

Chapter 4

Quantitative Thinking

Counting is important. Many problems in mathematics, computer science, and other technical fields involve counting the elements of some set of objects. But counting isn't always easy. In this chapter we will investigate tools for counting



Figure 4.1 A typical position in chess presents the players with several different possible moves. In order to look two or three moves ahead, players must consider hundreds of combinations, and the number of distinct 40-move games seems almost limitless. Enumerating these possibilities, even approximately, reveals the complex nature of the game.

204 Chapter 4 ■ Quantitative Thinking

certain types of sets, and we will learn how to think about problems from a quantitative point of view.

The goal of this chapter is to see how quantitative thinking is useful for analyzing discrete problems, especially in computer science. A course in combinatorics will teach you more about specific counting techniques; our aim here is to learn a few of these techniques, but also to see why these techniques are important in the study of discrete processes.

4.1 Basic Counting Techniques

Most counting problems can be reduced to adding and multiplying. This sounds easy, but the hard part is knowing when to add and when to multiply. We'll start with some very simple examples.

4.1.1 Addition

In Section 2.2, we introduced the inclusion–exclusion principle. It states that if A and B are finite sets, then the size of their union is given by

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

We say that A and B are *disjoint* if $A \cap B = \emptyset$. In this case, the inclusion–exclusion principle reduces to the equation $|A \cup B| = |A| + |B|$. In other words, if two sets have no elements in common, then you count the total number of elements in both sets by counting the elements in each and adding. This simple observation gives us our first counting principle.

Addition Principle. Suppose A and B are finite sets with $A \cap B = \emptyset$. Then there are $|A| + |B|$ ways to choose an element from $A \cup B$.

In counting problems, disjoint sets usually take the form of mutually exclusive options or cases. If a person has an “either/or” choice, or a problem reduces to separate cases, the addition principle is probably called for.

Example 4.1 Ray owns five bicycles and three cars. He can get to work using any one of these vehicles. How many different ways can he get to work?

Solution: Since it is impossible to take *both* a car and a bicycle to work, these are disjoint sets. Thus Ray has $5 + 3 = 8$ choices. \diamond

Example 4.2 On a certain day, a restaurant served 25 people breakfast and later served 37 people lunch. How many different customers did they have in total?

Solution: There isn't enough information to answer this question as stated. Before we can accurately count the total number of customers, we need to know whether any of the breakfast customers returned for lunch. If none did, then the breakfast customers are disjoint from the lunch customers, so the total is $25 + 37 = 62$. But suppose there were four breakfast customers who returned for lunch. Then by the inclusion–exclusion principle, there were only $25 + 37 - 4 = 58$ customers total. Another way to look at this situation is by thinking of the customers as belonging to three disjoint sets: those who ate only breakfast, those who ate only lunch, and those who ate both. The sizes of these three sets are 21, 33, and 4, respectively, so by the addition principle the total is $21 + 33 + 4 = 58$. \diamond

Strictly speaking, this last bit of reasoning used a slightly stronger version of the addition principle, which we will state as a theorem.

Theorem 4.1 *Suppose that $A_1, A_2, A_3, \dots, A_n$ are pairwise disjoint finite sets, that is, $A_i \cap A_j = \emptyset$ for all i and j with $i \neq j$. Then*

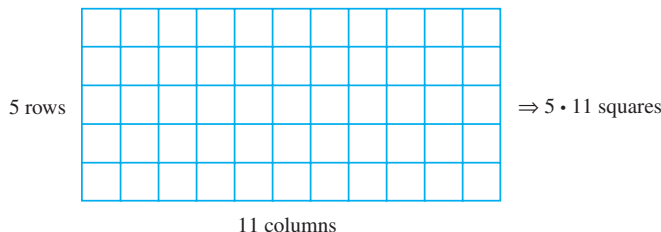
$$|A_1 \cup A_2 \cup A_3 \cup \dots \cup A_n| = |A_1| + |A_2| + |A_3| + \dots + |A_n|.$$

Proof Exercise. Use induction on n . \square

In other words, no matter how many disjoint cases you have, you can account for them all by adding up the counts of each.

4.1.2 Multiplication

Counting the elements in a rectangular grid is easy: you multiply the number of rows by the number of columns.



We can always think of a Cartesian product $A \times B$ of two finite sets A and B as a grid, where the columns are indexed by A and the rows are indexed by B . We state this observation as another principle.

Multiplication Principle. Let A and B be finite sets. The number of elements (i.e., ordered pairs) in $A \times B$ is $|A| \cdot |B|$. So there are $|A| \cdot |B|$

ways to choose two items in sequence, with the first item coming from A and the second item from B .

Example 4.3 Ray owns five bicycles and three cars. He plans to ride a bicycle to and from work, and then take one of his cars to go to a restaurant for dinner. How many different ways can he do this?

Solution: Ray is making two choices in sequence, so he is forming an ordered pair of the form (bicycle, car). Thus there are $5 \cdot 3 = 15$ ways possible. \diamond

The decision process in this last example has a nice graphical model in the form of a tree (Figure 4.2). Let the root of the tree represent Ray's situation before he has made any decisions. The nodes of depth 1 correspond to the five different bicycles Ray can choose for his commute to work (mountain, road, recumbent, tandem, and electric), and the nodes of depth 2 represent the choice of car for dinner transportation (Ford, BMW, GM). Each path from the root to a leaf represents a choice of an ordered sequence of the form (bicycle, car), so the number of paths through this tree (i.e., the number of leaves) equals the number of different sequences Ray can choose. Such a model is called a *decision tree*.

Like the addition principle, the multiplication principle generalizes to collections of more than two sets.

Theorem 4.2 Suppose that $A_1, A_2, A_3, \dots, A_n$ are finite sets. Then

$$|A_1 \times A_2 \times A_3 \times \cdots \times A_n| = |A_1| \cdot |A_2| \cdot |A_3| \cdots |A_n|.$$

Proof Exercise. Use induction on n . \square

Notice that unlike the addition principle, the sets in the multiplication principle need not be disjoint. The next example uses the fact that $|A \times A \times A| = |A| \cdot |A| \cdot |A|$.

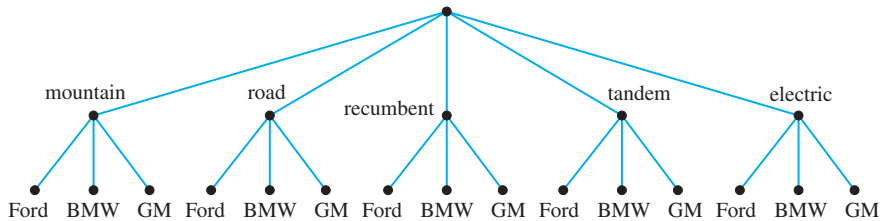


Figure 4.2 A decision tree for the counting problem in Example 4.3.

Example 4.4 How many strings of length 3 can be formed from a 26-symbol alphabet?

Solution: There are three choices to be made in sequence: the first letter, the second letter, and the third letter. We have 26 options for each choice. Therefore, the total number of length 3 strings is $26 \cdot 26 \cdot 26 = 26^3 = 17,576$. \diamond

Example 4.5 How many different binary strings of length 24 are there?

Solution: There are two choices for each digit: 0 or 1. Choosing a string of length 24 involves making this choice 24 times, in sequence. Thus, the number of possibilities is

$$\underbrace{2 \cdot 2 \cdots 2}_{24 \text{ two's}} = 2^{24} = 16,777,216.$$

In computer graphics, color values are often represented by such a string. This type of color is known variously as “true color,” “24-bit color,” or “millions of colors,” reflecting the number of possible color choices. \diamond

The last two examples don’t lend themselves well to decision trees; the trees would be much too large to draw. But we don’t really need decision trees for such simple problems; it is easy to see how to apply Theorem 4.2 directly. However, for problems involving separate choices, in which later choices are restricted by earlier ones, decision trees are quite useful.

Example 4.6 How many designs of the form



are possible, if each square must be either red, green, or blue, and no two adjacent squares may be the same color?

Solution: If there were no restrictions on adjacent squares, then this problem would be just like forming a three-letter string from a three-letter alphabet (R, G, B), so the number of designs would be 3^3 , reasoning as in Example 4.4. But this solution counts designs like RRG, which has two adjacent squares colored red, so it overcounts the correct number of designs. One way to avoid this error is to use the decision tree in Figure 4.3. This tree shows that there are only $3 \cdot 2 \cdot 2 = 12$ designs that conform to the given restrictions. \diamond

The decision tree for Example 4.6 helps us see how to apply the multiplication principle. Think of the process of making a design as a sequence of three decisions. You can make the first square any color you wish: R, G, or B. But after this decision is made, the second square must be one of the other two

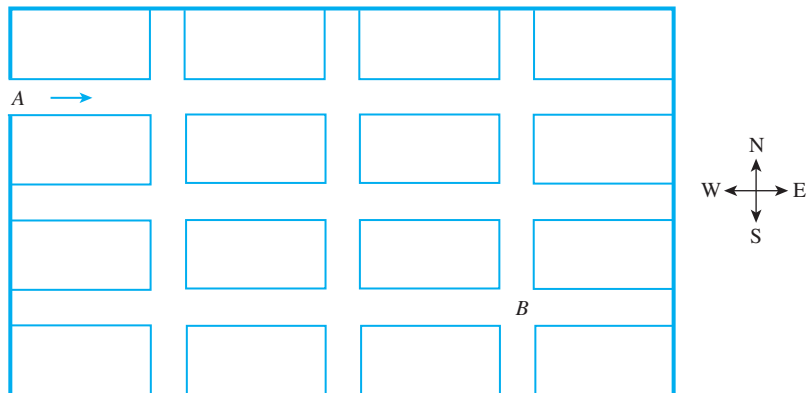


Figure 4.4 A shopping district. See Example 4.8.

a direct path to B , there is no corresponding branch in the tree. Observe that there are six possible direct paths from A to B . \diamond

4.1.3 Mixing Addition and Multiplication

At face value, the addition and multiplication principles are pretty simple. Things get a little more tricky when a counting problem calls for a mixture of the two principles. The next few examples follow the same basic recipe: when a problem breaks up into disjoint cases, use multiplication to count each individual case, and then use addition to tally up the separate cases.

