2),\ revpost(T_1).$$

The reverse of this list is

$$post(T_1),\ post(T_2),\ r,$$

which is the order in which postorder visits the nodes of T. The inductive step is complete.

32. Define an *initial segment* of a string to be the first $i \geq 1$ characters for some i. Define $r(x) = 1$, for $x = A, B, \ldots, Z$; and $r(x) = -1$, for $x = +, -, *, /$. If $x_1 \cdots x_n$ is a string over $\{A, \ldots, Z, +, -, *, /\}$, define

$$r(x_1 \cdots x_n) = r(x_1) + \cdots + r(x_n).$$

Then a string s is a postfix string if and only if $r(s) = 1$ and $r(s') \geq 1$, for all initial segments s' of s.

35. Let G be the graph with vertex set $\{1, 2, \ldots, n\}$ and edge set

$$\{(1, i) \mid i = 2, \ldots n\}.$$

The $\{1\}$ is a vertex cover of G of size 1.

38. Input: PT, the root of a nonempty tree

Output: Each vertex of the tree has a field *in_cover* that is set to true if that vertex is in the vertex cover or to false if that vertex is not in the vertex cover.

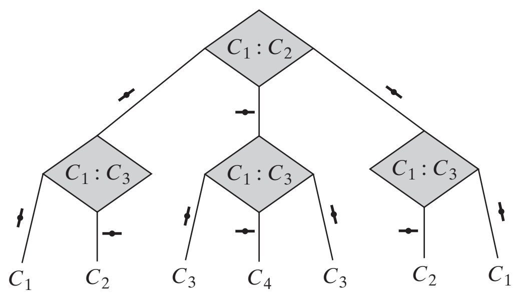
```
tree_cover(PT) {
   flag = false
   ptr = first child of PT
   while (ptr ! = null) {
      tree_cover(ptr)
      if (in_cover of ptr == false)
         flag = true
      ptr = next sibling of ptr
   }
   in_cover of PT = flag
}
```

Section 9.7 Review

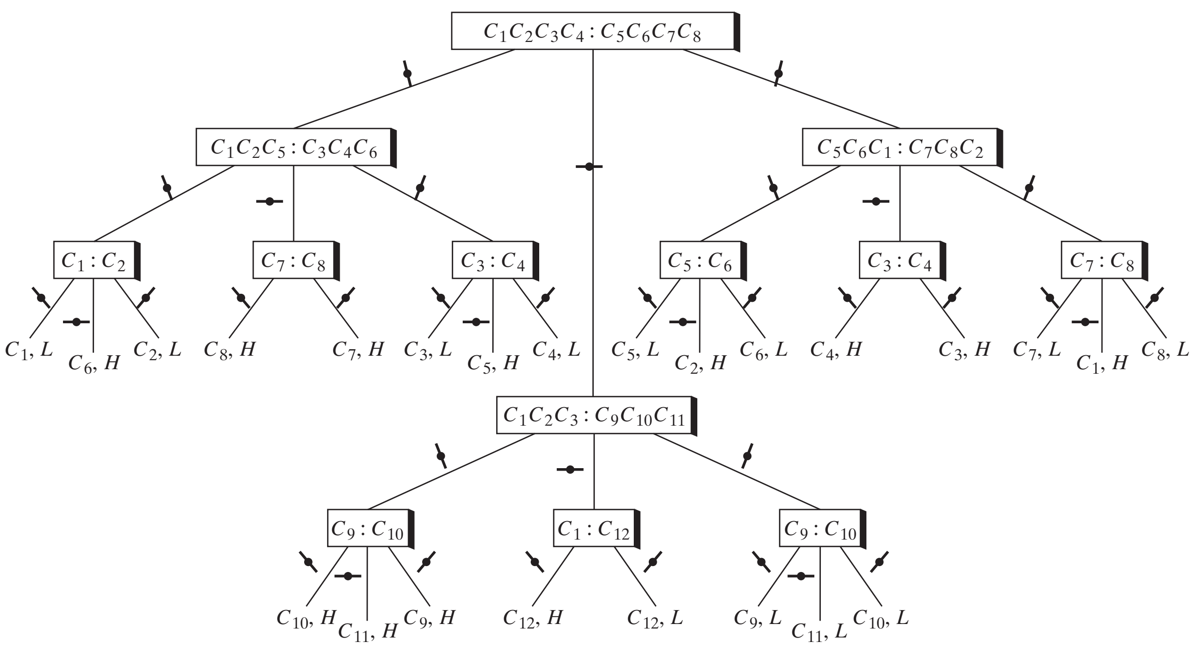
1. A decision tree is a binary tree in which the internal vertices contain questions with two possible answers, the edges are labeled with answers to the questions, and the terminal vertices represent decisions. If we begin at the root, answer each question, and follow the appropriate edges, we will eventually arrive at a terminal vertex that represents a decision.
2. The worst-case time of an algorithm is proportional to the height of the decision tree that represents the algorithm.
3. A decision tree that represents a sorting algorithm has $n!$ terminal vertices corresponding to the $n!$ possible permutations of input of size n. If h is the height of the tree, then h comparisons are required in the worst case. Since $\lg n! \le h$ and $\lg n! = \Theta(n \lg n)$, worst-case sorting requires at least $\Omega(n \lg n)$ comparisons.

Section 9.7

1.



4. In this graph only, if the left pan is heavier, go right.



7. There are 28 possible outcomes to the fourteen-coins puzzle. A tree of height 3 has at most 27 terminal vertices; thus at least four weighings are required in the worst case. In fact, there is an algorithm that uses four weighings in the worst case: We begin by weighing four coins against four coins. If the coins do not balance, we proceed as in the solution given for Exercise 4 (for the 12-coins puzzle). In this case, at most three weighings are required. If the coins do balance, we disregard these coins; our problem then is to find the bad coin from among the remaining six coins. The six-coins puzzle can be solved in at most three weighings in the worst case, which, together with the initial weighing, requires four weighings in the worst case.

9. Let $f(n)$ denote the number of weighings needed to solve the n-coin problem in the worst case. Let T be the decision tree that represents this algorithm for input of size n and let h denote the height of T. Then the algorithm requires h weighings in the worst case so $h = f(n)$. Since there are $n-1$ possible outcomes, T has at least $n-1$ terminal vertices. By the analog of Theorem 9.5.6 for "trinary" trees, $\log_3(n-1) \le h = f(n)$.

12. The decision tree analysis shows that at least $\lceil \lg 5! \rceil = 7$ comparisons are required to sort five items in the worst case. The following algorithm sorts five items using at most seven comparisons in the worst case.

Given the sequence $a_1, \ldots, a_5$, we first sort a_1, a_2 (one comparison) and then a_3, a_4 (one comparison). (We assume now that $a_1 < a_2$ and $a_3 < a_4$.) We then compare a_2 and a_4. Let us assume that $a_2 < a_4$. (The case $a_2 > a_4$ is symmetric and for this reason that part of the algorithm is omitted.) At this point we know that

$$a_1 < a_2 < a_4 \qquad \text{and} \qquad a_3 < a_4.$$

Next we determine where a_5 belongs among a_1, a_2, and a_4 by first comparing a_5 with a_2. If $a_5 < a_2$, we next compare a_5 with a_1; but if $a_5 > a_2$, we next compare a_5 with a_4. In either case, two additional comparisons are required. At this point, a_1, a_2, a_4, a_5 is sorted. Finally, we insert a_3 in its proper place. If we first compare a_3 with the second-smallest item among a_1, a_2, a_4, a_5, only one additional comparison will be required, for a total of seven comparisons. To justify this last statement, we note that the following arrangements are possible after we insert a_5 in its correct position:

$$\begin{gathered} a_5 < a_1 < a_2 < a_4 \\ a_1 < a_5 < a_2 < a_4 \\ a_1 < a_2 < a_5 < a_4 \\ a_1 < a_2 < a_4 < a_5. \end{gathered}$$

If a_3 is less than the second item, only one additional comparison is needed (with the first item) to locate the correct position for a_3. If a_3 is greater than the second item, at most one additional comparison is needed to locate the correct position for a_3. In the first three cases, we need only compare a_3 with either a_2 or a_5 to find the correct position for a_3 since we already know that $a_3 < a_4$. In the fourth case, if a_3 is greater than a_2, we know that it goes between a_2 and a_4.

14. We can consider the numbers as contestants and the internal vertices as winners where the larger value wins.

17. Suppose we have an algorithm that finds the largest value among $x_1, \ldots, x_n$. Let $x_1, \ldots, x_n$ be the vertices of a graph. An edge exists between x_i and x_j if the algorithm compares x_i and x_j. The graph must be connected. The least number of edges necessary to connect n vertices is $n-1$.

20. By Exercise 16, Tournament Sort requires $2^k - 1$ comparisons to find the largest element. By Exercise 18, Tournament Sort requires k comparisons to find the second-largest element. Similarly, Tournament Sort requires at most k comparisons to find the third-largest, at most k comparisons to find the fourth-largest, and so on. Thus the total number of comparisons is at most

$$\begin{aligned} [2^k - 1] + (2^k - 1)k &\le 2^k + k2^k \\ &\le k2^k + k2^k \\ &= 2 \cdot 2^k k = 2n \lg n. \end{aligned}$$

Section 9.8 Review

1. Free trees T_1 and T_2 are isomorphic if there is a one-to-one, onto function f from the vertex set of T_1 to the vertex set of T_2 satisfying the following: Vertices v_i and v_j are adjacent in T_1 if and only if the vertices $f(v_i)$ and $f(v_j)$ are adjacent in T_2.

2. Let T_1 be a rooted tree with root r_1 and let T_2 be a rooted tree with root r_2. Then T_1 and T_2 are isomorphic if there is a one-to-one, onto function f from the vertex set of T_1 to the vertex set of T_2 satisfying the following:

(a) v_i and v_j are adjacent in T_1 if and only if $f(v_i)$ and $f(v_j)$ are adjacent in T_2.

(b) $f(r_1) = f(r_2)$.

3. Let T_1 be a binary tree with root r_1 and let T_2 be a binary tree with root r_2. Then T_1 and T_2 are isomorphic if there is a one-to-one, onto function f from the vertex set of T_1 to the vertex set of T_2 satisfying the following:

(a) v_i and v_j are adjacent in T_1 if and only if $f(v_i)$ and $f(v_j)$ are adjacent in T_2.

(b) $f(r_1) = f(r_2)$.

(c) v is a left child of w in T_1 if and only if $f(v)$ is a left child of $f(w)$ in T_2.

(d) v is a right child of w in T_1 if and only if $f(v)$ is a right child of $f(w)$ in T_2.

4. $C(2n, n)/(n+1)$

5. Given binary trees T_1 and T_2, we first check whether either is empty (in which case it is immediate whether they are isomorphic). If both are nonempty, we first check whether the left subtrees are isomorphic and then whether the right subtrees are isomorphic. T_1 and T_2 are isomorphic if and only if their left and right subtrees are isomorphic.

Section 9.8

1. Isomorphic. $f(v_1) = w_1$, $f(v_2) = w_5$, $f(v_3) = w_3$, $f(v_4) = w_4$, $f(v_5) = w_2$, $f(v_6) = w_6$.

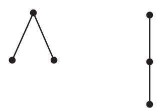
4. Not isomorphic. T_2 has a simple path of length 2 from a vertex of degree 1 to a vertex of degree 1, but T_1 does not.

7. Isomorphic as rooted trees. $f(v_1)=w_1$, $f(v_2)=w_4$, $f(v_3)=w_3$, $f(v_4)=w_2$, $f(v_5)=w_6$, $f(v_6)=w_5$, $f(v_7)=w_7$, $f(v_8)=w_8$. Also isomorphic as free trees.

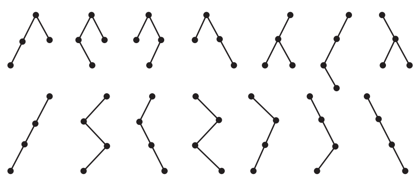
10. Not isomorphic as binary trees. The root of T_1 has a left child but the root of T_2 does not. Isomorphic as rooted trees and as free trees.

13.

16.



19.



22. Let b_n denote the number of nonisomorphic, n-vertex full binary trees. Since every full binary tree has an odd number of vertices, $b_n = 0$ if n is even. We show that if $n = 2i+1$ is odd,

$$b_n = C_i,$$

where C_i denotes the ith Catalan number.

The last equation follows from the fact that there is a one-to-one, onto function from the set of i-vertex binary trees to the set of $(2i + 1)$-vertex full binary trees. Such a function may be constructed as follows. Given an i-vertex binary tree, at every terminal vertex we add two children. At every vertex with one child, we add an additional child. Since the tree that is obtained has i internal vertices, there are $2i + 1$ vertices total (Theorem 9.5.4). The tree constructed is a full binary tree. Notice that this function is one-to-one. Given a $(2i+1)$-vertex full binary tree T', if we eliminate all the terminal vertices, we obtain an i-vertex binary tree T. The image of T is T'. Therefore, the function is onto.

25. There are four comparisons at lines 1 and 3. By Exercise 24, the call *bin_ tree_ isom*(lc_r_1, lc_r_2) requires $6(k - 1) + 2$ comparisons. The call *bin_tree_isom*(rc_r_1, rc_r_2) requires four comparisons. Thus the total number of comparisons is

$$4 + 6(k - 1) + 2 + 4 = 6k + 4.$$

27. Let T^* denote the tree constructed. Then T^* is a full binary tree. Each vertex in T becomes an internal vertex in T^*. Since we added only terminal vertices, the original $n - 1$ vertices in T are the only internal vertices in T^*. By Theorem 9.5.4, T^* has n terminal vertices. Therefore $T^* \in X_1$. We leave it to the reader to check that this mapping is a bijection. By Theorem 9.8.12, there are C_{n-1} $(n - 1)$-vertex binary trees. Therefore $|X_1| = C_{n-1}$.

29. By Theorem 9.5.4, a tree in X_1 has $n - 1$ internal vertices and $2n - 1$ total vertices. Thus we may choose the vertex v in $2n - 1$ ways and the vertex to mark (left or right) in 2 ways. Therefore $|X_T| = 2(2n - 1)$.

33. Using iteration, we have

$$\begin{aligned}
C_n &= \frac{2(2n-1)}{n+1}C_{n-1}\\
&= \frac{2(2n-1)}{n+1}\,\frac{2(2n-3)}{n}C_{n-2}\\
&= \frac{2^2(2n-1)(2n-3)}{(n+1)n}C_{n-2}\\
&= \frac{2^3(2n-1)(2n-3)(2n-5)}{(n+1)n(n-1)}C_{n-3}\\
&\ \ \vdots\\
&= \frac{2^{n-1}(2n-1)(2n-3)\cdots 3}{(n+1)n(n-1)\cdots 3}C_1\\
&= \frac{1}{n+1}\left[\frac{2^n(2n-1)(2n-3)\cdots 3}{n(n-1)\cdots 3\cdot 2}\right]\\
&= \frac{1}{n+1}\left[\frac{2^n n!(2n-1)(2n-3)\cdots 3}{n!n!}\right]\\
&= \frac{1}{n+1}\left\{\frac{[(2n)(2n-2)\cdots 2][(2n-1)(2n-3)\cdots 3]}{n!n!}\right\}\\
&= \frac{1}{n+1}\,\frac{(2n)!}{n!n!} = \frac{1}{n+1}C(2n,n).
\end{aligned}$$

Section 9.9 Review

1. In a game tree, each vertex shows a particular position in the game. In particular, the root shows the initial configuration of the game. The children of a vertex show all possible responses by a player to the position shown in the vertex.

2. In the minimax procedure, values are first assigned to the terminal vertices in a game tree. Then, working from the bottom up, the value of a circle is set to the minimum of the values of its children, and the value of a box is set to the maximum of the values of its children.

3. A search that terminates n levels below the given vertex.

4. An evaluation function assigns to each possible game position the value of the position to the first player.

5. Alpha-beta pruning deletes (prunes) parts of the game tree and thus omits evaluating parts of it when the minimax procedure is applied. Alpha-beta pruning works as follows. Suppose that a box vertex v is known to have a value of at least x. When a grandchild w of v has a value of at most x, the subtree whose root is the parent of w is deleted. Similarly, suppose that a circle vertex v is known to have a value of at most x. When a grandchild w of v has a value of at least x, the subtree whose root is the parent of w is deleted.