Example 4.9 How many (nonempty) strings of at most length 3 can be formed from a 26-symbol alphabet?

Solution: Reasoning as in Example 4.4, we see that there are 26 strings of length 1, 26^2 of length 2, and 26^3 of length 3. Since these cases are mutually exclusive, the total number of strings is $26 + 26^2 + 26^3 = 18,278$. \diamond

Example 4.10 Illinois license plates used to consist of either three letters followed by three digits or two letters followed by four digits. How many such plates are possible?

Solution: The two types of license plates can be thought of as two disjoint sets; the cases are mutually exclusive. The first case involves a choice of three letters (26^3) followed by a choice of three digits (10^3). For the second case, we

210 Chapter 4 ■ Quantitative Thinking

first choose two letters (26^2) and then choose four digits (10^4). Putting these together, we have a total of

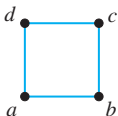
$$26^3 \cdot 10^3 + 26^2 \cdot 10^4 = 24,336,000$$

different possible license plates. \diamond

Example 4.10 is somewhat prototypical: it often helps to recognize certain counting problems as “license plate” problems. A license plate problem involves successive independent choices (multiplication), possibly divided up into disjoint cases (addition). All of the above examples could be thought of as license plate problems (although a system of 24-bit binary license plates, for example, would be a little strange).

Sometimes it is easy to see how to break a counting problem into separate cases, but often it is not so obvious. In the next example, the two cases become apparent only after attempting to count all possibilities as a single case.

Example 4.11 Using the four colors red, green, blue, and violet, how many different ways are there to color the vertices of the graph



so that no two adjacent vertices have the same color?

Solution: Notice the similarity to Example 4.6; we’ll start by attempting a similar solution. Color one vertex, say a , one of the three colors, R, G, B, or V. Now there are three possible colors for each of the adjacent vertices b and d , so we have $4 \cdot 3 \cdot 3$ ways to color these three vertices. We must now count the ways to color vertex c . But here we get stuck, for if b and d are the same color, then we have three choices for c , but if the colors of b and d are different, we are left with only two choices for c . Therefore we should consider two disjoint cases:

- Case 1.** Suppose b and d are different colors. Then, as above, we have four choices for a , three choices for b , and then two choices for d , since it must differ from both b and a . We are left with only two choices for c , for a total of $4 \cdot 3 \cdot 2 \cdot 2 = 48$ different colorings.
- Case 2.** Suppose b and d are colored the same color. Then we have four choices for a , and then three choices for the color that b and d share. There are then three choices for c , for a total of $4 \cdot 3 \cdot 3 = 36$ ways to color this case.

By the addition principle, the total number of colorings is $48 + 36 = 84$. \diamond

Exercises 4.1

1. Professor N. Timmy Date has 30 students in his Calculus class and 24 students in his Discrete Mathematics class.
 - (a) Assuming that there are no students who take both classes, how many students does Professor Date have?
 - (b) Assuming that there are eight students who take both classes, how many students does Professor Date have?
2. A restaurant offers two different kinds of soup and five different kinds of salad.
 - (a) If you can have either soup or salad, how many choices do you have?
 - (b) If you can have both soup and salad, how many choices do you have?
3. There are 18 major sea islands in the Queen Elizabeth Islands of Canada. There are 15 major lakes in Saskatchewan, Canada.
 - (a) If you are planning a trip to visit one of these islands, followed by one of these lakes, how many different trips could you make?
 - (b) If you plan to visit either one of these lakes or one of these islands, how many different visits could you make?
4. Bill has three one-piece jump suits, five pairs of work pants, and eight work shirts. He either wears a jump suit, or pants and a shirt to work. How many different possible outfits does he have?
5. A new car is offered with ten different optional packages. The dealer claims that there are “more than 1,000 different combinations” available. Is this claim justified? Explain.
6. License plates in India begin with a code that identifies the state and district where the vehicle is registered, and this code is followed by a four-digit identification number. These identification numbers are given sequentially, starting with 0000, 0001, 0002, etc. Once this sequence reaches 9999, a letter from the set $\{A, \dots, Z\}$ is added (in order), and once these run out, additional letters are added, and so on. So the sequence of identification numbers proceeds as follows: 0000, 0001, \dots , 9999, A0000, A0001, \dots , A9999, B0000, B0001, \dots , B9999, \dots , Z0000, Z0001, \dots , Z9999, AA0000, AA0001, \dots
 - (a) How many identification numbers are there using two or fewer letters?

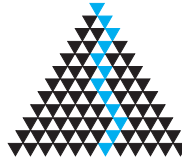
212 Chapter 4 ■ Quantitative Thinking

- (b) If a district registers 10 million cars, how many identification numbers must have three letters?
 - (c) Suppose a district registers 500,000 cars. What percentage of identification numbers have exactly one letter?
 - (d) Suppose a district registers 500,000 cars. What percentage of identification numbers have no letters?
 - (e) Suppose you see a plate in Bangalore, India, with the identification number CR7812. How many vehicles were registered in this district before the vehicle with this plate?
7. License plates in China begin with a Chinese character designating the province, followed by a letter from the set $\{A, \dots, Z\}$, followed by a five-character alphanumeric string (using symbols from the set $\{A, \dots, Z, 0, 1, \dots, 9\}$). What is the maximum number of plates of this type for a given Chinese province?
8. The protein-coding strand of the average human gene consists of 1350 nucleotides. Assuming that each nucleotide can take any of four values (A, T, C, or G), how many different genes with exactly 1350 nucleotides are possible?
9. Refer to the previous problem. Assuming that gene strands can have between 1200 and 1500 nucleotides, write an expression for the number of possible genes. (Don't bother trying to evaluate this expression!)
10. How many numbers between 1 and 999 (inclusive) are divisible by either 2 or 5?
11. The following problems refer to strings in A, B, \dots , Z.
- (a) How many different four-letter strings are there?
 - (b) How many four-letter strings are there that begin with X?
 - (c) How many four-letter strings are there that contain exactly two X's? (Hint: Consider the disjoint cases determined by where the X's are in the string.)
12. There is often more than one way to do a counting problem, and finding an alternate solution is a good way to check answers. Redo Example 4.11 by considering three disjoint cases: using two different colors, using three different colors, and using four different colors.
13. There are 16 soccer teams in Thailand's Premier League, and there are 22 teams in England's Premier League.
- (a) How many different ways are there to pair up a team from Thailand with a team from England?

- (b) How many different ways are there to pair up two teams from Thailand? (Careful: Pairing Bangkok Bank FC with Chonburi FC is the same as pairing Chonburi FC with Bangkok Bank FC.)
14. Use a decision tree to count the number of strings of length 4 using the symbols **a**, **t**, and **e** with the restriction that **et**, **at**, and **ta** do not appear anywhere in the string.
15. Using the nouns $N = \{\text{dog, man, mouse, bird}\}$ and the verbs $V = \{\text{bites, eats, kicks}\}$, how many “sentences” of the form

$$\langle \text{noun} \rangle \quad \langle \text{verb} \rangle \quad \langle \text{noun} \rangle$$

- are there, with the restriction that every word in the sentence has a different length? (For example, “**dog eats mouse**” is such a sentence, but “**man bites dog**” is not, because it contains two words of length three.) Use a decision tree to arrive at your answer.
16. How many four-digit binary strings are there that do not contain 000 or 111? (Use a decision tree.)
17. Find an alternate solution to Exercise 16. (Count the number of strings that contain 000 or 111 and subtract from the total number of four-digit binary strings.)
18. Let X be a set containing 20 elements. Use the multiplication principle to compute $|\mathcal{P}(X)|$, the size of the power set of X . (Hint: To choose a subset of X , you must choose whether or not to include each element of X .)
19. The *Museo de la Matemática* in Querétaro, Mexico, contains an exhibit with the following figure.



- How many ways are there to choose a sequence of triangles starting at the top triangle and proceeding down to the bottom row, such that the sequence always proceeds down to an adjacent triangle? (One such path is indicated by the blue triangles.)
20. Consider the map in Figure 4.5. Omedi wants to get from point A to some point on the subway (represented by the thick dotted line). At each intersection, he can decide to go either south or east. How many different paths can he take? Draw a decision tree representing the different possible paths.

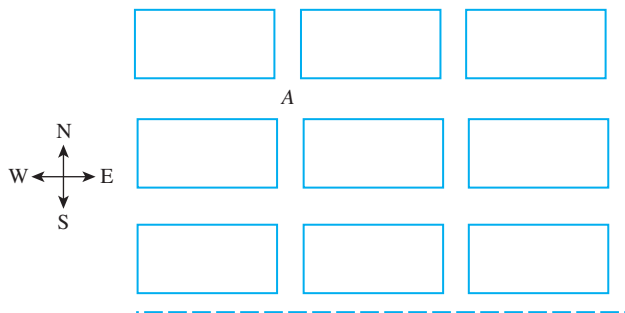


Figure 4.5 Street map for Exercise 20.