6. An alpha value is a lower bound for a box vertex.

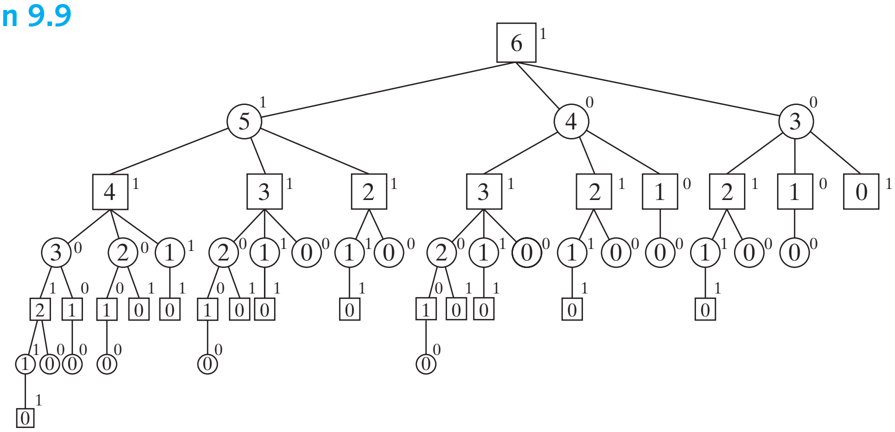
7. An alpha cutoff occurs at a box vertex when a grandchild w of v has a value less than or equal to the alpha value of v.

8. A beta value is an upper bound for a circle vertex.

9. A beta cutoff occurs at a circle vertex when a grandchild w of v has a value greater than or equal to the beta value of v.

Section 9.9

1.



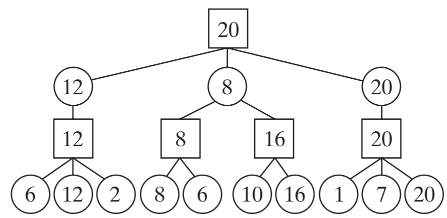
The first player always wins. The winning strategy is to first take one token; then, whatever the second player does, leave one token.

4. The second player always wins. If two piles remain, leave piles with equal numbers of tokens. If one pile remains, take it.

7. Suppose that the first player can win in nim. The first player can always win in nim′ by adopting the following strategy: Play nim′ exactly like nim unless the move would leave an odd number of singleton piles and no other pile. In this case, leave an even number of piles.

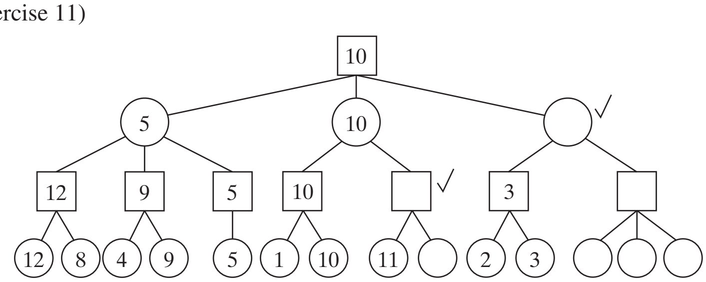
Suppose that the first player can always win in nim′. The first player can always win in nim by adopting the following strategy: Play nim exactly like nim′ unless the move would leave an even number of singleton piles and no other pile. In this case, leave an odd number of piles.

9.



12. The value of the root is 3.

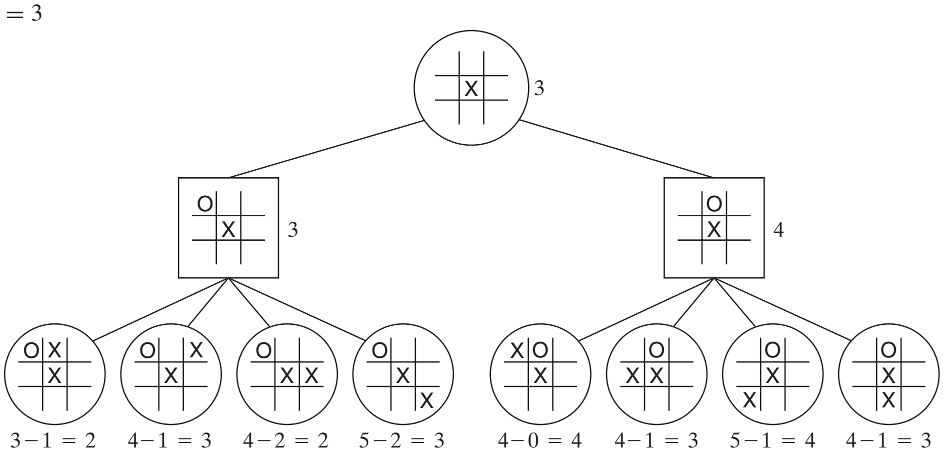
14. (For Exercise 11)



15. $3 - 2 = 1$

18. $4 - 1 = 3$

19.



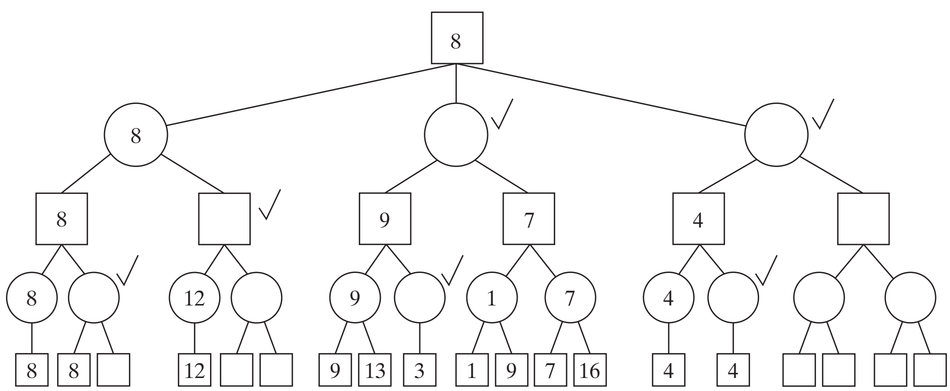
O will move to a corner.

22. Input: The root PT of a game tree; the type PT_type of PT (*box* or *circle*); the level PT_level of PT; the maximum level n to which the search is to be conducted; an evaluation function E; and a number ab_val (which is either the alpha- or beta-value of the parent of PT). (The initial call sets ab_val to ∞ if PT is a box vertex or to $-\infty$ if PT is a circle vertex.)

Output: The game tree with PT evaluated

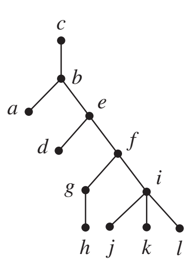
```
alpha_beta_prune(PT, PT_type, PT_level, n, E, ab_val) {
   if (PT_level == n) {
      contents(PT) = E(PT)
      return
   }
   if (PT_type == box) {
      contents(PT) = −∞
      for each child C of PT {
         alpha_beta_prune(C, circle, PT_level + 1, n,
            E, content(PT))
         c_val = contents(C)
         if (c_val ≥ ab_val) {
            contents(PT) = ab_val
            return
         }
         if (c_val > contents(PT))
            contents(PT) = c_val
      }
   }
   else {
      contents(PT) = ∞
      for each child C of PT {
         alpha_beta_prune(C, box, PT_level + 1, n,
            E, content(PT))
         c_val = contents(C)
         if (c_val ≤ ab_val) {
            contents(PT) = ab_val
            return
         }
         if (c_val < contents(PT))
            contents(PT) = c_val
      }
   }
}
```

23. We first obtain the values 6, 6, 7 for the children of the root. Then we order the children of the root with the rightmost child first and use the alpha-beta procedure to obtain



Chapter 9 Self-Test

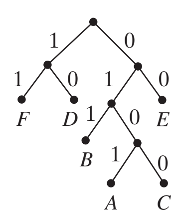
1.



2. a-2, b-1, c-0, d-3, e-2, f-3, g-4, h-5, i-4, j-5, k-5, l-5

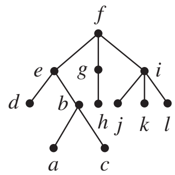
3. 5

4.



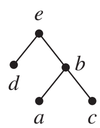
5. (a) b

(b) a, c



(c) d, a, c, h, j, k, l

(d)

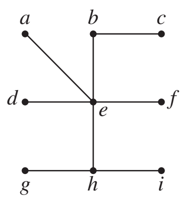


6. True. See Theorem 9.2.3.

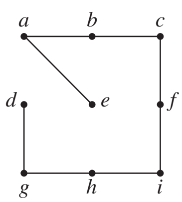
7. True. A tree of height 6 or more must have seven or more vertices.

8. False.

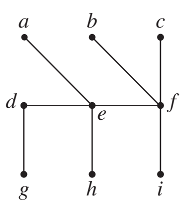
9.



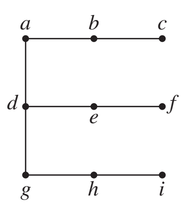
10.



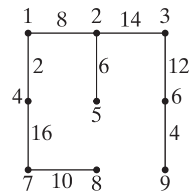
11.



12.



13.

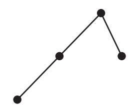


14. (1, 4), (1, 2), (2, 5), (2, 3), (3, 6), (6, 9), (4, 7), (7, 8)

15. (6, 9), (3, 6), (2, 3), (2, 5), (1, 2), (1, 4), (4, 7), (7, 8)

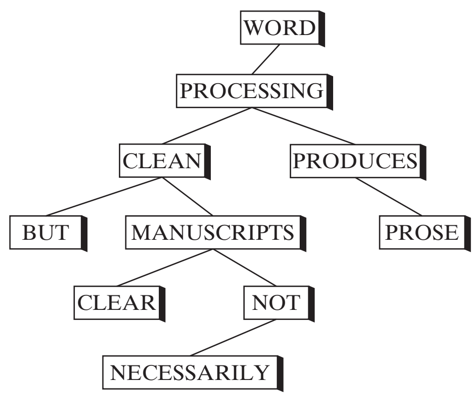
16. Consider a "shortest-path algorithm" in which at each step we select an available edge having minimum weight incident on the most recently added vertex (see the discussion preceding Theorem 9.4.5).

17.



18. 16

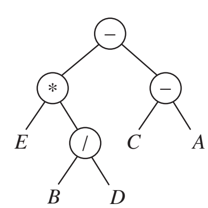
19.



20. We first compare MORE with the word WORD in the root. Since MORE is less than WORD, we go to the left child. Next, we compare MORE with PROCESSING. Since MORE is less than PROCESSING, we go to the left child. Since MORE is greater than CLEAN, we go to the right child. Since MORE is greater than MANUSCRIPTS, we go to the right child. Since MORE is less than NOT, we go to the left child. Since MORE is less than NECESSARILY, we attempt to go to the left child. Since there is no left child, we conclude that MORE is not in the tree.

21. *ABFGCDE* **22.** *BGFAEDC* **23.** *GFBEDCA*

24.

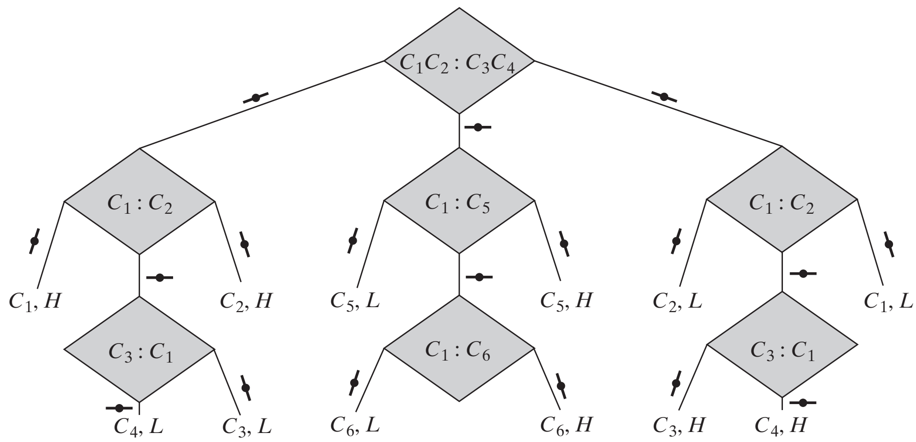


postfix: $EBD/*CA--$

parened infix: $((E*(B/D))-(C-A))$

25. An algorithm that requires at most two weighings can be represented by a decision tree of height at most 2. However, such a tree has at most nine terminal vertices. Since there are 12 possible outcomes, there is no such algorithm. Therefore, at least three weighings are required in the worst case to identify the bad coin and determine whether it is heavy or light.

26.



27. According to Theorem 9.7.3, any sorting algorithm requires at least $Cn \lg n$ comparisons in the worst case. Since Professor Sabic's algorithm uses at most $100n$ comparisons, we must have $Cn \lg n \le 100n$ for all $n \ge 1$. If we cancel n, we obtain $C \lg n \le 100$ for all $n \ge 1$, which is false. Therefore, the professor does not have a sorting algorithm that uses at most $100n$ comparisons in the worst case for all $n \ge 1$.

28. In the worst case, three comparisons are required to sort three items using an optimal sort (see Example 9.7.2).

If $n = 4$, binary insertion sort sorts three items (three comparisons—worst case) and then inserts the fourth item in the sorted three-item list (two comparisons—worst case) for a total of five comparisons in the worst case.

If $n = 5$, binary insertion sort sorts four items (five comparisons—worst case) and then inserts the fifth item in the sorted four-item list (three comparisons—worst case) for a total of eight comparisons in the worst case.

If $n = 6$, binary insertion sort sorts five items (eight comparisons—worst case) and then inserts the sixth item in the sorted five-item list (three comparisons—worst case) for a total of eleven comparisons in the worst case.

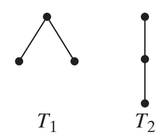
The decision tree analysis shows that any algorithm requires at least five comparisons in the worst case to sort four items. Thus binary insertion sort is optimal if $n = 4$.

The decision tree analysis shows that any algorithm requires at least seven comparisons in the worst case to sort five items. It is possible, in fact, to sort five items using seven comparisons in the worst case. Thus binary insertion sort is not optimal if $n = 5$.

The decision tree analysis shows that any algorithm requires at least ten comparisons in the worst case to sort six items. It is possible, in fact, to sort six items using ten comparisons in the worst case. Thus binary insertion sort is not optimal if $n = 6$.

29. True. If f is an isomorphism of T_1 and T_2 as rooted trees, f is also an isomorphism of T_1 and T_2 as free trees.

30. False.



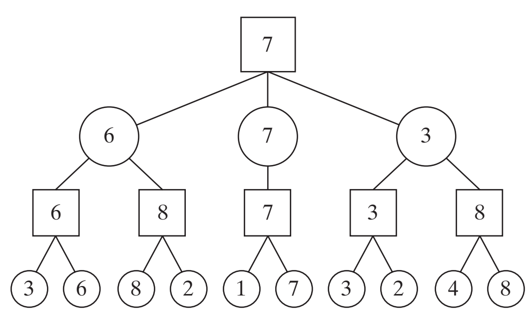
31. Isomorphic. $f(v_1) = w_6$, $f(v_2) = w_2$, $f(v_3) = w_5$, $f(v_4) = w_7$, $f(v_5) = w_4$, $f(v_6) = w_1$, $f(v_7) = w_3$, $f(v_8) = w_8$.

32. Not isomorphic. T_1 has a vertex (v_3) on level 1 of degree 3, but T_2 does not.

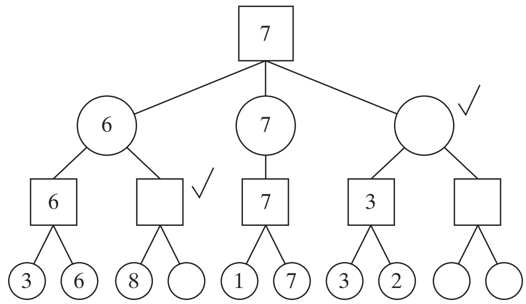
33. $3 - 1 = 2$

34. Let each row, column, or diagonal that contains one X and two blanks count 1. Let each row, column, or diagonal that contains two X's and one blank count 5. Let each row, column, or diagonal that contains three X's count 100. Let each row, column, or diagonal that contains one O and two blanks count -1. Let each row, column, or diagonal that contains two O's and one blank count -5. Let each row, column, or diagonal that contains three O's count -100. Sum the values obtained.

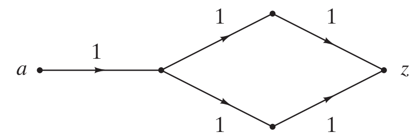
35.



36.



10.



Section 10.2 Review

1. A maximal flow is a flow with maximum value.
2. Ignoring the direction of edges, let $P = (v_0, \ldots, v_n)$ be a path from the source to the sink. If an edge in P is directed from v_{i-1} to v_i, we say that it is properly oriented with respect to P.
3. Ignoring the direction of edges, let $P = (v_0, \ldots, v_n)$ be a path from the source to the sink. If an edge in P is directed from v_i to v_{i-1}, we say that it is improperly oriented with respect to P.
4. We can increase the flow in a path when every properly oriented edge is under capacity and every improperly oriented edge has positive flow.
5. Let Δ be the minimum of the numbers $C_{ij} - F_{ij}$, for properly oriented edges (i, j) in the path, and F_{ij}, for improperly oriented edges (i, j) in the path. Then the flow can be increased by Δ by adding Δ to the flow in each properly oriented edge and by subtracting Δ from the flow in each improperly oriented edge.
6. Start with a flow (e.g., assign each edge flow zero). Search for a path as described in Exercise 4. Increase the flow in such a path as described in Exercise 5.

Section 10.1 Review

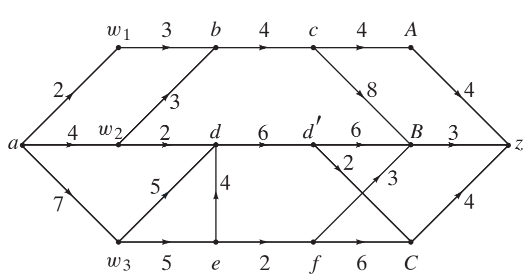
1. A network is a simple, weighted, directed graph with a designated vertex having no incoming edges, a designated vertex having no outgoing edges, and nonnegative weights.
2. A source is a vertex with no incoming edges.
3. A sink is a vertex with no outgoing edges.
4. The weight of an edge is called its capacity.
5. A flow assigns each edge a nonnegative number that does not exceed the capacity of the edge such that for each vertex v, which is neither the source nor the sink, the flow into v equals the flow out of v.
6. The flow in an edge is the nonnegative number assigned to it as in Exercise 5.
7. If F_{ij} is the flow in edge (i, j), the flow into vertex j is $\sum_i F_{ij}$.
8. If F_{ij} is the flow in edge (i, j), the flow out of vertex i is $\sum_j F_{ij}$.
9. Conservation of flow refers to the equality of the flow into and out of a vertex.
10. They are equal.
11. If a network has multiple sources, they can be tied together into a single vertex called the supersource.
12. If a network has multiple sinks, they can be tied together into a single vertex called the supersink.

Section 10.1

1. (b, c) is 6, 3; (a, d) is 4, 2; (c, e) is 6, 1; (c, z) is 5, 2. The value of the flow is 5.

4. Add edges (a, w_1), (a, w_2), (a, w_3), (A, z), (B, z), and (C, z) each having capacity ∞.

7.

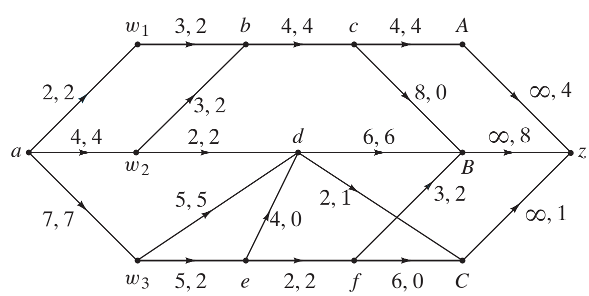


Section 10.2

1. 1

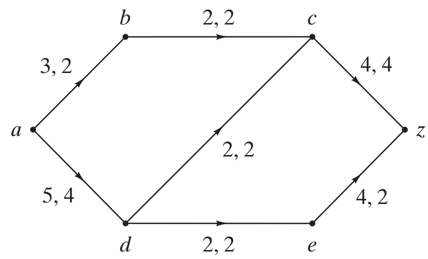
4. $(a, w_1) - 6$, $(a, w_2) - 0$, $(a, w_3) - 3$, $(w_1, b) - 6$, $(w_2, b) - 0$, $(w_3, d) - 3$, $(d, c) - 3$, $(b, c) - 2$, $(b, A) - 4$, $(c, A) - 2$, $(c, B) - 3$, $(A, z) - 6$, $(B, z) - 3$

7.



10. $(a, A-7{:}00) - 3000$, $(a, A-7{:}15) - 3000$, $(a, A-7{:}30) - 2000$, $(A-7{:}00, B-7{:}30) - 1000$, $(A-7{:}00, C-7{:}15) - 2000$, $(A-7{:}15, B-7{:}45) - 1000$, $(A-7{:}15, C-7{:}30) - 2000$, $(A-7{:}30, C-7{:}45) - 2000$, $(B-7{:}30, D-7{:}45) - 1000$, $(C-7{:}15, D-7{:}30) - 2000$, $(B-7{:}45, D-8{:}00) - 1000$, $(C-7{:}30, D-7{:}45) - 2000$, $(C-7{:}45, D-8{:}00) - 2000$, $(D-7{:}45, z) - 3000$, $(D-7{:}30, z) - 2000$, $(D-8{:}00, z) - 3000$. All other edges have flow equal to 0.

13.



16. The maximum flow is 9.

19. Suppose that the sum of the capacities of the edges incident on a is U. Each iteration of Algorithm 10.2.5 increases the flow by 1. Since the flow cannot exceed U, eventually the algorithm must terminate.

Section 10.3 Review

1. A cut in a network consists of a set P of vertices and the complement $\overline{P}$ of P, where the source is in P and the sink is in $\overline{P}$.

2. The capacity of a cut $(P, \overline{P})$ is the number

$$\sum_{i \in P} \sum_{j \in \overline{P}} C_{ij}.$$

3. The capacity of any cut is greater than or equal to the value of any flow.

4. A minimal cut is a cut having minimum capacity.

5. If the value of a flow equals the capacity of a cut, then the flow is maximal and the cut is minimal. The value of a flow F equals the capacity of a cut $(P, \overline{P})$ if and only if $F_{ij} = C_{ij}$ for all $i \in P$, $j \in \overline{P}$, and $F_{ij} = 0$ for all $i \in \overline{P}$, $j \in P$.

6. Let P be the set of labeled vertices, and let $\overline{P}$ be the set of unlabeled vertices at the termination of Algorithm 10.2.4. It can be shown that the conditions

- $F_{ij} = C_{ij}$ for all $i \in P$, $j \in \overline{P}$
- $F_{ij} = 0$ for all $i \in \overline{P}$, $j \in P$

of Exercise 5 hold. Thus the flow is maximal.

Section 10.3

1. 8; minimal

4. $P = \{a, b, d\}$

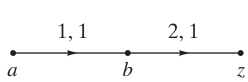
7. $P = \{a, d\}$

10. $P = \{a, w_1, w_2, w_3, b, d, e\}$

13. $P = \{a, w_1, w_2, w_3, b, c, d, d', e, f, A, B, C\}$

16. $P = \{a, b, c, f, g, h, j, k, l, m\}$

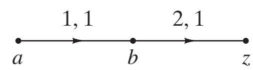
17.



with $C_{ab} = 1, C_{bz} = 2, m_{ab} = 1, m_{bz} = 2$.

20. Alter Algorithm 10.2.4.

23. False. Consider the flow



and the cut $P = \{a, b\}$.

Section 10.4 Review

In the solutions to Exercises 1–5, G is a directed, bipartite graph with disjoint vertex sets V and W in which the edges are directed from V to W.

1. A matching for G is a set of edges with no vertices in common.

2. A maximal matching for G is a matching containing the maximum number of edges.

3. A complete matching for G is a matching E having the property that if $v \in V$, then $(v, w) \in E$ for some $w \in W$.

4. Add a supersource a and edges from a to each vertex in V. Add a supersink z and edges from each vertex in W to z. Assign all edges capacity 1. We call the resulting network a matching network. Then, a flow in the matching network gives a matching in G [v is matched with w if and only if the flow in edge (v, w) is 1]; a maximal flow corresponds to a maximal matching; and a flow whose value is $|V|$ corresponds to a complete matching.

5. If $S \subseteq V$, let

$$R(S) = \{w \in W \mid v \in S \text{ and } (v, w) \text{ is an edge in } G\}.$$

Hall's Marriage Theorem states that there exists a complete matching in G if and only if $|S| \le |R(S)|$ for all $S \subseteq V$.

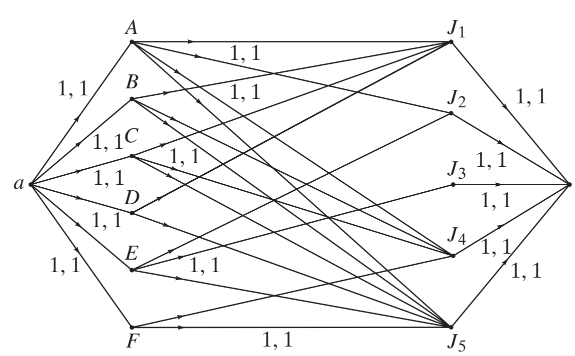
Section 10.4

1. $P = \{a, A, B, D, J_2, J_5\}$

3. Finding qualified persons for jobs

6. Finding qualified persons for all jobs

9. All unlabeled edges are 1, 0. There is no complete matching.



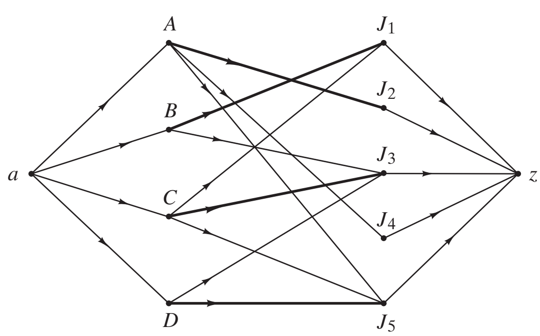
13. Each row and column has at most one label.

17. If $\delta(G) = 0$, then $|S| - |R(S)| \leq 0$, for all $S \subseteq V$. By Theorem 10.4.7, G has a complete matching.

If G has a complete matching, then $|S| - |R(S)| \leq 0$, for all $S \subseteq V$, so $\delta(G) \leq 0$. If $S = \varnothing$, $|S| - |R(S)| = 0$, so $\delta(G) = 0$.

Chapter 10 Self-Test

1. In each edge, the flow is less than or equal to the capacity and, except for the source and sink, the flow into each vertex v is equal to the flow out of v.
2. 3
3. 3
4. 3
5. (a, b, e, f, g, z)
6. Change the flows to $F_{a,b} = 2$, $F_{e,b} = 1$, $F_{e,f} = 1$, $F_{f,g} = 1$, $F_{g,z} = 1$.
7. $F_{a,b} = 3$, $F_{b,c} = 3$, $F_{c,d} = 4$, $F_{d,z} = 4$, $F_{a,e} = 2$, $F_{e,f} = 2$, $F_{f,c} = 2$, $F_{f,g} = 1$, $F_{g,z} = 1$, and all other edge flows zero.
8. $F_{a,b} = 0$, $F_{b,c} = 5$, $F_{c,d} = 5$, $F_{d,z} = 8$, $F_{e,b} = 3$, $F_{g,d} = 3$, $F_{a,e} = 8$, $F_{e,f} = 3$, $F_{f,g} = 3$, $F_{a,h} = 4$, $F_{e,i} = 2$, $F_{j,z} = 6$, $F_{h,i} = 4$, $F_{i,j} = 6$, and all other edge flows zero.
9. a—True, b—False, c—False, d—True
10. 6
11. No. The capacity of $(P, \overline{P})$ is 6, but the capacity of $(P', \overline{P'})$, $P' = \{a, b, c, e, f\}$, is 5.
12. $P = \{a, b, c, e, f, g, h, i\}$
13.



14. See the solution to Exercise 13.
15. $A - J_2$, $B - J_1$, $C - J_3$, $D - J_5$ is a complete matching.
16. $P = \{a\}$

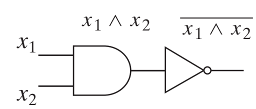
Section 11.1 Review

1. A combinatorial circuit is a circuit in which the output is uniquely defined for every combination of inputs.
2. A sequential circuit is a circuit in which the output is a function of the input and state of the system.
3. An AND gate receives input x_1 and x_2, where x_1 and x_2 are bits, and produces output 1 if x_1 and x_2 are both 1, and 0 otherwise.
4. An OR gate receives input x_1 and x_2, where x_1 and x_2 are bits, and produces output 0 if x_1 and x_2 are both 0, and 1 otherwise.
5. A NOT gate receives input x, where x is a bit, and produces output 1 if x is 0, and 0 if x is 1.
6. An inverter is a NOT gate.
7. A logic table of a combinatorial circuit lists all possible inputs together with the resulting outputs.
8. Boolean expressions in the symbols $x_1, \ldots, x_n$ are defined recursively as follows. $0, 1, x_1, \ldots, x_n$ are Boolean expressions. If X_1 and X_2 are Boolean expressions, then (X_1), $\overline{X_1}$, $X_1 \vee X_2$, and $X_1 \wedge X_2$ are Boolean expressions.
9. A literal is the symbol x or $\overline{x}$ that appears in a Boolean expression.

Section 11.1

1. $\overline{x_1 \wedge x_2}$

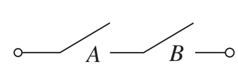
x_1	x_2	$\overline{x_1 \wedge x_2}$
1	1	0
1	0	1
0	1	1
0	0	1



4.

x_1	x_2	x_3	$((x_1 \wedge x_2) \vee \overline{(x_1 \wedge x_3)}) \wedge \overline{x}_3$
1	1	1	0
1	1	0	1
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	0
0	0	0	1

7. If $x = 1$, the output y is undetermined: Suppose that $x = 1$ and $y = 0$. Then the input to the AND gate is 1, 0. Thus the output of the AND gate is 0. Since this is then NOTed, $y = 1$. Contradiction. Similarly, if $x = 1$ and $y = 1$, we obtain a contradiction.
10. 0
13. 1
16. Is a Boolean expression. x_1, x_2, and x_3 are Boolean expressions by (11.1.2). $x_2 \vee x_3$ is a Boolean expression by (11.1.3c). $(x_2 \vee x_3)$ is a Boolean expression by (11.1.3a). $x_1 \wedge (x_2 \vee x_3)$ is a Boolean expression by (11.1.3d).
19. Not a Boolean expression
22.



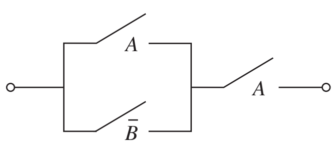
25. $(A \wedge B) \vee (C \wedge \overline{A})$

A	B	C	$(A \wedge B) \vee (C \wedge \overline{A})$
1	1	1	1
1	1	0	1
1	0	1	0
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	1
0	0	0	0

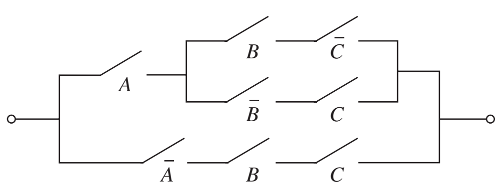
27. $(A \wedge (C \vee (D \wedge C))) \vee (B \wedge (\overline{D} \vee (C \wedge A) \vee \overline{C}))$

29.

A	B	$(A \vee \overline{B}) \wedge A$
1	1	1
1	0	1
0	1	0
0	0	0



32.



Section 11.2 Review

1. $(a \vee b) \vee c = a \vee (b \vee c)$, $(a \wedge b) \wedge c = a \wedge (b \wedge c)$
2. $a \vee b = b \vee a$, $a \wedge b = b \wedge a$
3. $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$, $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$
4. $a \vee 0 = a$, $a \wedge 1 = a$
5. $a \vee \overline{a} = 1$, $a \wedge \overline{a} = 0$
6. Boolean expressions are equal if they have the same values for all possible assignments of bits to the literals.
7. Combinatorial circuits are equivalent if, whenever the circuits receive the same inputs, they produce the same outputs.
8. Let C_1 and C_2 be combinatorial circuits represented, respectively, by the Boolean expressions X_1 and X_2. Then C_1 and C_2 are equivalent if and only if $X_1 = X_2$.

Section 11.2

1.

x_1	x_2	$\overline{x_1 \wedge x_2}$	$\overline{x}_1 \vee \overline{x}_2$
1	1	0	0
1	0	1	1
0	1	1	1
0	0	1	1

4.

x_1	x_2	x_3	$\overline{x}_1 \vee (\overline{x}_2 \vee x_3)$	$\overline{(x_1 \wedge x_2)} \vee x_3$
1	1	1	1	1
1	1	0	0	0
1	0	1	1	1
1	0	0	1	1
0	1	1	1	1
0	1	0	1	1
0	0	1	1	1
0	0	0	1	1

6.

x_1	$x_1 \vee x_1$
1	1
0	0

9.

x_1	x_2	x_3	$x_1 \wedge \overline{(x_2 \wedge x_3)}$	$(x_1 \wedge \overline{x}_2) \vee (x_1 \wedge \overline{x}_3)$
1	1	1	0	0
1	1	0	1	1
1	0	1	1	1
1	0	0	1	1
0	1	1	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	0	0

11.

x	$\overline{\overline{x}}$
1	1
0	0

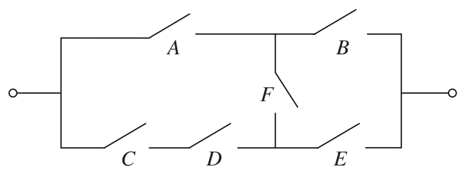
14. False. Take $x_1 = 1$, $x_2 = 1$, $x_3 = 0$.