21. Two teams (A and B) play a best-of-five match. The match ends when one team wins three games. How many different win or loss scenarios are possible? (Use a decision tree.)
22. Prove Theorem 4.1.
23. Prove Theorem 4.2.

4.2 Selections and Arrangements

So far, we have seen how to enumerate sets using addition and multiplication. These basic principles apply to almost any counting problem in discrete mathematics, but there are many more counting techniques that we could study. While one could easily spend a semester learning these techniques, the next two sections focus on a few of the most important ideas for solving quantitative problems.

In this section we will concentrate on two tasks: selecting and arranging. A selection problem involves choosing a subset of elements from a given set. An arrangement problem involves choosing a subset, and then putting its elements into some particular order. When you are able to think of a counting problem in terms of selections or arrangements, the solution is often easy to see.

4.2.1 Permutations: The Arrangement Principle

Here is an example of an arrangement problem:

Example 4.12 Yifan has 26 refrigerator magnets in the shapes of the letters from A to Z . How many different three-letter strings can he form with these?

Solution: This is a “license plate” problem (see Example 4.10), but with a restriction. Since Yifan has only one magnet per letter, he is not allowed to repeat letters. There are three slots to fill. He has 26 choices for the first slot. Since he can’t reuse that letter, he has 25 choices for the second slot, and similarly, 24 for the third. So the total number of possible strings is $26 \cdot 25 \cdot 24$. \diamond

This solution applies the multiplication principle, but at each successive decision, the number of letters is reduced by one. The arrangement principle states the general rule.

Arrangement Principle. The number of ways to form an ordered list of r distinct elements drawn from a set of n elements is

$$P(n, r) = n \cdot (n - 1) \cdot (n - 2) \cdots (n - r + 1).$$

Such a list is called an *arrangement*. Note that arrangements have two key properties: the order of the elements matters, and all the elements are distinct.

The notation $P(n, r)$ comes from the mathematical term for arrangements: *permutations*. Notice that

$$P(n, r) = \frac{n!}{(n - r)!}$$

is a convenient way of expressing the number of permutations in terms of the factorial function. Recall that the factorial function is defined as

$$n! = n \cdot (n - 1) \cdots 3 \cdot 2 \cdot 1,$$

and, by convention, $0! = 1$. Note that $P(n, n) = n!$.

Example 4.13 A baseball team has a 24-man roster. How many different ways are there to choose a 9-man batting order?

Solution: A batting order is simply a list of nine players in order, so there are $P(24, 9) = 24!/15! = 474,467,051,520 \approx 4.74 \times 10^{11}$ ways to make such a list. \diamond

Example 4.14 How many different ways are there to rearrange the letters in the word GOURMAND?

Solution: The important thing to notice about the word GOURMAND is that all the letters are different. Hence they form a set of eight letters, and rearranging the letters amounts to choosing an ordered list of eight distinct elements from this set. The number of ways to do this is $P(8, 8) = 8! = 40,320$. \diamond

216 Chapter 4 ■ Quantitative Thinking

Rearranging the letters in the word GOURMAND is the same as finding a one-to-one correspondence:

$$f: \{G, O, U, R, M, A, N, D\} \longrightarrow \{G, O, U, R, M, A, N, D\}.$$

For any letter l in GOURMAND, $f(l)$ is the letter that replaces it in the rearrangement. In general, if X is a finite set with n elements, then the number of one-to-one correspondences $f: X \longrightarrow X$ is $n!$. Such a function is called a *permutation* of the set X .

Example 4.15 A kitchen drawer contains ten different plastic food containers and ten different lids, but any lid will fit on any container. How many different ways are there to pair up containers with lids?

Solution: The key to solving this problem is thinking about it the right way. In order to pair up each container with a lid, start by lining all the containers up in a row. (It doesn't matter how you line them up.) Now choose an arrangement of the ten lids, and place the lids in this arrangement next to the containers. This determines a pairing, and all pairings are determined this way. The only choice was in the arrangement of the lids, so there are $P(10, 10) = 10! = 3,628,800$ ways to match up lids with containers. \diamond

The next example highlights the difference between the multiplication principle and the arrangement principle.

Example 4.16 An urn contains 10 ping-pong balls, numbered 1 through 10. Four balls are drawn from the urn in sequence, and the numbers on the balls are recorded. How many ways are there to do this, if

- (a) the balls are replaced before the next one is drawn.
- (b) the balls are drawn and not replaced.

Solution: In case (a), there are always 10 balls in the urn, so there are always 10 choices. By the multiplication principle, the number of ways to draw four balls is $10^4 = 10,000$. In case (b), the balls are not replaced, so the number of choices goes down by one each time a ball is drawn. Hence there are $P(10, 4) = 10 \cdot 9 \cdot 8 \cdot 7 = 5,040$ ways to draw four balls. \diamond

Example 4.16 belongs to the arcane genre of “urn problems.” Although we don't often encounter urns in real life, this type of problem is somewhat prototypical. Like license plate problems, urn problems provide a simple way to classify a certain type of enumeration tasks. And one often hears the terms “with replacement” and “without replacement” associated to arrangements or selections; this terminology makes sense in the context of urns.

4.2.2 Combinations: The Selection Principle

Here is a slight variation on Example 4.13.

Example 4.17 A baseball team has a 24-man roster. How many different ways are there to choose a group of nine players to start the game?

Solution: The only difference between this problem and Example 4.13 is that no order is imposed on the group of starters. Observe that any choice of nine starters accounts for exactly $P(9, 9) = 9!$ batting orders. Thus the number of batting orders is $9!$ times as big as the number of choices for a group of starters. Hence, the number of ways to choose this group is

$$\frac{P(24, 9)}{9!} = \frac{24!}{15!9!} = 1,307,504.$$



The distinction between Examples 4.13 and 4.17 is important: in the latter, the group was an unordered set. This type of choice is called a *selection*.

Selection Principle. The number of ways to choose a subset of r elements from a set of n elements is

$$C(n, r) = \frac{n!}{r!(n-r)!}.$$

Since selections involve choosing a subset, we read the expression “ $C(n, r)$ ” as “ n choose r .” Sometimes we use the notation

$$C(n, r) = \binom{n}{r}.$$

Note that $C(n, n) = 1$, because there is only one subset containing all the elements: the whole set. Similarly, $C(n, 0) = 1$, because the empty set is the only subset with zero elements.

Compare the formula for $C(n, r)$ to the formula for $P(n, r)$. In the selection principle, the $r!$ factor in the denominator accounts for the fact that no order is imposed on the elements of the subset. Arrangements and selections both involve choosing a subset of some set. The key distinction bears repeating: *In arrangements, the order of the elements in the subset matters; in selections, it doesn't.*

Example 4.18 As in Example 4.16, suppose an urn contains 10 ping-pong balls numbered 1 through 10. Instead of drawing four balls in sequence, reach in and grab a handful of four balls. How many different handfuls can you grab?

218 Chapter 4 ■ Quantitative Thinking

Solution: A “handful” of four ping-pong balls is an unordered set, so there are $C(10, 4) = 210$ different possible outcomes. ◇

Take a moment to compare parts (a) and (b) of Example 4.16 with Example 4.18. The sizes of an ordered sequence with replacement, an ordered sequence without replacement, and an unordered set (without replacement) are 10,000, 5040, and 210, respectively.

Example 4.19 How many different ways are there to rearrange the letters in the word PFFPPPPFFFF?

Solution: Although this example resembles Example 4.14, the solution is quite different because the letters of PFFPPPPFFFF are not all distinct. In fact, there are only two letters, P and F, and we must form a ten-letter word using four P’s and six F’s. In order to view this as a selection problem, notice that we have to fill ten blanks

— — — — — — — — — —

using four P’s and six F’s. Once we choose where the P’s go, there are no more choices to make, since the remaining blanks get filled up with the F’s. The order of the blanks we choose doesn’t matter, because we are putting P’s in all of them. So the number of ways to fill in the blanks is $C(10, 6) = 210$. ◇

You may have noticed that we could also have solved this problem by choosing where the F’s go, and then our answer would be $C(10, 4)$. Fortunately, $C(10, 4) = 210$ as well. In fact,

$$C(n, k) = C(n, n - k)$$

for all n and k with $n \geq k \geq 0$. The proof of this identity is left as an exercise.

Rearranging letters in a word may seem like a pointless diversion, but there are a variety of counting problems that are equivalent to Example 4.19. Two examples follow.

Example 4.20 In Example 4.8, we used a decision tree to count all the direct paths along a street grid traveling two blocks east and two blocks south. We could restate this problem as follows: how many different four-symbol strings are there using two E’s and two S’s? By the method of Example 4.19, there are $C(4, 2) = 6$ strings of this form.

Example 4.21 How many solutions are there to the equation

$$x_1 + x_2 + x_3 + x_4 + x_5 = 13,$$

if x_1, \dots, x_5 must be non-negative integers?