16.

a	b	c	$a \vee (b \wedge c)$	$(a \vee b) \wedge (a \vee c)$
1	1	1	1	1
1	1	0	1	1
1	0	1	1	1
1	0	0	1	1
0	1	1	1	1
0	1	0	0	0
0	0	1	0	0
0	0	0	0	0

18. The Boolean expressions that represent the circuits are $(A \wedge \overline{B}) \vee (A \wedge C)$ and $A \wedge (\overline{B} \vee C)$. The expressions are equal by Theorem 11.2.1(c). Therefore, the switching circuits are equivalent.

21.



Section 11.3 Review

1. A Boolean algebra consists of a set S containing distinct elements 0 and 1, binary operators $+$ and $\cdot$, and a unary operator $'$ on S satisfying the associative, commutative, distributive, identity, and complement laws.
2. $x + x = x$, $xx = x$
3. $x + 1 = 1$, $x0 = 0$
4. $x + xy = x$, $x(x + y) = x$
5. $(x')' = x$
6. $0' = 1$, $1' = 0$
7. $(x + y)' = x'y'$, $(xy)' = x' + y'$
8. The dual of a Boolean expression is obtained by replacing 0 by 1, 1 by 0, $+$ by $\cdot$, and $\cdot$ by $+$.
9. The dual of a theorem about Boolean algebras is also a theorem.

Section 11.3

2. One can show that the Associative and Distributive Laws hold for lcm and gcd directly. The Commutative Law clearly holds. To see that the Identity Laws hold, note that

$$\operatorname{lcm}(x, 1) = x \quad \text{and} \quad \gcd(x, 6) = x.$$

Since

$$\operatorname{lcm}(x, 6/x) = 6 \quad \text{and} \quad \gcd(x, 6/x) = 1,$$

the Complement Laws hold. Therefore, $(S, +, \cdot, ', 1, 6)$ is a Boolean algebra.

4. We show only

$$x \cdot (x + z) = (x \cdot y) + (x \cdot z) \quad \text{for all } x, y, z \in S_n.$$

Now

$$x \cdot (y + z) = \min\{x, \max\{y, z\}\}$$
$$(x \cdot y) + (x \cdot z) = \max\{\min\{x, y\}, \min\{x, z\}\}.$$

We assume that $y \le z$. (The argument is similar if $y > z$.) There are three cases to consider: $x < y$; $y \le x \le z$; and $z < x$.

If $x < y$, we obtain

$$\begin{aligned} x \cdot (y + z) &= \min\{x, \max\{y, z\}\} \\ &= \min\{x, z\} = x = \max\{x, x\} \\ &= \max\{\min\{x, y\}, \min\{x, z\}\} \\ &= (x \cdot y) + (x \cdot z). \end{aligned}$$

If $y \le x \le z$, we obtain

$$\begin{aligned} x \cdot (y + z) &= \min\{x, \max\{y, z\}\} \\ &= \min\{x, z\} = x = \max\{y, x\} \\ &= \max\{\min\{x, y\}, \min\{x, z\}\} \\ &= (x \cdot y) + (x \cdot z). \end{aligned}$$

If $z < x$, we obtain

$$\begin{aligned} x \cdot (y + z) &= \min\{x, \max\{y, z\}\} \\ &= \min\{x, z\} = z = \max\{y, z\} \\ &= \max\{\min\{x, y\}, \min\{x, z\}\} \\ &= (x \cdot y) + (x \cdot z). \end{aligned}$$

7. If $X \cup Y = U$ and $X \cap Y = \varnothing$, then $Y = \overline{X}$.

8. $xy + x0 = x(x + y)y$

11. $x + y' = 1$ if and only if $x + y = x$.

14. $x(x + y0) = x$

15. (For Exercise 12)

$$\begin{aligned} 0 = x + y &= (x + x) + y \\ &= x + (x + y) = x + 0 = x \end{aligned}$$

Similarly, $y = 0$.

18. [For part (c)]

$$\begin{aligned} x(x + y) &= (x + 0)(x + y) \\ &= x + 0y = x + y0 = x + 0 = x \end{aligned}$$

21. First, show that if $ba = ca$ and $ba' = ca'$, then $b = c$. Now take $a = x$, $b = x + (y + z)$, and $c = (x + y) + z$ and use this result.

23. If the prime p divides n, p^2 does not divide n.

Section 11.4 Review

1. The exclusive-OR of x_1 and x_2 is 0 if $x_1 = x_2$, and 1 otherwise.
2. A Boolean function is a function of the form
$$f(x_1, \ldots, x_n) = X(x_1, \ldots, x_n),$$
where X is a Boolean expression.
3. A minterm is a Boolean expression of the form
$$y_1 \wedge y_2 \wedge \cdots \wedge y_n,$$
where each y_i is either x_i or $\overline{x_i}$.
4. The disjunctive normal form of a not identically zero Boolean function f is
$$f(x_1, \ldots, x_n) = m_1 \vee m_2 \vee \cdots \vee m_k,$$
where each m_i is a minterm.
5. Let $A_1, \ldots, A_k$ denote the elements A_i of Z_2^n for which $f(A_i) = 1$. For each $A_i = (a_1, \ldots, a_n)$, set $m_i = y_1 \wedge \cdots \wedge y_n$, where $y_j = x_j$ if $a_j = 1$, and $y_j = \overline{x_j}$ if $a_j = 0$. Then
$$f(x_1, \ldots, x_n) = m_1 \vee m_2 \vee \cdots \vee m_k.$$

6. A maxterm is a Boolean expression of the form

$$y_1 \vee y_2 \vee \cdots \vee y_n,$$

where each y_i is either x_i or $\overline{x_i}$.

7. The conjunctive normal form of a not identically one Boolean function f is

$$f(x_1, \ldots, x_n) = m_1 \wedge m_2 \wedge \cdots \wedge m_k,$$

where each m_i is a maxterm.

Section 11.4

In these hints, $a \wedge b$ is written ab.

1. $xy \vee \overline{x}y \vee \overline{x}\,\overline{y}$

4. $xyz \vee xy\overline{z} \vee x\overline{y}\,\overline{z} \vee \overline{x}yz \vee \overline{x}y\overline{z}$

7. $xyz \vee x\overline{y}\,\overline{z} \vee \overline{x}\,\overline{y}\,\overline{z}$

10. $wx\overline{y}z \vee wx\overline{y}\,\overline{z} \vee w\overline{x}yz \vee w\overline{x}y\overline{z} \vee w\overline{x}\,\overline{y}\,\overline{z}$
$\vee \overline{w}xy\overline{z} \vee \overline{w}x\overline{y}z \vee \overline{w}x\overline{y}\,\overline{z} \vee \overline{w}\,\overline{x}y\overline{z} \vee \overline{w}\,\overline{x}\,\overline{y}\,\overline{z}$

11. $xy \vee x\overline{y}$

14. $xy\overline{z}$

17. $xyz \vee \overline{x}yz \vee xy\overline{z} \vee \overline{x}y\overline{z}$

20. 0

22. 2^{2^n}

25. (For Exercise 3)

$$(\overline{x} \vee y \vee \overline{z})(x \vee \overline{y} \vee \overline{z})(x \vee \overline{y} \vee z)$$

28. (For Exercise 3)

$$(\overline{x} \vee \overline{y} \vee \overline{z})(\overline{x} \vee \overline{y} \vee z)(\overline{x} \vee y \vee z)(x \vee y \vee \overline{z})(x \vee y \vee z)$$

Section 11.5 Review

1. A gate is a function from Z_2^n into Z_2.
2. A set of gates G is functionally complete if, given any positive integer n and a function f from Z_2^n into Z_2, it is possible to construct a combinatorial circuit that computes f using only the gates in G.
3. {AND, OR, NOT}
4. A NAND gate receives input x_1 and x_2, where x_1 and x_2 are bits, and produces output 0 if x_1 and x_2 are both 1, and 1 otherwise.
5. Yes
6. The problem of finding the best circuit
7. Small components that are themselves entire circuits
8. See Figure 11.5.8.
9. See Figure 11.5.9.

Section 11.5

1. AND can be expressed in terms of OR and NOT: $xy = \overline{\overline{x} \vee \overline{y}}$.

2. A combinatorial circuit consisting only of AND gates would always output 0 when all inputs are 0.

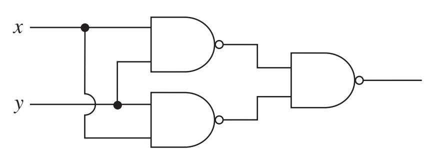
5. We use induction on n to show that there is no n-gate combinatorial circuit consisting of only AND and OR gates that computes $f(x) = \overline{x}$.

If $n = 0$, the input x equals the output x, and so it is impossible for a 0-gate circuit to compute f. The Basis Step is proved.

Suppose that there is no n-gate combinatorial circuit consisting of only AND and OR gates that computes f. Consider an $(n+1)$-gate combinatorial circuit consisting of only AND and OR gates. The input x first arrives at either an AND or an OR gate. Suppose that x first arrives at an AND gate. (The argument is similar if x first arrives at an OR gate and is omitted.) Because the circuit is a combinatorial circuit, the other input to the AND gate is either x itself, the constant 1, or the constant 0. If both inputs to the AND gate are x itself, then the output of the AND gate is equal to the input. In this case, the behavior of the circuit is unchanged if we remove the AND gate and connect x to what was the output line of the AND gate. But we now have an equivalent n-gate circuit, which, by the inductive hypothesis, cannot compute f. Thus the $(n+1)$-gate circuit cannot compute f.

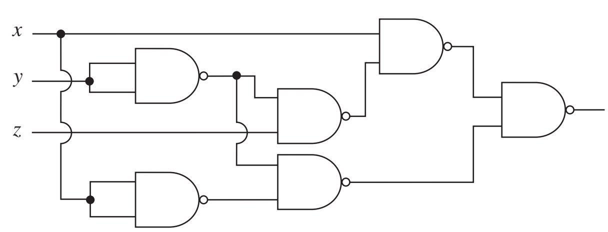
If the other input to the AND gate is the constant 1, the output of the AND gate is again equal to the input and we can argue as in the previous case that the $(n+1)$-gate circuit cannot compute f.

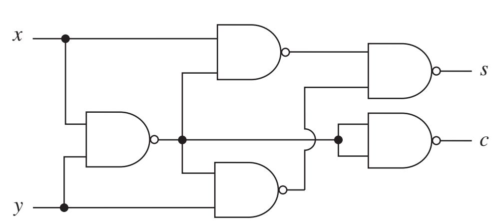
If the other input to the AND gate is the constant 0, the AND gate always outputs 0 and, so, changing the value of x does not affect the output of the circuit. In this case, the circuit cannot compute f. The Inductive Step is complete. Therefore, no n-gate combinatorial circuit consisting of only AND and OR gates can compute $f(x) = \overline{x}$. Thus {AND, OR} is not functionally complete.

6. 

9. $y_1 = x_1x_2 \vee \overline{(x_2 \vee x_3)}$; $y_2 = \overline{x_2 \vee x_3}$

12. (For Exercise 3) The dnf may be simplified to $xy \vee x\overline{z} \vee \overline{x}\,\overline{y}$ and then rewritten as $x(y \vee \overline{z}) \vee \overline{x}\,\overline{y} = (x\overline{\overline{y}z}) \vee \overline{x}\,\overline{y} = \overline{\overline{x\overline{\overline{y}z}}\;\overline{\overline{x}\,\overline{y}}}$, which gives the circuit



15. 

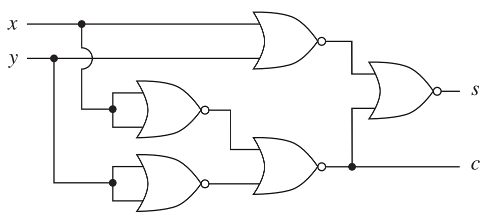
17. $xy = (x \downarrow x) \downarrow (y \downarrow y)$
$x \vee y = (x \downarrow y) \downarrow (x \downarrow y)$ $\overline{x} = x \downarrow x$
$x \uparrow y = [(x \downarrow x) \downarrow (y \downarrow y)] \downarrow [(x \downarrow x) \downarrow (y \downarrow y)]$

20. Since

$$\overline{x} = x \downarrow x, \quad x \vee y = (x \downarrow y) \downarrow (x \downarrow y),$$

and {NOT, OR} is functionally complete, {NOR} is functionally complete.

23.



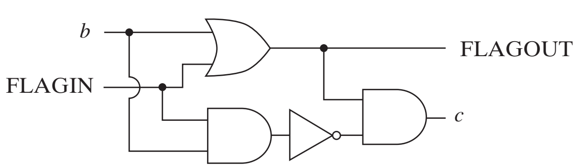
25. The logic table is

x	y	z	*Output*
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	0
0	0	0	0

27. The logic table is

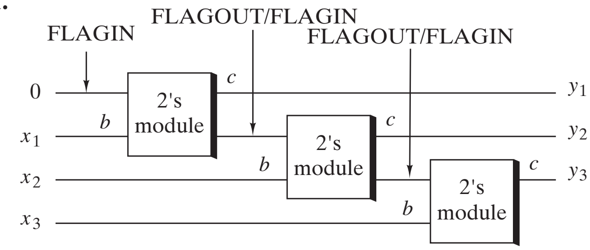
b	FLAGIN	c	FLAGOUT
1	1	0	1
1	0	1	1
0	1	1	1
0	0	0	0

Thus $c = b \oplus$ FLAGIN and FLAGOUT $= b \vee$ FLAGIN. We obtain the circuit



28. 010100

31.



34. Writing the truth tables shows that

$$\overline{x} = x \to 0, \qquad x \vee y = (x \to 0) \to y.$$

Therefore a NOT gate can be replaced by one $\to$ gate, and an OR gate can be replaced by two $\to$ gates. Since the set {NOT, OR} is functionally complete, it follows that the set $\{\to\}$ is functionally complete.

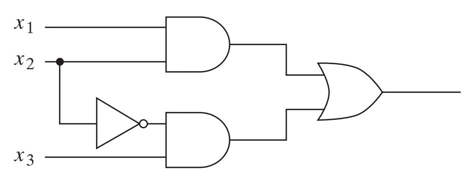
Chapter 11 Self-Test

1.

x	y	z	$\overline{(x \wedge \overline{y})} \vee z$
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	1

2. 1

3.



4. Suppose that x is 1. Then the upper input to the OR gate is 0. If y is 1, then the lower input to the OR gate is 0. Since both inputs to the OR gate are 0, the output y of the OR gate is 0, which is impossible. If y is 0, then the lower input to the OR gate is 1. Since an input to the OR gate is 1, the output y of the OR gate is 1, which is impossible. Therefore, if the input to the circuit is 1, the output is not uniquely determined. Thus the circuit is not a combinatorial circuit.

5. The circuits are equivalent. The logic table for either circuit is

x	y	*Output*
1	1	0
1	0	1
0	1	0
0	0	0

6. The circuits are not equivalent. If $x = 0$, $y = 1$, and $z = 0$, the output of circuit (a) is 1, but the output of circuit (b) is 0.

7. The equation is true. The logic table for either expression is

x	y	z	*Value*
1	1	1	1
1	1	0	1
1	0	1	0
1	0	0	0
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	0

8. The equation is false. If $x = 1$, $y = 0$, and $z = 1$, then

$$(x \wedge y \wedge z) \vee \overline{(x \vee z)} = 0,$$

but

$$(x \wedge z) \vee (\overline{x} \wedge \overline{z}) = 1.$$

9. Bound laws:

$$X \cup U = U, \quad X \cap \varnothing = \varnothing \quad \text{for all } X \in S.$$

Absorption laws:

$$X \cup (X \cap Y) = X, \quad X \cap (X \cup Y) = X \quad \text{for all } X, Y \in S.$$

10.
$$\begin{aligned} (x(x + y \cdot 0))' &= (x(x + 0))' && \text{(Bound law)} \\ &= (x \cdot x)' && \text{(Identity law)} \\ &= x' && \text{(Idempotent law)} \end{aligned}$$

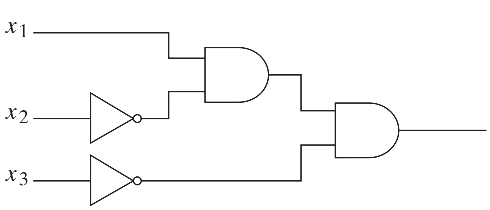
11. Dual: $(x + x(y + 1))' = x'$

$$\begin{aligned} (x + x(y + 1))' &= (x + x \cdot 1)' && \text{(Bound law)} \\ &= (x + x)' && \text{(Identity law)} \\ &= x' && \text{(Idempotent law)} \end{aligned}$$

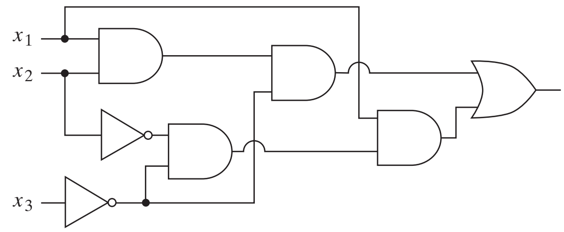
12. $^{-}$ is not a unary operator on S. For example, $\overline{\{1, 2\}} \notin S$.