Solution: Such a solution corresponds to a distribution of 13 units among the five variables x_1, \dots, x_5 . For example, the solution

$$x_1 = 4, \quad x_2 = 0, \quad x_3 = 5, \quad x_4 = 1, \quad x_5 = 3$$

amounts to dividing up thirteen 1's into groups as follows:

$$1\ 1\ 1\ 1\ ||\ 1\ 1\ 1\ 1\ 1\ 1\ | 1\ | 1\ 1\ 1.$$

We can view this division into groups as a string containing four |'s and thirteen 1's; every such string defines a different solution to the equation, and all solutions can be represented this way. So we just need to count the strings of this type. By the method of Example 4.19, the number of possible strings (and also the number of possible solutions) is $C(17, 4) = 2380$. \diamond

The addition, multiplication, arrangement, and selection principles are powerful enough to solve most counting problems that arise in discrete mathematics. But this is easier said than done: it often takes quite a bit of skill to put these four principles together.

Example 4.22 Two teams, A and B , play a best-of-seven match. The match ends when one team wins four games. How many different win or loss scenarios are possible?

Solution: (Version #1.) The match could go four, five, six, or seven games, and these cases are all disjoint. There are only two ways the winners of a four-game match could go: $AAAA$ or $BBBB$. In a five-game match, the winning team must lose one of the first four games, so there are $2 \cdot C(4, 1) = 8$ ways this can happen; the $C(4, 1)$ factor accounts for choosing which game to lose, and the factor of 2 accounts for either A or B winning the match. Similarly, there are $2 \cdot C(5, 2) = 20$ scenarios for a six-game match, and $2 \cdot C(6, 3) = 40$ scenarios for a seven game match. By the addition principle, there are $2 + 8 + 20 + 40 = 70$ different possible win or loss scenarios. \diamond

This last solution is a nice illustration of using the addition principle in tandem with the selection principle. However, there is an alternate solution that is possibly simpler to understand.

Solution: (Version #2.) Regard every match as lasting seven games: once one team has won four games, that team forfeits the remaining games. This is the same as ending the match after four wins by one team, so the total number of win or loss scenarios should be the same. We must then count the number of seven-symbol strings using four A's and three B's (when A wins the match) and the number of seven-symbol strings using four B's and three A's (when B wins). This is just like Example 4.19; in each case there are $C(7, 4) = 35$ such strings, for a total of 70 win or loss scenarios. \diamond

A third way of solving Example 4.22 would be to use a decision tree (though such a tree would be quite large). It is always a good idea to look for alternate solutions to a counting problem; it's a way of checking your answer.

4.2.3 The Binomial Theorem ‡

In high school algebra, you learned how to expand expressions like $(3x - 5)^4$ by multiplying polynomials:

$$\begin{aligned}(3x - 5)^4 &= (3x - 5)(3x - 5)(3x - 5)(3x - 5) \\ &= (9x^2 - 15x - 15x + 25)(3x - 5)(3x - 5) \\ &= (27x^3 - 45x^2 - 45x^2 + 75x - 45x^2 + 75x + 75x - 125)(3x - 5) \\ &= (27x^3 - 135x^2 + 225x - 125)(3x - 5) \\ &= 81x^4 - 405x^3 + 675x^2 - 375x - 135x^3 + 675x^2 - 1125x + 625 \\ &= 81x^4 - 540x^3 + 1350x^2 - 1500x + 625.\end{aligned}$$

After working through several problems like this one, patterns become apparent. You probably remember that $(a + b)^2 = a^2 + 2ab + b^2$, and you might even remember the formula for $(a + b)^3$.

$$(a + b)^3 = a^3 + 3a^2b + 3ab^2 + b^3.$$

There is a general pattern for the expansion of $(a + b)^n$ that is worth knowing.

Theorem 4.3 *Let j and k be non-negative integers such that $j + k = n$. The coefficient of the $a^j b^k$ term in the expansion of $(a + b)^n$ is $C(n, j)$.*

Proof We use induction on n . If $n = 1$, we have $(a + b)^1 = a + b$, so the coefficient of the $a^0 b^1$ term is $C(1, 0) = 1$ and the coefficient of the $a^1 b^0$ term is $C(1, 1) = 1$.

Suppose as inductive hypothesis that the coefficient of the $a^{j'} b^{k'}$ term in the expansion of $(a + b)^{n-1}$ is $C(n-1, j')$, for any j' and k' such that $j' + k' = n-1$. Now let $j + k = n$. Now apply the inductive hypothesis to evaluate the expansion of $(a + b)^n$. In the calculation below, we only need to keep track of the terms that are capable of contributing to the $a^j b^k$ term:

$$\begin{aligned}(a + b)^n &= (a + b)^{n-1}(a + b) \\ &= \left(\cdots + \binom{n-1}{j-1} a^{j-1} b^k + \binom{n-1}{j} a^j b^{k-1} + \cdots \right) (a + b) \\ &= \left(\cdots + \binom{n-1}{j-1} a^j b^k + \binom{n-1}{j} a^j b^k + \cdots \right).\end{aligned}$$

Therefore the coefficient of $a^j b^k$ is $C(n-1, j-1) + C(n-1, j)$. But this simplifies:

$$\begin{aligned}
 \binom{n-1}{j-1} + \binom{n-1}{j} &= \frac{(n-1)!}{(j-1)!(n-1-(j-1))!} + \frac{(n-1)!}{j!(n-1-j)!} \\
 &= \frac{(n-1)!}{(j-1)!(n-j)!} \cdot \frac{j}{j} + \frac{(n-1)!}{j!(n-1-j)!} \cdot \frac{n-j}{n-j} \\
 &= \frac{(n-1)!(j)}{j!(n-j)!} + \frac{(n-1)!(n-j)}{j!(n-j)!} \\
 &= \frac{(n-1)!(j) + (n-1)!(n-j)}{j!(n-j)!} \\
 &= \frac{n!}{j!(n-j)!} \\
 &= \binom{n}{j}
 \end{aligned}$$

as required. □

This proof was a little messy, but the only tools we used were induction, algebra, and the definition of $C(n, j)$. However, there is another way to look at the result from the point of view of counting. If you were going to expand the product

$$\underbrace{(a+b)(a+b)\cdots(a+b)}_n$$

you would have to use the distributive property repeatedly to get a sum of a bunch of monomial terms, and then you would have to combine like terms. Each monomial is the product of a selection of a 's and b 's; to get a monomial, you must choose either an a or a b from each $(a+b)$ factor and multiply these together. The coefficient of the $a^j b^k$ term is therefore the number of ways you can get the monomial $a^j b^k$. But this is the number of ways you can choose j different $(a+b)$ factors—the factors that contribute an a to the monomial—out of n total: $C(n, j)$.

Often this result is stated as an equation.

Corollary 4.1 The Binomial Theorem.

$$(a+b)^n = \binom{n}{0} a^n + \binom{n}{1} a^{n-1} b + \binom{n}{2} a^{n-2} b^2 + \cdots + \binom{n}{j} a^j b^{n-j} + \cdots + \binom{n}{n} b^n.$$

Example 4.23 Use the binomial theorem to expand $(3x - 5)^4$.

Solution: Apply the corollary with $a = 3x$ and $b = -5$.

$$\begin{aligned} & (3x - 5)^4 \\ &= (3x)^4 + \binom{4}{1}(3x)^3(-5) + \binom{4}{2}(3x)^2(-5)^2 + \binom{4}{3}(3x)(-5)^3 + (-5)^4 \\ &= 81x^4 + (4)(27x^3)(-5) + (6)(9x^2)(25) + (4)(3x)(-125) + (625) \\ &= 81x^4 - 540x^3 + 1350x^2 - 1500x + 625 \end{aligned}$$

Notice that expanding $(3x - 5)^4$ by multiplying polynomials takes a lot more work. ◇

Exercises 4.2

1. A committee of three is chosen from a group of 20 people. How many different committees are possible, if
 - (a) the committee consists of a president, vice president, and treasurer?
 - (b) there is no distinction among the three members of the committee?
2. Hugo and Viviana work in an office with eight other coworkers. Out of these 10 workers, their boss needs to choose a group of four to work together on a project.
 - (a) How many different working groups of four can the boss choose?
 - (b) Suppose Hugo and Viviana absolutely refuse, under any circumstances, to work together. Under this restriction, how many different working groups of four can be formed?
3. Ruth has the following set of refrigerator magnets: $\{A, B, C, D, E, F, G\}$.
 - (a) How many different three-letter strings can she form with these magnets?
 - (b) How many different three-letter strings can she form if the middle letter must be a vowel?
4. Refer to Example 4.22. Use the selection principle to count the number of different possible win or loss scenarios when two teams play a best-of-five match . . .

- (a) using the method of Solution #1.
 - (b) using the method of Solution #2.
5. Form a seven-letter word by mixing up the letters in the word COMBINE.
 - (a) How many ways can you do this?
 - (b) How many ways can you do this if all the vowels have to be at the beginning?
 - (c) How many ways can you do this if no vowel is isolated between two consonants?
6. How many different strings can be formed by rearranging the letters in the word ABABA?
7. Possible grades for a class are A, B, C, D, and F. (No +/-'s.)
 - (a) How many ways are there to assign grades to a class of seven students?
 - (b) How many ways are there to assign grades to a class of seven students, if nobody receives an F and exactly one person receives an A?
8. The school board consists of three men and four women.
 - (a) When they hold a meeting, they sit in a row. How many different seating arrangements are there?
 - (b) How many different ways can the row be arranged if no two women sit next to each other?
 - (c) How many ways are there to select a subcommittee of four board members?
 - (d) How many ways are there to select a subcommittee of four board members if the subcommittee must contain at least two women?
9. The legislature of Puerto Rico consists of a 27-member Senate and a 51-member House of Representatives.
 - (a) How many ways are there to choose a group of six members from the Puerto Rican legislature?
 - (b) How many ways are there to choose a group of six members if three members must come from the Senate and three must come from the House of Representatives?
10. A men's field lacrosse team consists of ten players: three attackmen, three midfielders, three defenders, and one goaltender. Given a set of 10 players, how many different ways are there to assign the roles of attackmen, midfielders, defenders, and goaltender?

224 Chapter 4 ■ Quantitative Thinking

11. There are 10 first-tier national rugby union teams: Argentina, Australia, England, France, Ireland, Italy, New Zealand, Scotland, South Africa, and Wales.
- How many different 2-team pairings are possible among these 10 teams?
 - How many different ways are there to select a first-, second-, and third-ranked team from these 10 teams?
 - Suppose four teams are going to gather in Auckland, and the other six teams are going to gather in Melbourne. How many ways can this be done?
 - Suppose five teams are going to gather in Sydney, and the other five teams are going to gather in Wellington. How many ways can this be done?

12. How many solutions (using only non-negative integers) are there to the following equation?

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = 20$$

13. A certain brand of jellybean comes in four colors: red, green, purple, and yellow. These jellybeans are packaged in bags of 50, but there is no guarantee as to how the colors will be distributed; you might get a mixture of all four colors, or just some red and some green, or even (if you are very lucky) a whole bag of purple.
- Explain how to view the color distribution of a bag of jellybeans as a solution to an equation like the one in Example 4.21.
 - Compute the total number of different possible color distributions.
14. How many different ways are there to distribute 12 identical bones among three different dogs?
15. The North South Line of the Singapore Mass Rapid Transit system has 25 stations. How many different ways are there to divide this line into three segments, where each segment contains at least one station? (One possible such division is shown below.)



16. The streets of many cities (e.g., Vancouver, British Columbia) are based primarily on a rectangular grid. In such a city, if you start at a given street corner, how many different ways are there to walk directly to the street corner that is 5 blocks north and 10 blocks east? (For example, how many ways can you walk from the corner of MacDonald and Broadway to the corner of 4th and Burrard? See Figure 4.6.)

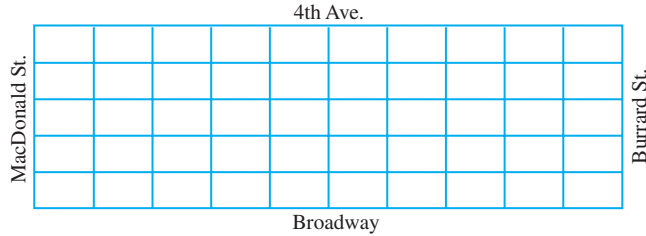
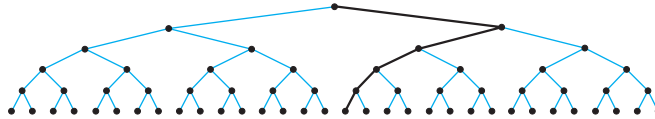


Figure 4.6 Most of the streets in the Canadian city of Vancouver, British Columbia are based on a rectangular grid.

17. The following binary tree has as many nodes as possible for a tree of height 5.



Definition: A *climb* is a path that starts at the root and ends at a leaf. For example, the path indicated by the black lines is a climb.

- How many different climbs are there in this tree?
 - Suppose you have a list of 100 names, and you need to assign a name from this list to each climb. You may not repeat names. How many different ways are there to do this?
 - Notice that as a climb goes from the root to a leaf, it must go either right or left at each node. We say that a climb has a *change of direction* if it goes right after having gone left, or left after having gone right. For example, the climb indicated in the above figure has one change of direction. How many climbs are there with exactly two changes of direction?
 - Suppose you were given a binary tree like the one above, with as many nodes as possible, but with height 10. In this new tree, how many climbs are there with exactly two changes of direction?
18. Let $n \geq 0$ and let j and k be non-negative integers such that $j + k = n$. Use algebra to prove that $C(n, j) = C(n, k)$.
19. Use the Binomial Theorem to expand $(2x + 7)^5$.
20. Use the Binomial Theorem to expand $(x + 1)^{10}$.
21. Compute the coefficient of x^8 in the expansion of $(3x - 2)^{13}$.

226 Chapter 4 ■ Quantitative Thinking

*22. Let $R(n, j)$ be a function of two variables that is defined recursively as follows.

$$\mathbf{B.} \quad R(n, 0) = R(n, n) = 1 \text{ for all } n \geq 0.$$

$$\mathbf{R.} \quad R(n, j) = R(n - 1, j) + R(n - 1, j - 1).$$

Prove (using induction and one of the identities in the proof of Theorem 4.3) that $R(n, j) = C(n, j)$ for all $n \geq j \geq 0$.

*23. The following diagram is called *Pascal's Triangle*, named after the philosopher/theologian/mathematician Blaise Pascal (1623–1662). Explain what this diagram has to do with $C(n, j)$. (Use the result of the previous problem in your explanation.)

$$\begin{array}{cccccc}
 & & & & & 1 \\
 & & & & & 1 & 1 \\
 & & & & 1 & 2 & 1 \\
 & & & 1 & 3 & 3 & 1 \\
 & & 1 & 4 & 6 & 4 & 1 \\
 1 & \dots & \dots & \dots & \dots & \dots & 1
 \end{array}$$

4.3 Counting with Functions

When faced with a new kind of mathematical problem, it often helps to relate it to something familiar. In Examples 4.17 and 4.21, our counting arguments were based on seeing relationships between mathematical objects. We observed that $9!$ batting orders correspond to each choice of nine starters, and we saw that every solution to a particular equation could be represented uniquely as a certain kind of string. These relationships can be thought of as functions. In this section, we explore how to count using this type of relational thinking.

4.3.1 One-to-One Correspondences

The next two observations are fairly easy to prove. The proofs are left as exercises.

Theorem 4.4 *Let $|X| = m$ and $|Y| = n$. If there is some $f: X \rightarrow Y$ that is one-to-one, then $m \leq n$.*

Theorem 4.5 *Let $|X| = m$ and $|Y| = n$. If there is some $f: X \rightarrow Y$ that is onto, then $m \geq n$.*

Together, they imply this corollary.

Corollary 4.2 Let $|X| = m$ and $|Y| = n$. If there is a one-to-one correspondence $f: X \rightarrow Y$, then $m = n$.

The term “one-to-one correspondence” suggests exactly what this corollary states: if two finite sets are in one-to-one correspondence with each other, then the sets have the same number of elements. Although we seldom use this result explicitly when we solve counting problems, it can often guide our thinking.

Example 4.24 In a single elimination tournament players are paired up in each round, and the winner of each match advances to the next round. If the number of players in a round is odd, one player gets a bye to the next round. The tournament continues until only two players are left; these two players play the championship game to determine the winner of the tournament. In a tournament with 270 players, how many games must be played?

Solution: This problem is easy if you realize that there is a one-to-one correspondence

$$f: G \rightarrow L$$

where G is the set of all games played, and L is the set of players who lose a game. The function is defined for any game x as $f(x) = l$, where l is the loser of game x . Since every game has a single loser, the function is well-defined. Since this is a single elimination tournament, no player can lose two different games, so f is one-to-one. And since every loser lost some game, f is onto. So, by Corollary 4.2, the number of games equals the number of losers. The winner of the tournament is the only nonloser, so there are 269 losers, hence 269 games. \diamond

Example 4.25 Draw a diagram with six lines subject to the following conditions:

- Every line intersects every other line.
- No three lines intersect in a single point.

See Figure 4.7 for an example. Your diagram will form lots of overlapping triangles. How many triangles are there?

Solution: Observe that every triangle is formed from three lines, and any set of three lines forms a triangle. Thus, there is a one-to-one correspondence

$$\{\text{triangles in figure}\} \longleftrightarrow \{\text{sets } \{l_1, l_2, l_3\} \mid l_i \text{ is a line}\}.$$

So to count the number of triangles, we can just count the number of sets of three lines. There are $C(6, 3) = 20$ of these. \diamond

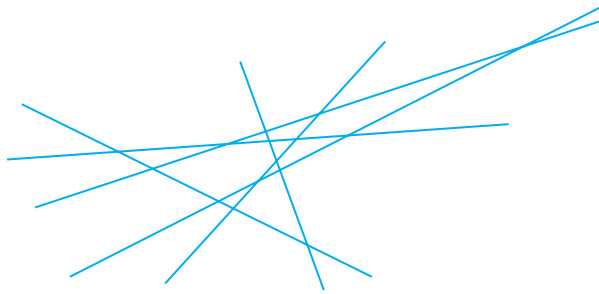


Figure 4.7 Can you find all 20 triangles?

The idea in these last two examples is clear: to count the elements in a set Y , find some one-to-one correspondence $f: X \rightarrow Y$, and count X instead. We can extend this technique by considering functions with the following property.

Definition 4.1 A function $f: X \rightarrow Y$ is called *n-to-one* if every y in the image of the function has exactly n different elements of X that map to it. In other words, f is *n-to-one* if

$$|\{x \in X \mid f(x) = y\}| = n$$

for all $y \in f(X)$.