In Exercises 13–16, $a \wedge b$ is written ab.

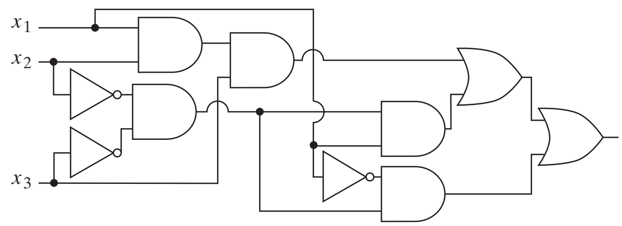
13. $x_1\overline{x}_2\overline{x}_3$



14. $x_1x_2\overline{x}_3 \vee x_1\overline{x}_2\overline{x}_3$



15. $x_1x_2x_3 \vee x_1\overline{x}_2\overline{x}_3 \vee \overline{x}_1\overline{x}_2\overline{x}_3$



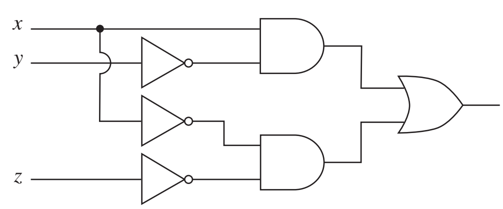
16. $x_1x_2\overline{x}_3 \vee x_1\overline{x}_2\overline{x}_3 \vee \overline{x}_1x_2x_3 \vee \overline{x}_1\overline{x}_2x_3$

17.

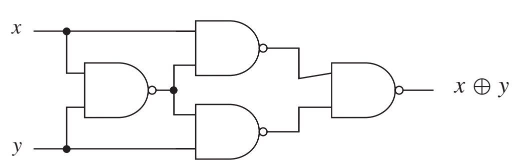
x	y	z	*Output*
1	1	1	1
1	1	0	0
1	0	1	1
1	0	0	0
0	1	1	0
0	1	0	0
0	0	1	1
0	0	0	0

18. Disjunctive normal form: $x\overline{y}z \vee x\overline{y}\,\overline{z} \vee \overline{x}y\overline{z} \vee \overline{x}\,\overline{y}\,\overline{z}$

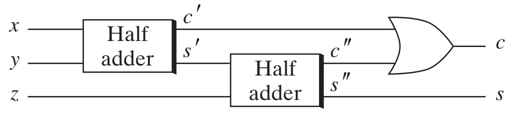
$$\begin{aligned} (x\overline{y}z \vee x\overline{y}\,\overline{z}) \vee \overline{x}y\overline{z} \vee \overline{x}\,\overline{y}\,\overline{z} &= x\overline{y} \vee (\overline{x}y\overline{z} \vee \overline{x}\,\overline{y}\,\overline{z}) \\ &= x\overline{y} \vee \overline{x}\,\overline{z} \end{aligned}$$



19.



20.



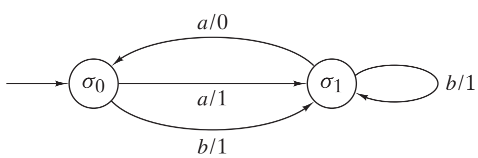
Section 12.1 Review

1. A unit time delay accepts as input a bit x_t at time t and outputs x_{t-1}, the bit received as input at time $t-1$.
2. A serial adder inputs two binary numbers and outputs their sum.
3. A finite-state machine consists of a finite set $\mathcal{I}$ of input symbols, a finite set $\mathcal{O}$ of output symbols, a finite set $\mathcal{S}$ of states, a next-state function f from $\mathcal{S} \times \mathcal{I}$ into $\mathcal{S}$, an output function g from $\mathcal{S} \times \mathcal{I}$ into $\mathcal{O}$, and an initial state $\sigma \in \mathcal{S}$.
4. Let $M = (\mathcal{I}, \mathcal{O}, \mathcal{S}, f, g, \sigma)$ be a finite-state machine. The transition diagram of M is a digraph G whose vertices are the states. An arrow designates the initial state. A directed edge (σ_1, σ_2) exists in G if there exists an input i with $f(\sigma_1, i) = \sigma_2$. In this case, if $g(\sigma_1, i) = o$, the edge (σ_1, σ_2) is labeled i/o.
5. The *SR* flip-flop is defined by the table

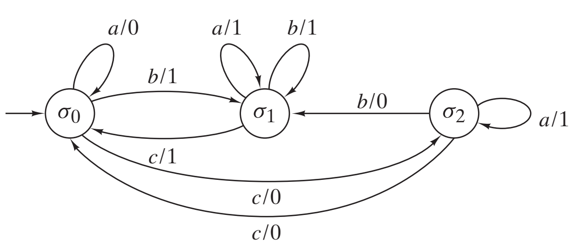
S	R	Q
1	1	Not allowed
1	0	1
0	1	0
0	0	$\begin{cases} 1 & \text{if } S \text{ was last equal to 1} \\ 0 & \text{if } R \text{ was last equal to 1} \end{cases}$

Section 12.1

1.



4.



6. $\mathcal{I} = \{a, b\}$; $\mathcal{O} = \{0, 1\}$; $\mathcal{S} = \{\sigma_0, \sigma_1\}$; initial state $= \sigma_0$

$\mathcal{S}$ \ $\mathcal{I}$	a	b	a	b
σ_0	σ_1	σ_0	0	1
σ_1	σ_1	σ_1	1	1

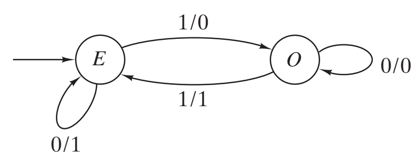
9. $\mathcal{I} = \{a, b\}$; $\mathcal{O} = \{0, 1\}$; $\mathcal{S} = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}$; initial state $= \sigma_0$

$\mathcal{S}$ \ $\mathcal{I}$	a	b	a	b
σ_0	σ_1	σ_2	0	0
σ_1	σ_0	σ_2	1	0
σ_2	σ_3	σ_0	0	1
σ_3	σ_1	σ_3	0	0

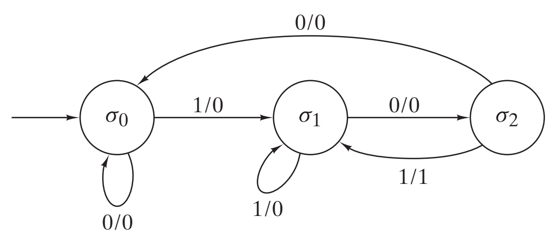
11. 1110 **14.** 001110

17. 001110001 **20.** 020022201020

21.



24.



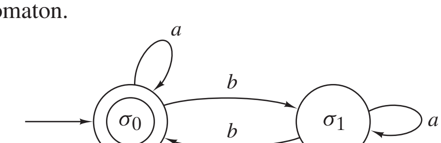
27. When γ is input, the machine outputs $x_n, x_{n-1}, \ldots$ until $x_i = 1$. Thereafter, it outputs $\overline{x}_i$. However, according to Algorithm 11.5.16, this is the 2's complement of α.

Section 12.2 Review

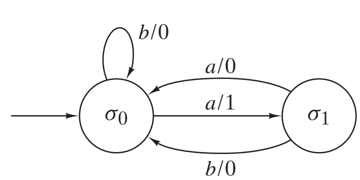
1. A finite-state automaton consists of a finite set $\mathcal{I}$ of input symbols, a finite set $\mathcal{S}$ of states, a next-state function f from $\mathcal{S} \times \mathcal{I}$ into $\mathcal{S}$, a subset $\mathcal{A}$ of $\mathcal{S}$ of accepting states, and an initial state $\sigma \in \mathcal{S}$.
2. A string is accepted by a finite-state automaton A if, when the string is input to A, the last state reached is an accepting state.
3. Finite-state automata are equivalent if they accept precisely the same strings.

Section 12.2

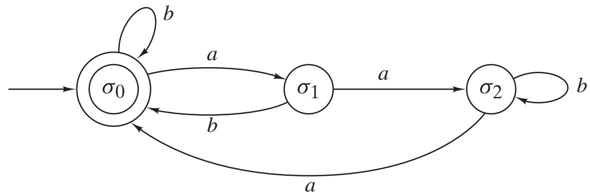
1. All incoming edges to σ_0 output 1 and all incoming edges to σ_1 output 0; hence the finite-state machine is a finite-state automaton.



4.



7.



10. (For Exercise 1) $\mathcal{I} = \{a, b\}$; $\mathcal{S} = \{\sigma_0, \sigma_1\}$; $\mathcal{A} = \{\sigma_0\}$; initial state $= \sigma_0$

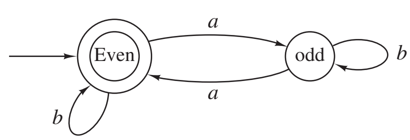
$\mathcal{S}$ \ $\mathcal{I}$	a	b
σ_0	σ_0	σ_1
σ_1	σ_1	σ_0

13. Accepted

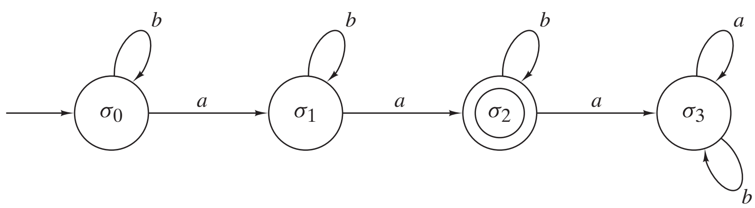
16. Accepted

18. No matter which state we are in, after an a we move to an accepting state; however, after a b we move to a nonaccepting state.

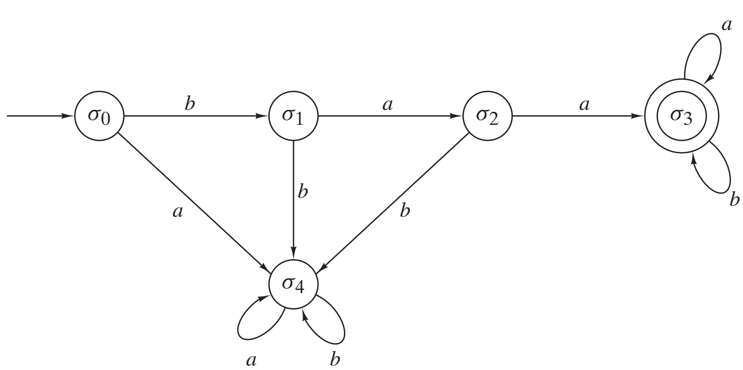
21.



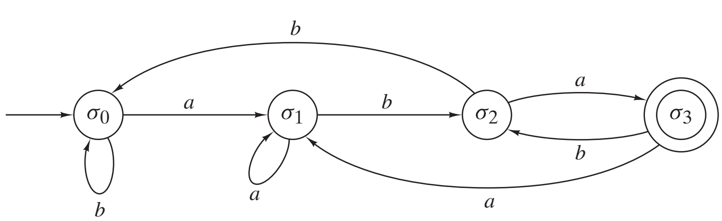
24.



27.



30.



32. (For Exercise 1) This algorithm determines whether a string over $\{a, b\}$ is accepted by the finite-state automaton whose transition diagram is given in Exercise 1.

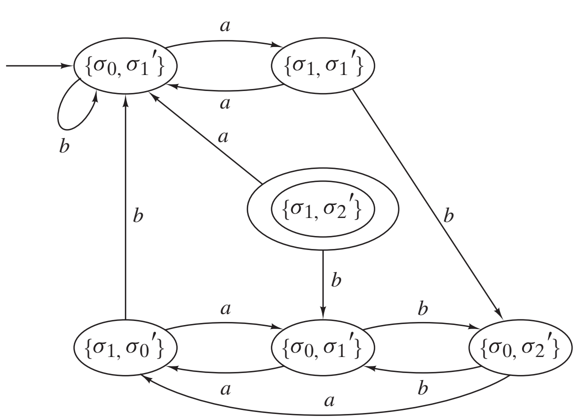
Input: n, the length of the string ($n = 0$ designates the null string); $s_1 \cdots s_n$, the string

Output: "Accept" if the string is accepted
"Reject" if the string is not accepted

```
ex32(s, n) {
   state = 'σ0'
   for i = 1 to n {
      if (state == 'σ0' ∧ s_i == 'b')
         state = 'σ1'
      if (state == 'σ1' ∧ s_i == 'b')
         state = 'σ0'
   }
   if (state == 'σ0')
      return "Accept"
   else
      return "Reject"
}
```

35. Make each accepting state nonaccepting and each nonaccepting state accepting.

38. Using the construction given in Exercises 36 and 37, we obtain the following finite-state automaton that accepts $L_1 \cap L_2$. (We designate the states in Exercise 5 with primes.)



The finite-state automaton that accepts $L_1 \cup L_2$ is the same as the finite-state automaton that accepts $L_1 \cap L_2$ except that the set of accepting states is

$$\{(\sigma_1, \sigma_0'), \quad (\sigma_1, \sigma_1'), \quad (\sigma_1, \sigma_2'), \quad (\sigma_0, \sigma_2')\}.$$

41. Use the construction of Exercises 36 and 37.

Section 12.3 Review

1. A "natural language" refers to ordinary written and spoken words and combinations of words. A "formal language" is an artificial language consisting of a specified set of strings. Formal languages are used to model natural languages and to communicate with computers.

2. A phrase-structure grammar consists of a finite set N of nonterminal symbols, a finite set T of terminal symbols where $N \cap T = \varnothing$, a finite subset of $[(N \cup T)^* - T^*] \times (N \cup T)^*$ called the set of productions, and a starting symbol in N.

3. If $\alpha \to \beta$ is a production and $x\alpha y \in (N \cup T)^*$, we say that $x\beta y$ is directly derivable from $x\alpha y$.

4. If $\alpha_i \in (N \cup T)^*$ for $i = 1, \ldots, n$, and α_{i+1} is directly derivable from α_i for $i = 1, \ldots, n-1$, we say that α_n is derivable from α_1 and write $\alpha_1 \Rightarrow \alpha_n$.

5. We call $\alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_n$ a derivation of α_n from α_1.

6. The language generated by a grammar consists of all strings in terminals derivable from the start symbol.

7. Backus normal form (BNF) is a way to write the productions of a grammar. In BNF the nonterminal symbols typically begin with "⟨" and end with "⟩". Also the arrow → is replaced with ::=. Productions with the same left-hand side are combined using the bar "|". An example is

⟨signed integer⟩ ::=
+⟨unsigned integer⟩ | − ⟨unsigned integer⟩

8. In a context-sensitive grammar, every production is of the form $\alpha A \beta \to \alpha\delta\beta$, where $\alpha, \beta \in (N \cup T)^*$, $A \in N$, and $\delta \in (N \cup T)^* - \{\lambda\}$.

9. In a context-free grammar, every production is of the form $A \to \delta$, where $A \in N$ and $\delta \in (N \cup T)^*$.

10. In a regular grammar, every production is of the form $A \to a$, $A \to aB$, or $A \to \lambda$, where $A, B \in N$ and $a \in T$.

11. A context-sensitive grammar

12. A context-free grammar

13. A regular grammar

14. A language is context-sensitive if there is a context-sensitive grammar that generates it.

15. A language is context-free if there is a context-free grammar that generates it.

16. A language is regular if there is a regular grammar that generates it.

17. A context-free, interactive Lindenmayer grammar consists of a finite set N of nonterminal symbols; a finite set T of terminal symbols where $N \cap T = \varnothing$; a finite set of productions $A \to B$, where $A \in N \cup T$ and $B \in (N \cup T)^*$; and a starting symbol in N.

18. The von Koch snowflake is generated by the context-free, interactive Lindenmayer grammar

$$N = \{D\}$$
$$T = \{d, +, -\}$$
$$P = \{D \to D - D + + D - D,\ D \to d,\ + \to +,\ - \to -\}.$$

d means "draw a straight line of a fixed length in the current direction," + means "turn right by 60°," and − means "turn left by 60°."

19. Fractal curves are characterized by having a part of the whole curve resemble the whole.

Section 12.3

1. Regular, context-free, context-sensitive

4. Context-free, context-sensitive

7. $\sigma \Rightarrow b\sigma \Rightarrow bb\sigma \Rightarrow bbaA \Rightarrow bbabA \Rightarrow bbabbA \Rightarrow bbabba\sigma \Rightarrow bbabbab$

10. $\sigma \Rightarrow ABA \Rightarrow ABBA \Rightarrow ABBAA$
$\Rightarrow ABBaAA \Rightarrow abBBaAA \Rightarrow abbBaAA$
$\Rightarrow abbbaAA \Rightarrow abbbaabA \Rightarrow abbbaabab$

12. (For Exercise 1)

$$<\sigma> ::= b<\sigma> \mid a<A> \mid b$$
$$<A> ::= a<\sigma> \mid b<A> \mid a$$

15. $S \to aA$, $A \to aA$, $A \to bA$, $A \to a$, $A \to b$, $S \to a$

18. $S \to aA$, $S \to bS$, $S \to \lambda$, $A \to aA$, $A \to bB$, $A \to \lambda$, $B \to aA$, $B \to bS$

21. <exp number> ::= <integer> E <integer> |
<float number> |
<float number> E <integer>

24. $S \to aSa$, $S \to bSb$, $S \to a$, $S \to b$, $S \to \lambda$

25. If a derivation begins $S \Rightarrow aSb$, the resulting string begins with a and ends with b. Similarly, if a derivation begins $S \Rightarrow bSa$, the resulting string begins with b and ends with a. Therefore, the grammar does not generate the string $abba$.

28. If a derivation begins $S \Rightarrow abS$, the resulting string begins ab. If a derivation begins $S \Rightarrow baS$, the resulting string begins ba. If a derivation begins $S \Rightarrow aSb$, the resulting string starts with a and ends with b. If a derivation begins $S \Rightarrow bSa$, the resulting string begins with b and ends with a. Therefore, the grammar does not generate the string $aabbabba$.

31. The grammar does generate L, the set of all strings over $\{a, b\}$ with equal numbers of a's and b's.

Any string generated by the grammar has equal numbers of a's and b's since whenever any of the productions are used in a derivation, equal numbers of a's and b's are added to the string.

To prove the converse, we consider an arbitrary string α in L, and we use induction on the length $|\alpha|$ of α to show that α is generated by the grammar. The Basis Step is $|\alpha| = 0$. In this case, α is the null string, and $S \Rightarrow \lambda$ is a derivation of α.

Let α be a nonnull string, and suppose that any string in L whose length is less than $|\alpha|$ is generated by the grammar. We first consider the case that α starts with a. Then α can be written $\alpha = a\alpha_1 b\alpha_2$, where α_1 and α_2 have equal numbers of a's and b's. By the inductive hypothesis, there are derivations $S \Rightarrow \alpha_1$ and $S \Rightarrow \alpha_2$ of α_1 and α_2. But now

$$S \Rightarrow aSbS \Rightarrow a\alpha_1 b\alpha_2$$

is a derivation of α. Similarly, if α starts with b, there is a derivation of α. The Inductive Step is finished, and the proof is complete.

32. Replace each production

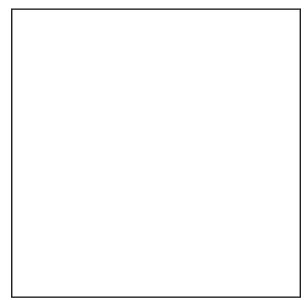
$$A \to x_1 \cdots x_n B,$$

where $n > 1$, $x_i \in T$, and $B \in N$, with the productions

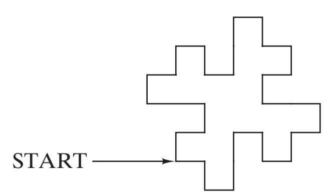
$$\begin{aligned} A &\to x_1 A_1 \\ A_1 &\to x_2 A_2 \\ &\vdots \\ A_{n-1} &\to x_n B, \end{aligned}$$

where $A_1, \ldots, A_{n-1}$ are additional nonterminal symbols.

35. $S \Rightarrow D + D + D + D \Rightarrow d + d + d + d$



$$\begin{aligned} S &\Rightarrow D + D + D + D \\ &\Rightarrow D + D - D - DD + D + D - D \\ &\quad + D + D - D - DD + D + D - D \\ &\quad + D + D - D - DD + D + D - D \\ &\quad + D + D - D - DD + D + D - D \\ &\Rightarrow d + d - d - dd + d + d - d \\ &\quad + d + d - d - dd + d + d - d \\ &\quad + d + d - d - dd + d + d - d \\ &\quad + d + d - d - dd + d + d - d \end{aligned}$$



Section 12.4 Review

1. Let σ be the start state, let T be the set of input symbols, and let N be the set of states. Let P be the set of productions $S \to xS'$, if there is an edge labeled x from S to S', and $S \to \lambda$ if S is an accepting state. Let G be the regular grammar (N, T, P, σ). Then the set of strings accepted by A is equal to $L(G)$.

2. A nondeterministic finite-state automaton consists of a finite set $\mathcal{I}$ of input symbols, a finite set $\mathcal{S}$ of states, a next-state function f from $\mathcal{S} \times \mathcal{I}$ into $\mathcal{P}(\mathcal{S})$, a subset $\mathcal{A}$ of $\mathcal{S}$ of accepting states, and an initial state $\sigma \in \mathcal{S}$.

3. A string α is accepted by a nondeterministic finite-state automaton A if there is some path representing α in the transition diagram of A beginning at the initial state and ending in an accepting state.

4. Nondeterministic finite-state automata are equivalent if they accept precisely the same strings.

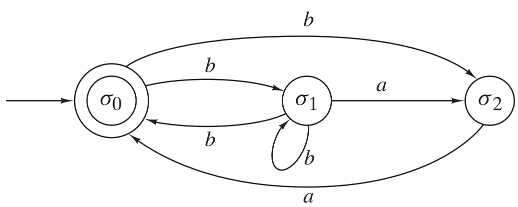
5. Let $G = (N, T, P, \sigma)$ be a regular grammar. The finite-state automaton A is constructed as follows. The set of input symbols is T. The set of states is N together with an additional state $F \notin N \cup T$. The next-state function f is defined as

$$f(S, x) = \{S' \mid S \to xS' \in P\} \cup \{F \mid S \to x \in P\}.$$

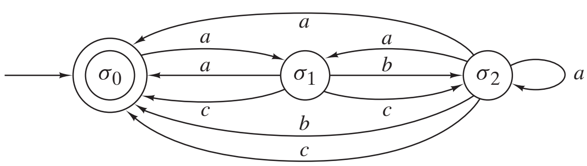
The set of accepting states is F together with all S for which $S \to \lambda$ is a production. Then A accepts precisely the strings $L(G)$.

Section 12.4

1.



4.



6. $\mathcal{I} = \{a, b\}$; $\mathcal{S} = \{\sigma_0, \sigma_1, \sigma_2\}$; $\mathcal{A} = \{\sigma_1, \sigma_2\}$; initial state $= \sigma_0$

$\mathcal{S}$ \ $\mathcal{I}$	a	b
σ_0	$\{\sigma_1, \sigma_2\}$	$\varnothing$
σ_1	$\{\sigma_1\}$	$\{\sigma_0, \sigma_2\}$
σ_2	$\varnothing$	$\varnothing$

9. $\mathcal{I} = \{a, b\}$; $\mathcal{S} = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}$; $\mathcal{A} = \{\sigma_3\}$; initial state $= \sigma_0$

$\mathcal{S}$ \ $\mathcal{I}$	a	b
σ_0	$\{\sigma_0\}$	$\{\sigma_0, \sigma_1\}$
σ_1	$\{\sigma_2\}$	$\varnothing$
σ_2	$\varnothing$	$\{\sigma_3\}$
σ_3	$\{\sigma_3\}$	$\{\sigma_3\}$

11. (For Exercise 5) $N = \{\sigma_0, \sigma_1, \sigma_2\}$, $T = \{a, b\}$,

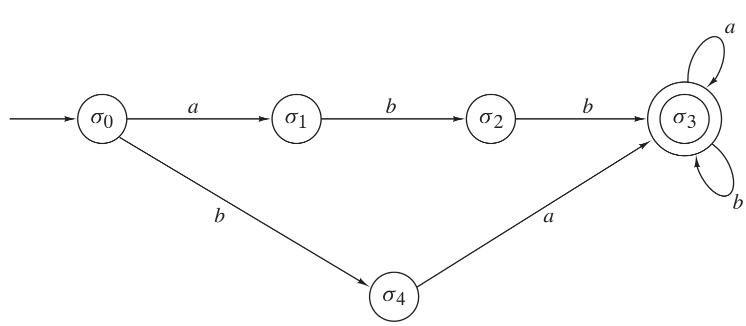
$$\sigma_0 \to a\sigma_1, \quad \sigma_0 \to b\sigma_0, \quad \sigma_1 \to a\sigma_0, \quad \sigma_1 \to b\sigma_2,$$

$$\sigma_2 \to b\sigma_1, \quad \sigma_2 \to a\sigma_0, \quad \sigma_2 \to \lambda$$

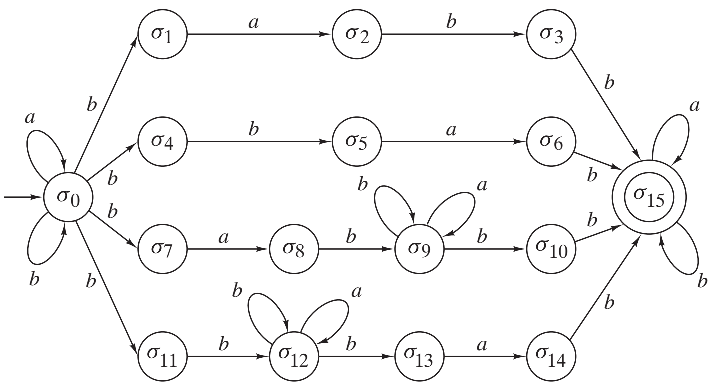
14. No. For the first three characters, bba, the moves are determined and we end at C. From C, no edge contains an a; therefore, $bbabab$ is not accepted.

17. Yes. The path $(\sigma, \sigma, \sigma, \sigma, C, C)$, which represents the string $aaaab$, ends at C, which is an accepting state.

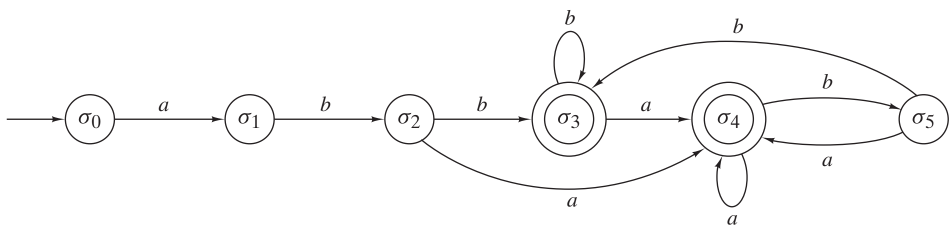
21.



24.



27.



30. (For Exercise 21) $\sigma_0 \to a\sigma_1$, $\sigma_0 \to b\sigma_4$, $\sigma_1 \to b\sigma_2$, $\sigma_2 \to b\sigma_3$, $\sigma_3 \to a\sigma_3$, $\sigma_3 \to b\sigma_3$, $\sigma_4 \to a\sigma_3$, $\sigma_3 \to \lambda$

Section 12.5 Review

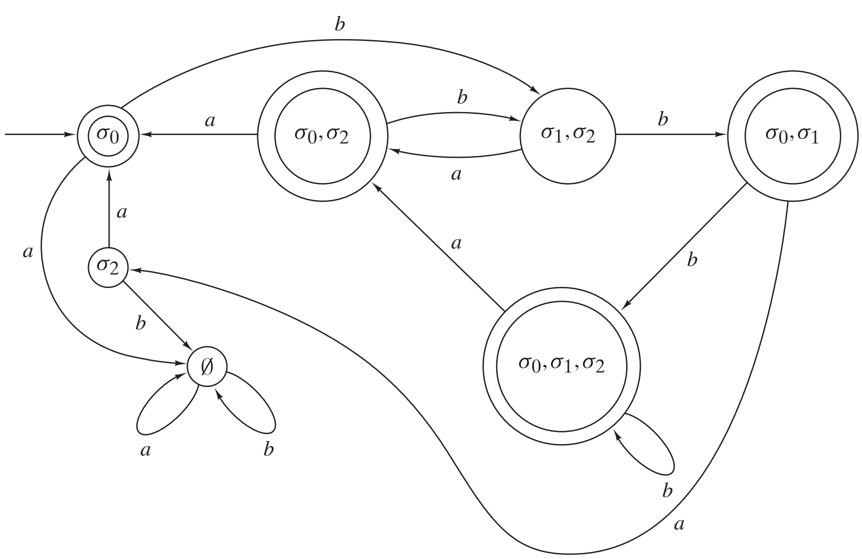
1. Let $A = (\mathcal{I}, \mathcal{S}, f, \mathcal{A}, \sigma)$ be a nondeterministic finite-state automaton. An equivalent deterministic finite-state automaton can be constructed as follows. The set of states is the power set of $\mathcal{S}$. The set of input symbols is $\mathcal{I}$ (unchanged). The start symbol is $\{\sigma\}$ (essentially unchanged). The set of accepting states consists of all subsets of $\mathcal{S}$ that contain at least one accepting state of A. The next state function is defined by the rule

$$f'(X, x) = \begin{cases} \varnothing & \text{if } X = \varnothing \\ \bigcup_{S \in X} f(S, x) & \text{if } X \neq \varnothing. \end{cases}$$

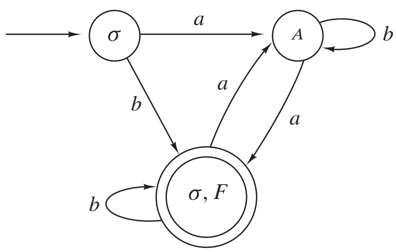
2. A language L is regular if and only if there exists a finite-state automaton that accepts precisely the strings in L.

Section 12.5

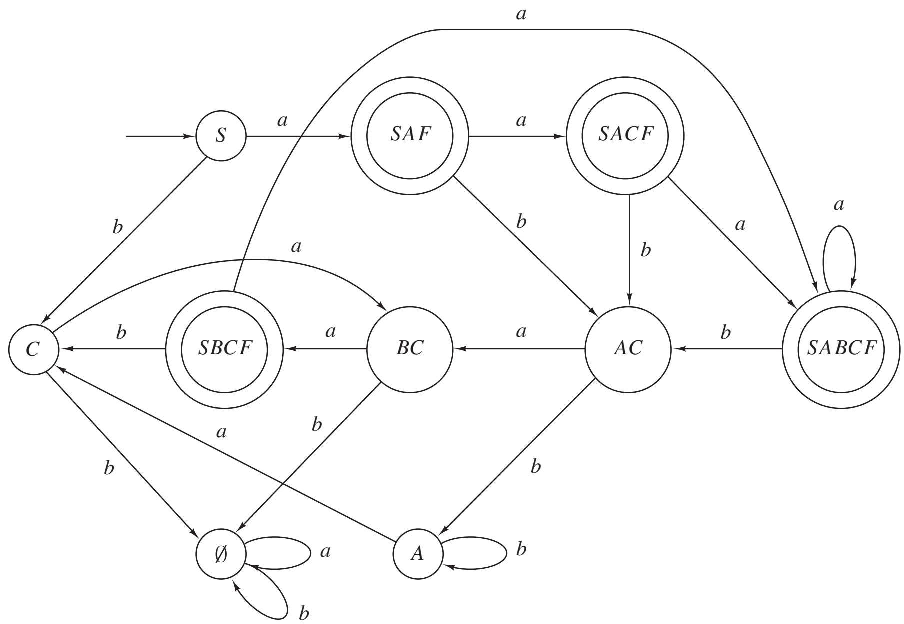
1. (For Exercise 1)



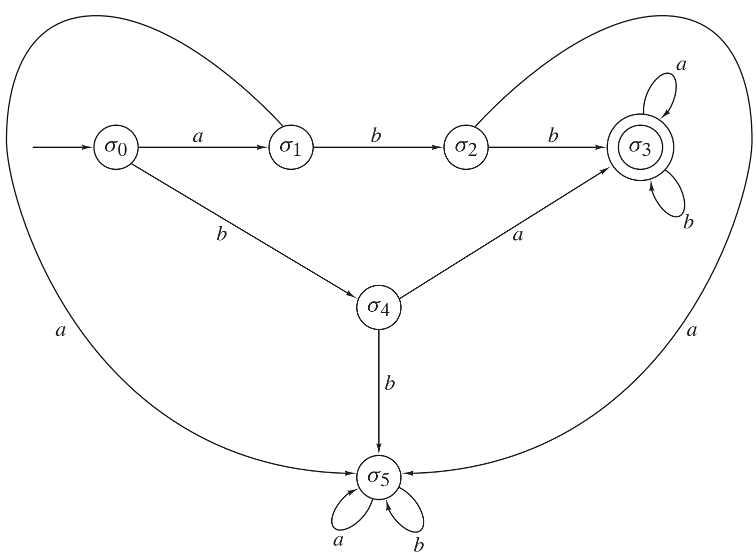
2.



5.

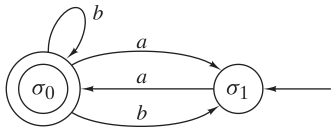


7. (For Exercise 21)

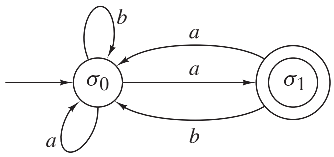


10. Figure 12.5.7 accepts the string ba^n, $n \geq 1$, and strings that end b^2 or aba^n, $n \geq 1$. Using Example 12.5.8, we see that Figure 12.5.9 accepts the string $a^n b$, $n \geq 1$, and strings that start b^2 or $a^n ba$, $n \geq 1$.