Note that this definition coincides with the definition of one-to-one when $n = 1$: if there is exactly one $x \in X$ such that $f(x) = y$ for every y in the image, then the only way $f(a) = f(b)$ can happen is if $a = b$.

Example 4.26 Let $X = \{0, 1, 2, 3, 4, 5, 6\}$ and let $Y = \{0, 1\}$. Define a function $m: X \rightarrow Y$ by

$$m(x) = x \pmod{2}.$$

This function is three-to-one, because there are three numbers that map to 0 and three numbers that map to 1.

The next theorem is just the observation that the domain of an *n-to-one* function must be n times as big as the image.

Theorem 4.6 Let $|X| = p$ and $|Y| = q$. If there is an *n-to-one* function $f: X \rightarrow Y$ that maps X onto Y , then $p = qn$.

We have already used this idea to relate permutations and combinations. Let's revisit that discussion.

Example 4.27 Let S be a set with k elements, let X be the set of all arrangements of r elements of S , and let Y be the set of all selections (i.e., subsets) of r elements of S . Define a function $f: X \rightarrow Y$ by

$$f(x_1x_2x_3 \cdots x_r) = \{x_1, x_2, x_3, \dots, x_r\}.$$

This function is onto. Since there are exactly $r!$ ways to arrange any r elements, this function is also $r!$ -to-one. Therefore $|X| = r! \cdot |Y|$, or equivalently, $P(n, r) = r! \cdot C(n, r)$.

Example 4.28 How many different strings can you form by rearranging the letters in the word ENUMERATE?

Solution: This would be a simple arrangement problem, except that ENUMERATE contains repeated letters—three E’s, to be precise. Let’s pretend those E’s are different for a moment: call them E_1 , E_2 , and E_3 . Let X be the set of all strings you can form by rearranging the letters in the word $E_1\text{NUME}_2\text{RATE}_3$. Since the elements of X are just permutations of nine distinct symbols, $|X| = 9!$. Now let Y be the number of ways to rearrange ENUMERATE, and define a function $f: X \rightarrow Y$ by $f(\lambda) = \lambda'$, where λ' is the string λ with the subscripts on the E’s removed. This function is onto, because you can always take a string in Y and put the subscripts 1, 2, and 3 on the E’s. Moreover, there are exactly $3! = 6$ ways to do this, so f is six-to-one. Therefore, by Theorem 4.6, $|X| = 6 \cdot |Y|$, so there are $|Y| = 9!/6 = 60,480$ arrangements. \diamond

Example 4.29 A group of 10 people sits in a circle around a campfire. How many different seating arrangements are there? In this situation, a seating arrangement is determined by who sits next to whom, not by where on the ground they sit. Let’s also agree not to distinguish between clockwise and counterclockwise; all that matters is who your two neighbors are, not who is on your left and who is on your right.

Solution: First, consider the related problem of seating ten people in a row. The set X of all such arrangements has $10!$ elements. Now define a function $f: X \rightarrow Y$ from X to the set Y of all circular seating arrangements as follows. If λ is a row seating arrangement, then $f(\lambda)$ is the circular seating arrangement you get by curving the row into a circle and joining the endpoints, as shown in Figure 4.8.

Notice that this map is onto, because given any $\lambda' \in Y$, you can find a $\lambda \in X$ that maps to it by breaking the circle between two people (say a and b) and straightening out the circle to form a row, with a on one end and b on the other. Also notice that this can be done in exactly 20 ways, because there are 10 different places to break the circle, and then there are two choices for

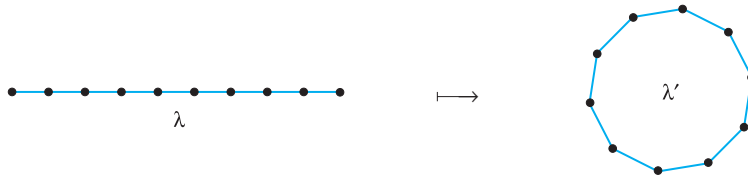


Figure 4.8 Mapping a row to a circle.

where to put a : on the left end of the row or on the right end. Therefore, f is a twenty-to-one function. By Theorem 4.6, $|Y| = 10!/20 = 181,440$. \diamond

4.3.2 The Pigeonhole Principle

The Pigeonhole Principle is the simple observation that if you put n pigeons into r holes, and $n > r$, then some hole must contain multiple pigeons. We can state this a little more mathematically using functions.

Theorem 4.7 *Let $|X| = n$ and $|C| = r$, and let $f: X \rightarrow C$. If $n > r$, then there are distinct elements $x, y \in X$ with $f(x) = f(y)$.*

Proof We proceed by contraposition. Suppose that for all pairs of distinct elements $x, y \in X$, $f(x) \neq f(y)$. This is the same as saying that f is one-to-one. By Theorem 4.4, this implies that $n \leq r$. \square

Instead of pigeons and holes, it sometimes helps to think of X as a set of objects and C as a set of colors. The function f assigns a color to each object, and if $|X| > |C|$, there is some pair of objects with the same color.

The following examples are direct applications of Theorem 4.7.

Example 4.30 In a club with 400 members, must there be some pair of members who share the same birthday?

Solution: Yes. Let X be the set of all people, and let C be the set of all possible birthdays. Let $f: X \rightarrow C$ be the defined so that $f(x)$ is the birthday of person x . Since $|X| > |C|$, there must be two people x and y with the same birthday, that is, with $f(x) = f(y)$. \diamond

Example 4.31 Chandra has a drawer full of 12 red and 14 green socks. In order to avoid waking his roommate, he must grab a selection of clothes in the dark and get dressed out in the hallway. How many socks must he grab in order to be assured of having a matching pair?

Solution: Let $C = \{\text{red, green}\}$ and let X be the number of socks Chandra selects. Let $f: X \rightarrow C$ be the function that assigns a color to each sock. There are two colors, so he needs $|X| > 2$ socks. Three is enough. \diamond

Example 4.32 In a round-robin tournament, every player plays every other player exactly once. Prove that, if no player goes undefeated, at the end of the tournament there must be two players with the same number of wins.

Solution: Apply Theorem 4.7 with X being the set of players, and let $|X| = n$. Each player plays $n - 1$ games, and no player wins every game, so the set of all possible numbers of wins is $C = \{0, 1, 2, \dots, n - 2\}$. Define $f: X \rightarrow C$ so that $f(x)$ is the number times player x wins. Since $|C| < |X|$, there exists a pair of players with the same number of wins. \diamond

4.3.3 The Generalized Pigeonhole Principle

We can extend the Pigeonhole Principle a little. If you have way more pigeons than holes, you would expect to find some hole with lots of pigeons in it.

Theorem 4.8 *Let $|X| = n$ and $|C| = r$, and let $f: X \rightarrow C$. If $n > r(l - 1)$, then there is some subset $U \subseteq X$ such that $|U| = l$ and $f(x) = f(y)$ for any $x, y \in U$.*

Note that if $l = 2$, this is Theorem 4.7. We can restate the theorem in terms of colors: “Suppose each object in a set X of n objects is assigned a color from a set C of r colors. If $n > r(l - 1)$, then there is a subset $U \subseteq X$ with l objects, all of the same color.”

Proof (By contraposition.) Suppose that, when you group the elements of X according to color, the size of any of these groups is at most $l - 1$. Then the total number of elements of X is at most $r(l - 1)$. \square

When you know n and r , it is handy to have a formula for l .

232 Chapter 4 ■ Quantitative Thinking

Corollary 4.3 Let $|X| = n$ and $|C| = r$, and let $f: X \rightarrow C$. Then there is some subset $U \subseteq X$ such that

$$U = \left\lceil \frac{n}{r} \right\rceil$$

and $f(x) = f(y)$ for any $x, y \in U$.

Proof Let $l = \lceil n/r \rceil$. Then

$$r(l-1) = (\lceil n/r \rceil - 1)r \leq (n/r - 1)r = n - r < n,$$

so the result follows from Theorem 4.8. \square

Example 4.33 A website displays an image each day from a bank of 30 images. In any given 100 day period, show that some image must be displayed four times.

Solution: Apply Theorem 4.8, with X being the set of days and C being the set of images. Let $f: X \rightarrow C$ be the function that returns the image $f(x)$ that gets displayed on day x . Since $100 > 30(4-1)$, there is some image that will be displayed four times. \diamond

Alternatively, we could have used Corollary 4.3: $\lceil 100/30 \rceil = \lceil 3.\overline{3} \rceil = 4$.

Example 4.34 Let G be the complete graph on six vertices. (See Figure 4.9.) This graph has 15 edges. Suppose that some edges are colored red and the rest green. Show that there must be some triangular circuit whose edges are the same color.

Solution: Pick a vertex v . There are five edges on v , so by Theorem 4.8 three of these edges e_1, e_2, e_3 must be the same color (say green, without loss of generality). Let the other vertices of edges e_1, e_2, e_3 be x, y, z , respectively. If the triangular circuit formed by x, y, z has all red edges, we are done. But if one of the edges is green, then it forms a green triangular circuit with v . \diamond

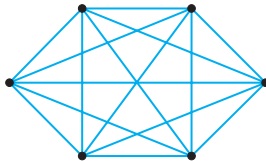


Figure 4.9 The complete graph on six vertices.

4.3.4 Ramsey Theory ‡

Example 4.34 shows that no matter how mixed up the edge colors of G are, there will always be a monochromatic triangle. This example illustrates a general phenomenon: no matter how disordered something might be, there will always be some small part of it that has some kind of order. The following theorem (which we will not prove) states this mathematically.

Theorem 4.9 (Ramsey) *For any positive integers r , k , and l , there is a positive integer n such that, if the k -element subsets of a set X with n elements are colored with r colors, then there is a subset $U \subseteq X$ with $|U| = l$ whose k -element subsets are all the same color.*

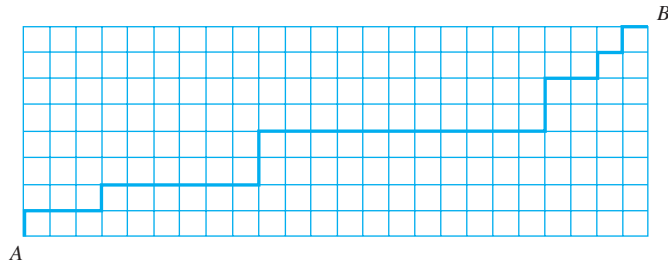
Given r , k , and l , the *Ramsey number* $R(r, k, l)$ is the smallest n satisfying the conclusion of the theorem. In Example 4.34, the edges in the graph represent all the two element subsets of the set X of six vertices. Thus we have $k = 2$ and $n = 6$. Since there are two colors, $r = 2$. The monochromatic triangle corresponds to a subset U with three elements, so $l = 3$. The example shows that, for a set X with six elements, there is always a three-element subset of X whose two-element subsets are all the same color. In other words, the example establishes that $R(2, 2, 3) \leq 6$. In the exercises, you will show that equality must hold by showing that $n = 5$ is not enough vertices to satisfy the conclusion of the theorem.

Exercises 4.3

1. Prove Theorem 4.4.
2. Prove Theorem 4.5.
3. Let G be the complete graph on n vertices. In other words, G is a simple graph with n vertices in which every vertex shares an edge with every other vertex.
 - (a) Explain why there is a one-to-one correspondence between the set of all (unordered) pairs of vertices in G and the set of all edges of G .
 - (b) Use part (a) to count the number of edges of G (in terms of n).

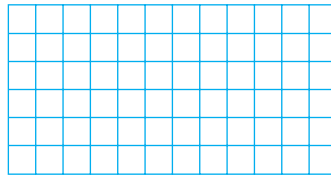
234 Chapter 4 ■ Quantitative Thinking

4. The following 8×24 grid is divided into squares that are 1 unit by 1 unit.



The shortest possible path on this grid from A to B is 32 units long. One such path is shown in the figure. Let X be the set of all 32-unit-long paths from A to B .

- There is a one-to-one correspondence between X and the set Y of all binary strings with 8 1's and 24 0's. Describe the function, and explain why it is a one-to-one correspondence.
 - Compute $|X|$, the number of 32-unit-long paths from A to B .
5. Why is there no such thing as a “one-to- n function,” for $n > 1$?
6. The following figure consists of 7 horizontal lines and 13 vertical lines. The goal of this problem is to count the number of rectangles (squares are a kind of rectangle, but line segments are not).

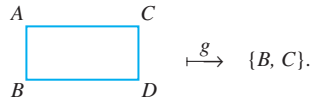


Let V be the set of all sets of two vertical lines, and let H be the set of all sets of two horizontal lines. Let R be the set of all rectangles in the figure. Define a function $f : R \rightarrow V \times H$ by

$$\begin{array}{ccc}
 A & & C \\
 \square & & \\
 B & & D
 \end{array}
 \xrightarrow{f}
 (\{AB, CD\}, \{AC, BD\}).$$

- Explain why f is well defined.
- Explain why f is one-to-one.
- Explain why f is onto.

- (d) Compute $|R|$, the number of rectangles in the figure.
- (e) The previous figure contains $7 \cdot 13 = 91$ intersection points. Let P be the set of all sets $\{X, Y\}$ of two intersection points. Suppose we tried counting R by defining a function $g: R \rightarrow P$ which maps a rectangle to the set containing its lower left and upper right vertices:



Explain why g is not a one-to-one correspondence.

7. Let S be a set of n numbers. Let X be the set of all subsets of S of size k , and let Y be the set of all ordered k -tuples (s_1, s_2, \dots, s_k) such that $s_1 < s_2 < \dots < s_k$. That is,

$$X = \{ \{s_1, s_2, \dots, s_k\} \mid s_i \in S \text{ and all } s_i \text{'s are distinct} \}, \text{ and}$$

$$Y = \{ (s_1, s_2, \dots, s_k) \mid s_i \in S \text{ and } s_1 < s_2 < \dots < s_k \}.$$

- (a) Define a one-to-one correspondence $f: X \rightarrow Y$. Explain why f is one-to-one and onto.
- (b) Determine $|X|$ and $|Y|$.
8. Let X be a set with n elements, and let $\mathcal{P}(X)$ be its power set.

- (a) Describe a one-to-one correspondence

$$f: \mathcal{P}(X) \rightarrow S$$

where S is the set of all n -digit binary strings.

- (b) Use this one-to-one correspondence to compute $|\mathcal{P}(X)|$.
- * (c) Let $P_k \subseteq \mathcal{P}(X)$ be the set of all k -element subsets of X , for $0 \leq k \leq n$. The restriction

$$f|_{P_k}: P_k \rightarrow S$$

is one-to-one. What is the image of $f|_{P_k}$?

- * (d) How many elements are in the image of $f|_{P_k}$?

9. How many ways are there to rearrange the letters in FUNCTION?
10. How many ways are there to rearrange the letters in BANANA?
11. How many ways are there to rearrange the letters in INANENESS?
12. Use the one-to-one correspondence defined in Example 2.30 on page 93 to count the number of points of intersection in Figure 4.10, not counting the points that lie on the circle.

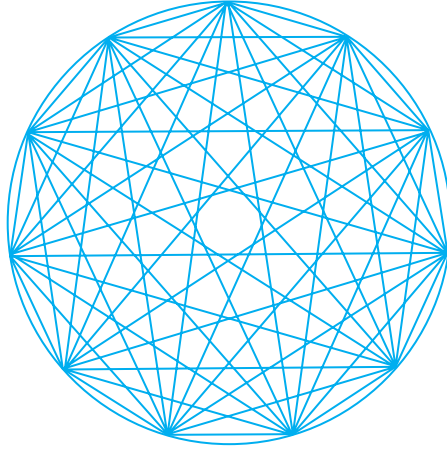
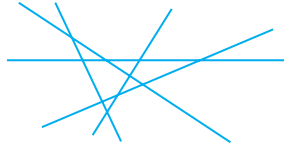


Figure 4.10 How many points of intersection are there? See Exercise 12.

13. Consider a diagram with n lines, all of which intersect, but no three of which pass through a single point. Here is an example with $n = 5$.



Let Y be the set of all points of intersection, and let X be the set of all sets of two lines. That is,

$$X = \{\{l_1, l_2\} \mid l_1 \text{ and } l_2 \text{ are distinct lines in the diagram}\}.$$

Define a function $f: X \rightarrow Y$ by setting $f(\{l_1, l_2\})$ equal to the point where l_1 and l_2 intersect.

- Explain why f is one-to-one.
 - Explain why f is onto.
 - How many point of intersection are there? Give your answer in terms of n .
14. Suppose you have eight squares of stained glass, all of different colors, and you would like to make a rectangular stained glass window in the shape of a 2×4 grid.



How many different ways can you do this, taking symmetry into account? (Note that any pattern may be rotated 180° , flipped vertically, or flipped horizontally. You should count all the possible resulting patterns as the same window.)

15. Suppose you have four squares of stained glass, all of different colors, and you wish to make a 2×2 square stained glass window. How many different windows are possible? (Beware: a square has more symmetries than a rectangle.)
16. A certain board game uses tokens made of transparent colored plastic. Each token looks like



where each of the four different regions is a different color: either red, green, yellow, blue, orange, or purple. How many different tokens of this type are possible?

17. A different board game also uses tokens made of transparent colored plastic. In this game, each token looks like



where each of the five different regions is a different color: either red, green, yellow, blue, orange, or purple. How many different tokens of this type are possible?

18. On the eve of an election, a radio station is forced to play 20 campaign ads in a row. Of these 20 ads, 15 are for the Tory candidate, and 5 are for the Labour candidate. Prove that the station must play at least three Tory ads in a row at some point. Use the generalized pigeonhole principle to justify your answer.
19. Explain why, in a class of 36, there will always be a group of at least 6 who were born on the same day of the week.
20. Let G be a simple graph with two or more vertices. Prove that there is a pair of vertices in G having the same degree.
21. Let $\triangle ABC$ be an equilateral triangle whose sides are two inches long. Prove that it is impossible to place five points inside the triangle without two of them being within one inch of each other.

238 Chapter 4 ■ Quantitative Thinking

22. A small college offers 250 different classes. No two classes can meet at the same time in the same room, of course. There are twelve different time slots at which classes can occur. What is the minimum number of classrooms needed to accommodate all the classes?
- *23. Use the pigeonhole principle to explain why every rational number has a decimal expansion that either terminates or repeats. In the case where a rational number m/n has a repeating decimal expansion, find an upper bound (in terms of the integer n) on the number of digits in the part that repeats. (Hint: In the long division problem

$$n \overline{)m}$$

consider the possible remainders at each step of the algorithm for long division.)