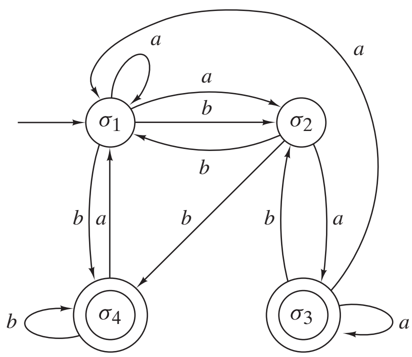
11.



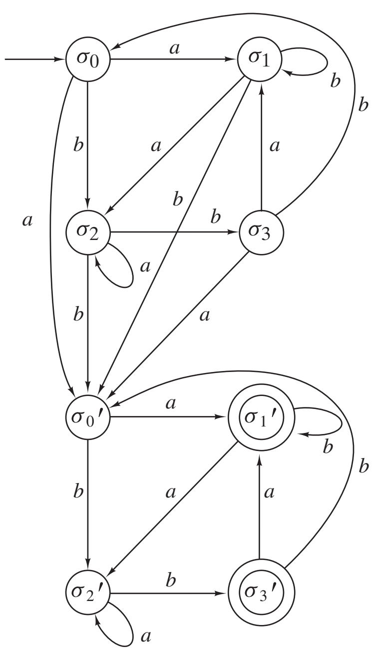
14.



17.



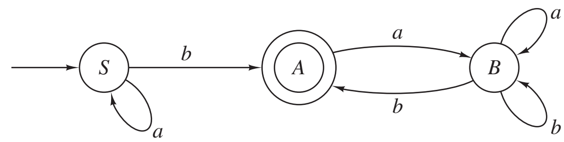
20.



22. $\sigma_0 \to a\sigma_1$, $\sigma_0 \to b\sigma_2$, $\sigma_0 \to a$, $\sigma_1 \to a\sigma_0$, $\sigma_1 \to a\sigma_2$, $\sigma_1 \to b\sigma_1$, $\sigma_1 \to b$, $\sigma_2 \to b\sigma_0$

25. Suppose that L is regular. Then there exists a finite-state automaton A with $L = \text{Ac}(A)$. Suppose that A has k states. Consider the string $a^k bba^k$ and argue as in Example 12.5.6.

28. The statement is false. Consider the regular language $L = \{a^n b \mid n \geq 0\}$, which is accepted by the finite-state automaton



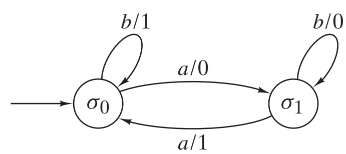
The language

$$L' = \{u^n \mid u \in L, n \in \{1, 2, \ldots\}\}$$

is not regular. Suppose that L' is regular. Then there is a finite-state automaton A that accepts L'. In particular, A accepts $a^n b$ for every n. It follows that for sufficiently large n, the path representing $a^n b$ contains a cycle of length k. Since A accepts $a^n ba^n b$, A also accepts $a^{n+k} ba^n b$, which is a contradiction.

Chapter 12 Self-Test

1.

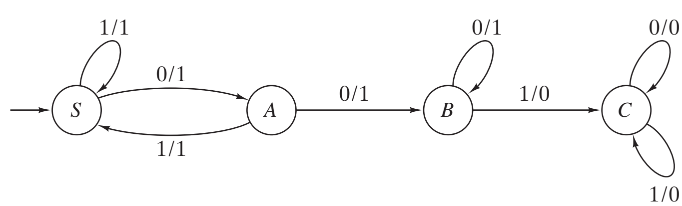


2. $\mathcal{I} = \{a, b\}$; $\mathcal{O} = \{0, 1\}$; $\mathcal{S} = \{S, A, B\}$; initial state $= S$

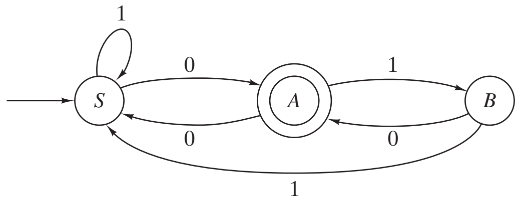
$\mathcal{S}$ \ $\mathcal{I}$	f: a	f: b	g: a	g: b
S	A	A	0	0
A	S	B	1	1
B	A	B	1	0

3. 1101

4.

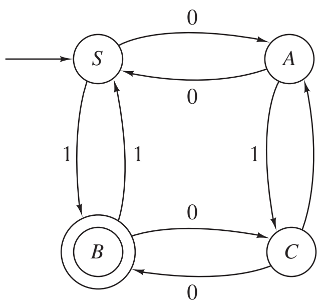


5.



6. Yes

7.



8. Every 0 is followed by a 1.

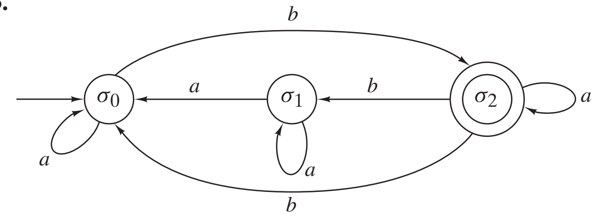
9. Context-free

10. $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaaAbbbb \Rightarrow aaaaAbbbb \Rightarrow aaaabbbb$

11. $a^i b^j$, $j \le 2 + i$, $j \ge 1$, $i \ge 0$

12. $S \to ASB$, $S \to AB$, $AB \to BA$, $BA \to AB$, $A \to a$, $B \to b$

13.



14. $\mathcal{I} = \{a, b\}$; $\mathcal{S} = \{\sigma_0, \sigma_1, \sigma_2\}$; $\mathcal{A} = \{\sigma_0\}$; initial state $= \sigma_0$

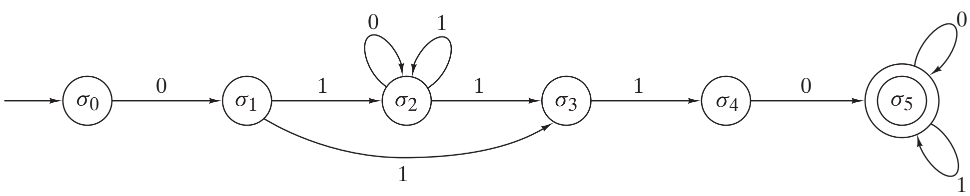
$\mathcal{S}$ \ $\mathcal{I}$	a	b
σ_0	$\{\sigma_0, \sigma_1\}$	$\varnothing$
σ_1	$\varnothing$	$\{\sigma_2\}$
σ_2	$\{\sigma_0, \sigma_2\}$	$\{\sigma_2\}$

15. Yes, since the path

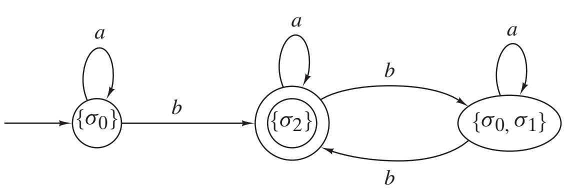
$$(\sigma_0, \sigma_0, \sigma_1, \sigma_2, \sigma_2, \sigma_2, \sigma_2, \sigma_0)$$

represents *aabaaba* and σ_0 is an accepting state.

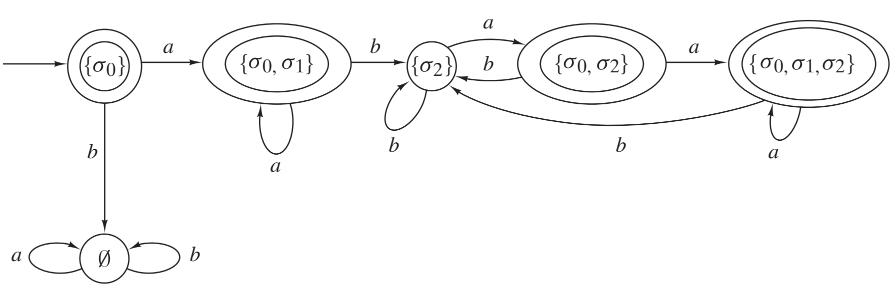
16.



17.



18.



19. Combine the nondeterministic finite-state automata that accept L_1 and L_2 in the following way. Let S be the start state of L_2. For each edge of the form (S_1, S_2) labeled a in L_1 where S_2 is an accepting state, add an edge (S_1, S) labeled a. The start state of the nondeterministic finite-state automaton is the start state of L_1. The accepting states of the nondeterministic finite-state automaton are the accepting states of L_2.

20. Let A' be a nondeterministic finite-state automaton that accepts a regular language that does not contain the null string. Add a state F. For each edge, (σ, σ') labeled a in A' where σ' is accepting, add the edge (σ, F) labeled a. Make F the only accepting state. The resulting nondeterministic finite-state automaton A has one accepting state. We claim that $\text{Ac}(A) = \text{Ac}(A')$.

We show that $\text{Ac}(A) \subseteq \text{Ac}(A')$. [The argument that $\text{Ac}(A') \subseteq \text{Ac}(A)$ is similar and omitted.] Suppose that $\alpha \in \text{Ac}(A)$. There is a path

$$(\sigma_0, \sigma_1, \ldots, \sigma_{n-1}, \sigma_n)$$

that represents α in A, with σ_n an accepting state. Since $\alpha \neq \lambda$, there is a last symbol a in α. Thus the edge (σ_{n-1}, σ_n) is labeled a. Now the path

$$(\sigma_0, \sigma_1, \ldots, \sigma_{n-1}, F)$$

represents α in A' and terminates in an accepting state. Therefore, $\alpha \in \text{Ac}(A')$.

To see that the statement is false for an arbitrary regular language, consider the regular language

$$L = \{\lambda\} \cup \{0^i \mid i \text{ is odd }\}$$

and a nondeterministic finite-state automaton A with start state S that accepts L. Since $\lambda \in L$, S is an accepting state. If S has a loop labeled 0, then A accepts all strings of 0's; therefore, there is no loop at S labeled 0. Since $0 \in L$ and there is no loop at S, there is an edge from S to an accepting state $S' \neq S$, which is a contradiction. Therefore, A has at least two accepting states.

Section 13.1 Review

1. Computational geometry is concerned with the design and analysis of algorithms to solve geometry problems.

2. Given n points in the plane, find a closest pair.

3. Compute the distance between each pair of points and choose the minimum distance.

4. Find a vertical line l that divides the points into two nearly equal parts. Then recursively solve the problem for each of the parts. Let δ_L be the distance between a closest pair in the left part, and let δ_R be the distance between a closest pair in the right part. Let $\delta = \min\{\delta_L, \delta_R\}$. Then examine the points that lie within a vertical strip of width 2δ centered about l. Order the points in this strip in increasing order of the y-coordinates and examine the points in this order. Compute the distance between each point p and the following seven points. Anytime there is a pair whose distance is less than δ, update δ. At the conclusion, δ is the distance between a closest pair.

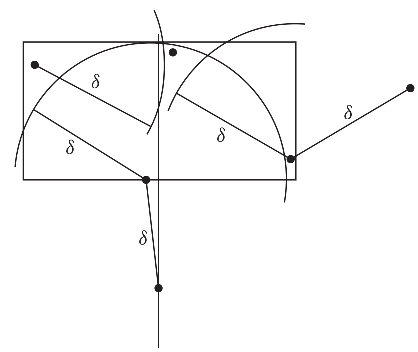
5. The worst-case time of the brute-force algorithm is $\Theta(n^2)$. The worst-case time of the divide-and-conquer algorithm is $\Theta(n \lg n)$.

Section 13.1

1. The 16 points sorted by x-coordinate are $(1,2)$, $(1,5)$, $(1,9)$, $(3,7)$, $(3,11)$, $(5,4)$, $(5,9)$, $(7,6)$, $(8,4)$, $(8,7)$, $(8,9)$, $(11,3)$, $(11,7)$, $(12,10)$, $(14,7)$, $(17,10)$, so the dividing point is $(7,6)$. We next find $\delta_L = \sqrt{8}$, the minimum distance among the left-side points $(1,2)$, $(1,5)$, $(1,9)$, $(3,7)$, $(3,11)$, $(5,4)$, $(5,9)$, $(7,6)$, and $\delta_R = 2$, the minimum distance among the right-side points $(8,4)$, $(8,7)$, $(8,9)$, $(11,3)$, $(11,7)$, $(12,10)$, $(14,7)$, $(17,10)$. Thus $\delta = \min\{\delta_L, \delta_R\} = 2$. The points, sorted by y-coordinate in the vertical strip, are $(8,4)$, $(7,6)$, $(8,7)$, $(8,9)$. In this case we compare each point in the strip to all the following points. The distances from $(8,4)$ to $(7,6)$, $(8,7)$, $(8,9)$ are not less than 2, so δ is not updated at this point. The distance from $(7,6)$ to $(8,7)$ is $\sqrt{2}$, so δ is updated to $\sqrt{2}$. The distances from $(7,6)$ to $(8,9)$ and from $(8,7)$ to $(8,9)$ are greater than $\sqrt{2}$, so δ remains $\sqrt{2}$. Therefore, the distance between the closest pair is $\sqrt{2}$.

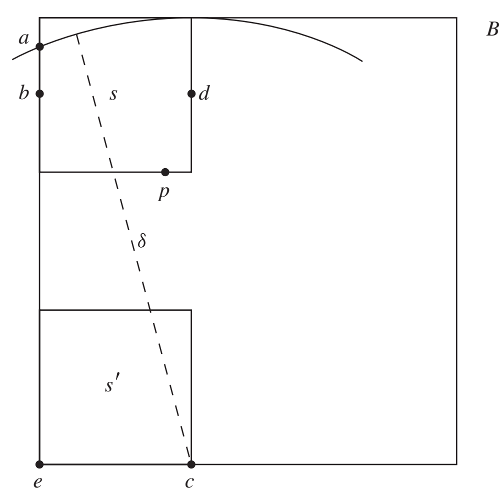
4. Consider the extreme case when all of the points are on the vertical line.

7.



10. Let B be either of the left or right $\delta \times \delta$ squares that make up the $\delta \times 2\delta$ rectangle (see Figure 13.1.2). We argue by contradiction and assume that B contains four or more points. We partition B into four $\delta/2 \times \delta/2$ squares as shown in Figure 13.1.3. Then each of the four squares contains at most one point, and therefore exactly one point. Subsequently we refer to these four squares as the subsquares of B.

The figure



shows the following construction. We reduce the size of the subsquares, if possible, so that

- Each subsquare contains one point.
- The subsquares are the same size.
- The subsquares are as small as possible.

Since at least one point is not in a corner of B, the subsquares do not collapse to points and so at least one point is on a side of a subsquare s interior to B. We choose such a point and call it p. We select a subsquare s' nearest p. We label the two corner points of s' on the side farthest from p, e and c. We draw a circle of radius δ with center at c and let a be the (noncorner) point where this circle meets the side of s. Note that this circle meets a side of s in a noncorner point. Choose a point b in s on the same side as a between a and e. Let d be the corresponding point on the opposite side of s. Now the length of the diameter of rectangle $R = bdce$ is less than δ; hence, R contains at most one point. This is contradiction since R contains p and the point in s'. Therefore, B contains at most three points.

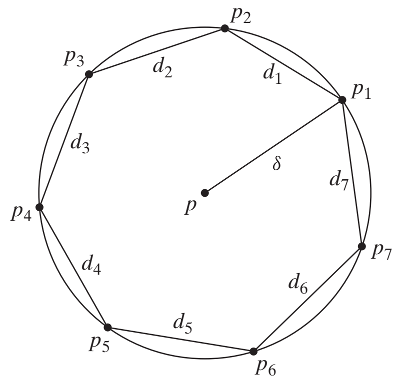
13. In addition to $p.x$ and $p.y$, we assume that each point p has another field $p.side$, which we use to indicate whether p is on the left side or the right side when the points are divided into two nearly equal parts. The extra argument, *label*, to *rec_find_all_2δ_once* sets $p.side$ to *label* for all points p.

```
find_all_2δ_once(p,n) {
   δ = closest_pair(p, n) // original procedure
   if (δ > 0) {
      sort p1, ..., pn by x-coordinate
      rec_find_all_2δ_once(p, 1, n, δ, λ) // λ = empty string
   }
}

rec_find_all_2δ_once(p, i, j, δ, label)
   if (j − i < 3) {
      sort pi, ..., pj by y-coordinate
      directly find and output all distinct pairs less than
         2δ apart
      for k = i to j
         pk.side = label
      return
   }
   k = ⌊(i + j)/2⌋
   l = pk.x
   rec_find_all_2δ_once(p, i, k, δ, L)
   rec_find_all_2δ_once(p, k + 1, j, δ, R)
   merge pi, ..., pk and pk+1, ..., pj by y-coordinate
   t = 0
   for k = i to j
      if (pk.x > l − 2 ∗ δ ∧ pk.x < l + 2 ∗ δ) {
         t = t + 1
         vt = pk
      }
   for k = 1 to t − 1
      for s = k + 1 to min{t, k + 31}
         if (dist(vk, vs) < 2 ∗ δ ∧ vk.side¬ = vs.side)
            println(vk + " " + vs)
            for k = i to j
               pk.side = label
      }
```

16. We show that for each point p, there are at most 6 distinct points whose distance to p is δ. It will then follow that the number of pairs δ apart is less than or equal to $6n$.

We argue by contradiction. Suppose that some point p has 7 distinct neighbors $p_1, \ldots, p_7$ whose distance to p is δ. Then we have the situation



Let C be the circumference of this circle. Since each d_i is at least δ, we have

$$2\pi\delta = C > \sum_{i=1}^{7} d_i \geq 7\delta.$$

Therefore, $\pi > 7/2 = 3.5$, which is a contradiction. (A more careful estimate shows that for each point p, there are at most 5 distinct points whose distance to p is δ.)

Section 13.2 Review

1. Given a finite set of points S in the plane, a point $p \in S$ is a hull point if there exists a line L through p such that all points in S except p lie on one side of L.

2. The convex hull of a finite set of points S in the plane is the sequence $p_1, p_2, \ldots, p_n$ of hull points of S listed in the following order. The point p_1 is the point with minimum y-coordinate. If several points have the same minimum y-coordinate, p_1 is the one with minimum x-coordinate. The remaining points p_i are listed in increasing order of the angle from the horizontal to the line segment p_1, p_i.

3. Let point p_i have coordinates (x_i, y_i). Then the cross product of the points p_0, p_1, p_2 is

$$\text{cross}(p_0, p_1, p_2) = (y_2 - y_0)(x_1 - x_0) - (y_1 - y_0)(x_2 - x_0).$$

4. Graham's Algorithm first finds the point p_1 with minimum y-coordinate. If several points have the same minimum y-coordinate, the point chosen is the one with minimum x-coordinate. It next sorts all of the remaining points p_i on the angle from the horizontal to the line segment p_1, p_i. It then examines successive triples of points. If these points make a left turn, the middle point is retained. If these points make a right turn, the middle point is discarded. At the conclusion of the algorithm, the remaining points are, in order, the convex hull.

5. $\Omega(n \lg n)$

6. Any convex hull algorithm can be used to sort real numbers whose values are between 0 and 1. The points are first projected onto the unit circle (see Figure 13.2.11). Next the convex hull algorithm is used to find the convex hull. Then the y-coordinates of the convex hull (in order) give the sorted order of the original points. Since the worst-case time of any sorting algorithm is $\Omega(n \lg n)$, the worst-case time of any convex hull algorithm must also be $\Omega(n \lg n)$.

Section 13.2

1. Let L be the horizontal line through p_1. By the choice of p_1, no points of S lie below L. If p_1 is the only point of S on L, p_1 is a hull point. If other points of S lie on L, they all lie to the right of p_1 (by the choice of p_1). In this case, if we rotate L clockwise slightly about p_1, L will contain only p_1 and all other points of S will lie above L. Again we conclude that p_1 is a hull point.

4. The points [sorted with respect to $(7, 1)$] are $(7, 1)$, $(10, 1)$, $(16, 4)$, $(12, 3)$, $(14, 5)$, $(16, 10)$, $(13, 8)$, $(10, 5)$, $(10, 9)$, $(10, 13)$, $(7, 7)$, $(7, 13)$, $(6, 10)$, $(3, 13)$, $(4, 8)$, $(1, 8)$, $(4, 4)$, $(2, 2)$. The following table shows each triple that is examined in the while loop, whether it makes a left turn, and the action taken with respect to the triple:

Triple	*Left Turn?*	*Discard Middle Point?*
$(7, 1), (10, 1), (16, 4)$	Yes	No
$(10, 1), (16, 4), (12, 3)$	Yes	No
$(16, 4), (12, 3), (14, 5)$	No	Yes
$(10, 1), (16, 4), (14, 5)$	Yes	No
$(16, 4), (14, 5), (16, 10)$	No	Yes
$(10, 1), (16, 4), (16, 10)$	Yes	No
$(16, 4), (16, 10), (13, 8)$	Yes	No
$(16, 10), (13, 8), (10, 5)$	Yes	No
$(13, 8), (10, 5), (10, 9)$	No	Yes
$(16, 10), (13, 8), (10, 9)$	No	Yes
$(16, 4), (16, 10), (10, 9)$	Yes	No
$(16, 10), (10, 9), (10, 13)$	No	Yes
$(16, 4), (16, 10), (10, 13)$	Yes	No
$(16, 10), (10, 13), (7, 7)$	Yes	No
$(10, 13), (7, 7), (7, 13)$	No	Yes
$(16, 10), (10, 13), (7, 13)$	Yes	No
$(10, 13), (7, 13), (6, 10)$	Yes	No
$(7, 13), (6, 10), (3, 13)$	No	Yes
$(10, 13), (7, 13), (3, 13)$	No	Yes
$(16, 10), (10, 13), (3, 13)$	Yes	No
$(10, 13), (3, 13), (4, 8)$	Yes	No
$(3, 13), (4, 8), (1, 8)$	No	Yes
$(10, 13), (3, 13), (1, 8)$	Yes	No
$(3, 13), (1, 8), (4, 4)$	Yes	No
$(1, 8), (4, 4), (2, 2)$	No	Yes
$(3, 13), (1, 8), (2, 2)$	Yes	No

The convex hull is $(7, 1)$, $(10, 1)$, $(16, 4)$, $(16, 10)$, $(10, 13)$, $(3, 13)$, $(1, 8)$, $(2, 2)$.

7. After finding $p_1, \ldots, p_i$, Jarvis's march finds the point p_{i+1} such that p_{i-1}, p_i, p_{i+1} make the smallest left turn. It follows that if the line L through p_i, p_{i+1} is rotated clockwise slightly about p_i, L will contain only p_i, and all other points of S will lie on one side of L. Thus p_i is a hull point. By construction, Jarvis's march finds all hull points. Thus Jarvis's march does find the convex hull.

10. Yes. Jarvis's march is faster when "most" points are not on the convex hull.

Chapter 13 Self-Test

1. The 18 points sorted by the x-coordinate are $(1, 8)$, $(2, 2)$, $(3, 13)$, $(4, 4)$, $(4, 8)$, $(6, 10)$, $(7, 1)$, $(7, 7)$, $(7, 13)$, $(10, 1)$, $(10, 5)$, $(10, 9)$, $(10, 13)$, $(12, 3)$, $(13, 8)$, $(14, 5)$, $(16, 4)$, $(16, 10)$, so the dividing point is $(7, 13)$. We next find $\delta_L = \sqrt{8}$, the minimum distance among the left-side points $(1, 8)$, $(2, 2)$, $(3, 13)$, $(4, 4)$, $(4, 8)$, $(6, 10)$, $(7, 1)$, $(7, 7)$, $(7, 13)$, and $\delta_R = \sqrt{5}$, the minimum distance among the right-side points $(10, 1)$, $(10, 5)$, $(10, 9)$, $(10, 13)$, $(12, 3)$, $(13, 8)$, $(14, 5)$, $(16, 4)$, $(16, 10)$. Thus $\delta = \min\{\delta_L, \delta_R\} = \sqrt{5}$. The points, sorted by y-coordinate in the vertical strip, are $(7, 1)$, $(7, 7)$, $(6, 10)$, $(7, 13)$. In this case we compare each point in the strip to all the following points. Since no pair is closer than $\sqrt{5}$, the algorithm does not update δ. Therefore, the distance between the closest pair is $\sqrt{5}$.

2. If we replace "three" by "two," when there are three points, the algorithm would be called recursively with inputs of sizes 1 and 2. But a set consisting of one point has no pair—let alone a closest pair.

3. Each $\delta/2 \times \delta/2$ box contains at most one point, so there are at most four points in the lower half of the rectangle.

4. $\Theta(n(\lg n)^2)$

5. Let L be the vertical line through p. By the choice of p, no points of S lie to the right of L. If p is the only point of S on L, p is a hull point. If other points of S lie on L, they all lie below p. In this case, if we rotate L clockwise slightly about p, L will contain only p and all other points of S will be to the left of L. Again we conclude that p is a hull point.

6. Let L be the line segment joining p and q. Let L' be the line through p perpendicular to L. There can be no other point r of S on L' or on the side of L' opposite q, for if there were such a point r, the distance from r to q would exceed the distance from p to q, which is impossible. Thus p is a hull point. Similarly, q is a hull point.

7. The points [sorted with respect to $(1, 2)$] are $(1, 2)$, $(11, 3)$, $(8, 4)$, $(14, 7)$, $(5, 4)$, $(11, 7)$, $(17, 10)$, $(7, 6)$, $(8, 7)$, $(12, 10)$, $(8, 9)$, $(5, 9)$, $(3, 7)$, $(3, 11)$, $(1, 5)$, $(1, 9)$. The following table shows each triple that is examined in the while loop, whether it makes a left turn, and the action taken with respect to the triple:

Triple	*Left Turn?*	*Discard Middle Point?*
(1, 2), (11, 3), (8, 4)	Yes	No
(11, 3), (8, 4), (14, 7)	No	Yes
(1, 2), (11, 3), (14, 7)	Yes	No
(11, 3), (14, 7), (5, 4)	Yes	No
(14, 7), (5, 4), (11, 7)	No	Yes
(11, 3), (14, 7), (11, 7)	Yes	No
(14, 7), (11, 7), (17, 10)	No	Yes
(11, 3), (14, 7), (17, 10)	No	Yes
(1, 2), (11, 3), (17, 10)	Yes	No
(11, 3), (17, 10), (7, 6)	Yes	No
(17, 10), (7, 6), (8, 7)	No	Yes
(11, 3), (17, 10), (8, 7)	Yes	No
(17, 10), (8, 7), (12, 10)	No	Yes
(11, 3), (17, 10), (12, 10)	Yes	No
(17, 10), (12, 10), (8, 9)	Yes	No
(12, 10), (8, 9), (5, 9)	No	Yes
(17, 10), (12, 10), (5, 9)	Yes	No
(12, 10), (5, 9), (3, 7)	Yes	No
(5, 9), (3, 7), (3, 11)	No	Yes
(12, 10), (5, 9), (3, 11)	No	Yes
(17, 10), (12, 10), (3, 11)	No	Yes
(11, 3), (17, 10), (3, 11)	Yes	No
(17, 10), (3, 11), (1, 5)	Yes	No
(3, 11), (1, 5), (1, 9)	No	Yes
(17, 10), (3, 11), (1, 9)	Yes	No

The convex hull is (1, 2), (11, 3), (17, 10), (3, 11), (1, 9).

8. Run the part of Graham's Algorithm that follows the sort on the remaining points.

Appendix A

1. $\begin{pmatrix} 2+a & 4+b & 1+c \\ 6+d & 9+e & 3+f \\ 1+g & -1+h & 6+i \end{pmatrix}$

2. $\begin{pmatrix} 5 & 7 & 7 \\ -7 & 10 & -1 \end{pmatrix}$

5. $\begin{pmatrix} 3 & 18 & 27 \\ 0 & 12 & -6 \end{pmatrix}$

8. $\begin{pmatrix} -2 & -35 & -56 \\ -7 & -18 & 13 \end{pmatrix}$

9. $\begin{pmatrix} 18 & 10 \\ 14 & -6 \\ 23 & 1 \end{pmatrix}$

12. (-4)

14. (a) 2×3, 3×3, 3×2

(b) $AB = \begin{pmatrix} 33 & 18 & 47 \\ 8 & 9 & 43 \end{pmatrix}$

$AC = \begin{pmatrix} 16 & 56 \\ 14 & 63 \end{pmatrix}$

$CA = \begin{pmatrix} 4 & 18 & 38 \\ 0 & 0 & 0 \\ 2 & 17 & 75 \end{pmatrix}$

$AB^2 = \begin{pmatrix} 177 & 215 & 531 \\ 80 & 93 & 323 \end{pmatrix}$

$BC = \begin{pmatrix} 18 & 65 \\ 34 & 25 \\ 12 & 54 \end{pmatrix}$

17. Let $A = (b_{ij})$, $I_n = (a_{jk})$, $AI_n = (c_{ik})$. Then

$$c_{ik} = \sum_{j=1}^{n} b_{ij}a_{jk} = b_{ik}a_{kk} = b_{ik}.$$

Therefore, $AI_n = A$. Similarly, $I_nA = A$.

20. The solution is $X = A^{-1}C$.

Appendix B

1. $-4x$

4. $\dfrac{15x - 3b}{3} = 5x - b$

7. $\dfrac{1}{n} - \dfrac{1}{n+1} = \dfrac{n+1-n}{n(n+1)} = \dfrac{1}{n(n+1)}$

We may use this equation to compute $\sum_{i=1}^{n} \frac{1}{i(i+1)}$ as follows:

$$\begin{aligned}
&\sum_{i=1}^{n} \frac{1}{i(i+1)} \\
&\quad = \sum_{i=1}^{n} \frac{1}{i} - \frac{1}{i+1} \\
&\quad = \left(1 - \frac{1}{2}\right) + \left(\frac{1}{2} - \frac{1}{3}\right) + \cdots + \left(\frac{1}{n-1} - \frac{1}{n}\right) \\
&\qquad + \left(\frac{1}{n} - \frac{1}{n+1}\right) \\
&\quad = 1 - \frac{1}{n+1} = \frac{n+1-1}{n+1} = \frac{n}{n+1}.
\end{aligned}$$

8. 81

11. 1/81

14. (a), (c), and (g) are equal. (b) and (f) are equal. (d) and (e) are equal.

16. $x^2 + 8x + 15$

19. $x^2 + 8x + 16$

22. $x^2 - 4$

25. $(x + 5)(x + 1)$

28. $(x - 4)^2$

31. $(2x + 1)(x + 5)$

34. $(2x + 3)(2x - 3)$

37. $(n+1)! + (n+1)(n+1)! = (n+1)![1+(n+1)] = (n+1)!(n+2) = (n+2)!$

40.

$$\begin{aligned}&7(3\cdot 2^{n-1} - 4\cdot 5^{n-1}) - 10(3\cdot 2^{n-2} - 4\cdot 5^{n-2})\\ &\quad = 2^{n-2}(7\cdot 3\cdot 2 - 10\cdot 3) + 5^{n-2}(-7\cdot 4\cdot 5 + 10\cdot 4)\\ &\quad = 2^{n-2}\cdot 12 + 5^{n-2}(-100)\\ &\quad = 2^{n-2}(2^2\cdot 3) - 5^{n-2}(5^2\cdot 4)\\ &\quad = 3\cdot 2^n - 4\cdot 5^n\end{aligned}$$

42. Factoring gives $(x-4)(x-2) = 0$, which has solutions $x = 4, 2$.

45. $2x \le 6$, $x \le 3$

48. $i \le n$ for $i = 1, \dots, n$. Summing these inequalities, we obtain

$$\sum_{i=1}^{n} i \le n\cdot n = n^2.$$

51. Multiply by $(n+2)n^2(n+1)^2$ to get

$$(2n+1)(n+1)^2 > 2(n+2)n^2$$

or

$$2n^3 + 5n^2 + 4n + 1 > 2n^3 + 4n^2$$

or

$$n^2 + 4n + 1 > 0,$$

which is true if $n \ge 1$.

54. 6

57. 10

59. 2.584962501

62. −0.736965594

64. 2.392231208

67. 0.480415248

68. 1.489896102

71. Let $u = \log_b y$ and $v = \log_b x$. By definition, $b^u = y$ and $b^v = x$. Now

$$x^{\log_b y} = x^u = (b^v)^u = b^{vu} = (b^u)^v = y^v = y^{\log_b x}.$$

Appendix C

1. First *large* is set to 2 and i is set to 2. Since $i \le n$ is true, the body of the while loop executes. Since $s_i > large$ is true, *large* is set to 3. i is set to 3 and the while loop executes again.

Since $i \le n$ is true, the body of the while loop executes. Since $s_i > large$ is true, *large* is set to 8. i is set to 4 and the while loop executes again.

Since $i \le n$ is true, the body of the while loop executes. Since $s_i > large$ is false, the value of *large* does not change. i is set to 5 and the while loop executes again.

Since $i \le n$ is false, the while loop terminates. The value of *large* is 8, the largest element in the sequence.

4. First x is set to 4. Since $b > x$ is false, $x = b$ is *not* executed. Since $c > x$ is true, $x = c$ executes, and x is set to 5. Thus x is the largest of the numbers a, b, and c.

7.

```
min(a, b) {
  if (a < b)
    return a
  else
    return b
}
```

10.

```
odds(n) {
  i = 1
  while (i ≤ n) {
    println(i)
    i = i + 2
  }
}
```

13.

```
product(s, n) {
  partial_product = 1
  for i = 1 to n
    partial_product = partial_product * s_i
  return partial_product
}
```