24. Suppose that 100 lottery tickets are given out in sequence to the first 100 guests to arrive at a party. Of these 100 tickets, only 12 are winning tickets. The generalized pigeonhole principle guarantees that there must be a streak of at least l losing tickets in a row. Find l .
25. Show that $R(2, 2, 3) > 5$ by coloring the edges of the complete graph on five vertices red and green in such a way that no triangular circuit has edges of a single color.
- *26. Show that if the edges of the complete graph on eight vertices are colored red and green, then there is either a three-circuit or a four-circuit whose edges are all the same color.

4.4 Discrete Probability

The previous three sections have introduced a range of tools for counting the elements in a finite set. While counting arguments are interesting and somewhat diverting, they also have important applications. One way to see connections between enumeration techniques and real-world problems is from the perspective of probability.

Intuitively, probability tells us how likely something is to occur. The probability of an event is a number between 0 and 1, with 0 representing impossibility and 1 representing certainty. We deal informally with probabilities often; statements like “there is a 40% chance of rain this weekend,” or “the odds of winning are 1 in 200,000,000,” are making quantitative predictions of some future event.

4.4.1 Definitions and Examples

The mathematical definition of probability is based on enumeration. Most of the counting problems in the previous sections are of the form

How many ⟨blanks⟩ have ⟨some property⟩?

More formally, this question is asking

How many elements of set U are in some subset A ?

which is basically the same as

What percentage of elements of U are in the subset A ?

The answer to this last question is just a ratio based on a counting problem. This ratio can be thought of in terms of chance:

What is the probability that a randomly chosen element of U is in the subset A ?

The following definition summarizes this discussion.

Definition 4.2 Suppose A is a subset of a nonempty finite set U . The *probability* that a randomly chosen element of U lies in A is the ratio

$$P(A) = \frac{|A|}{|U|}.$$

The set U is called the *sample space*, and the set A is called an *event*.

Example 4.35 Suppose you get a random license plate from all the possible plates described in Exercise 4.10. What is the probability that your plate contains the word CUB or the word SOX?

Solution: The sample space U is the set of all possible plates, which we found to have 24,336,000 elements. We are interested in the event A that a plate contains the word CUB or SOX, and these two cases are disjoint. If a plate contains one of these words, it must be the type that has three letters followed by three digits. There are 10^3 choices for the digits on such a plate, so there are 10^3 plates in each case. Thus the desired probability is

$$\frac{|A|}{|U|} = \frac{10^3 + 10^3}{24,336,000} = \frac{1}{12168} \approx 0.000082,$$

which is not very likely. ◇

Example 4.36 If you roll two standard six-sided dice, what is the probability that you roll an 8 (i.e., that the sum of the values on the two dice will be 8)?

Solution: It is important to be clear about our sample space. If D_1 represents the six possible values of the first die and D_2 represents the six values of the second, then our sample space is the Cartesian product $D_1 \times D_2$. (Note that, although the dice may be identical, the outcome that the first is 3 and the second is 5 is different from the first being 5 and the second 3.) Thus the size of the sample space is $|D_1 \times D_2| = |D_1| \cdot |D_2| = 36$. The following ordered pairs

$$\{(2, 6), (3, 5), (4, 4), (5, 3), (6, 2)\}$$

represent the event of rolling an 8. Hence the probability of such a roll is $5/36 \approx 0.139$. \diamond

The next theorem presents an important fact that makes some probability calculations easier.

Theorem 4.10 *Let A be a subset of a nonempty finite set U . Let A' be the complement of A in U . If $P(A) = p$, then $P(A') = 1 - p$.*

Proof Exercise. (Use the Addition Principle.) \square

In other words, if p is the probability that some event A happens, then $1 - p$ is the probability that A does *not* happen. The following example shows how to use this trick.

Example 4.37 If you roll two standard six-sided dice, what is the probability you roll 10 or less?

Solution: We could count up the ways of rolling 2, 3, 4, up to 10, and then add (since these cases are mutually exclusive) to get the size of this event. But it is easier to compute the probability of rolling more than 10. There are two ways to roll 11 and one way to roll 12, so the probability of rolling more than 10 is $(2+1)/36 = 1/12$. Thus the probability of rolling 10 or less is $1 - 1/12 = 11/12$, by Theorem 4.10. \diamond

4.4.2 Applications

Many applications of enumeration techniques involve probability. In business, probability is important because it helps measure risk. For example, choosing the right location for a restaurant can play a major role in determining how many diners show up. A restaurant owner needs to plan accordingly. The next example shows how this situation can present a discrete problem.

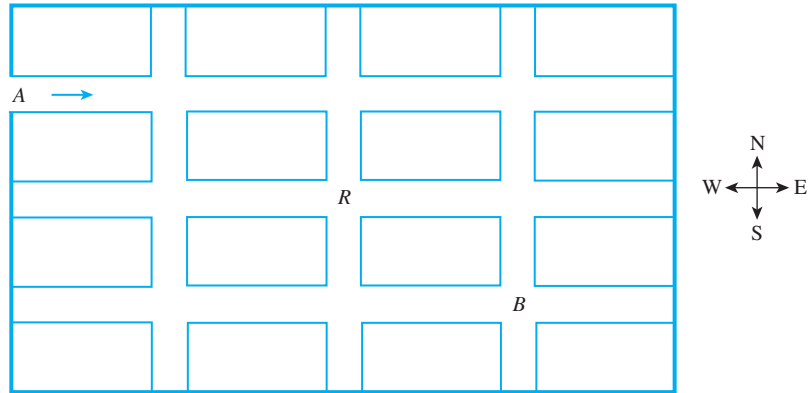
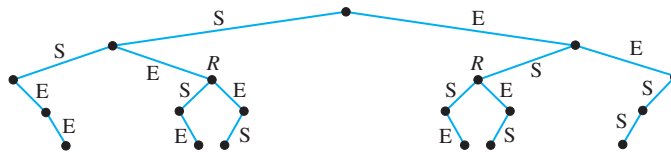


Figure 4.11 A shopping district with restaurant R and bookstore B . See Example 4.38.

Example 4.38 The streets of a shopping district are laid out on a grid, as in Figure 4.11. There is a restaurant at point R and a bookstore at point B . Suppose a customer enters the shopping district at point A and begins walking in the direction of the arrow. At each intersection, the customer chooses to go east or south, while taking as direct a path as possible to the bookstore at point B . Assuming that all such paths are equally likely, what is the probability that the customer passes by the restaurant at point R on the way to point B ?

Solution: Since all the direct paths from A to B are equally likely, the set of these paths is our sample space. The event is the set of all such paths that contain point R . In Example 4.8, we used a decision tree to count all the possible paths from A to B . We can use the same tree to determine how many of these paths pass by the restaurant at R .



Recall from Example 4.8 that the nodes represent the intersections, with the leaves representing B and the labeled nodes representing R . Of the six possible direct paths, four pass through R , so the desired probability is $4/6 = 2/3$. \diamond

Example 4.39 Suppose that there are 10 defective machines in a group of 200. A quality control inspector takes a sample of three machines and tests them for defects. How likely is it that the inspector discovers a defective machine?

242 Chapter 4 ■ Quantitative Thinking

Solution: We need to compute the probability that a defective machine shows up in the random sample. The sample space is the total number of selections of three machines: $C(200, 3)$. The event that at least one of the machines is defective is the opposite of the event that none are. There are 190 nondefective machines, so $C(190, 3)$ samples contain no defects. Therefore the desired probability is

$$1 - \frac{C(190, 3)}{C(200, 3)} = 1 - \frac{1,125,180}{1,313,400} \approx 0.1433.$$

Thus, it isn't very likely that this method of testing reveals a defect. \diamond

The previous two examples illustrate how discrete probability can apply to real-world situations. Compared to these applications, the following “urn problem” may seem more contrived, but it serves as an important template for future calculations.

Example 4.40 An urn contains seven red balls, seven white balls, and seven blue balls. A sample of five balls is drawn at random without replacement. What is the probability that the sample contains three balls of one color and two of another?

Solution: Nothing in this problem refers to the order of the sample, so we can consider the sample to be an unordered set—that is, a selection—of five balls from a total of 21. So there are $C(21, 5) = 20,349$ possible samples in the sample space. In order to count the number of samples with three balls of one color and two of another, we must make several choices in sequence.

1. Choose the color of the three balls.
2. Choose three balls of that color.
3. Choose the color of the two balls.
4. Choose two balls of that color.

There are three colors, so there are three options for the first choice. Then, no matter which color was chosen, there are $C(7, 3) = 35$ ways to choose three balls of this color. Similarly, choice number three has two options, since two colors remain. Then there are $C(7, 2) = 21$ ways to choose a pair of balls of the second color. Since these four choices are made in sequence, we use the multiplication principle to compute the size of the sample: $3 \cdot 35 \cdot 2 \cdot 21$. Therefore

$$\frac{3 \cdot 35 \cdot 2 \cdot 21}{20,349} \approx 0.2167$$

is the desired probability. \diamond

Real-world situations that involve *sampling*—choosing some elements of a population at random—can often be thought of as urn problems. While the study of random sampling lies beyond the scope of this book, the next example indicates the types of questions that courses in probability and statistics can address.

Example 4.41 Rodelio wants to know if a majority of the voters in his town will support his candidacy for mayor. Suppose, for the sake of this discussion, that out of the 300 voters in this town, 151 support Rodelio (but Rodelio doesn't know this information). Rodelio selects 20 voters at random from the population of 300. What is the probability that, out of this random sample, fewer than five support Rodelio?

Solution: Think of the 300 voters as 300 balls in an urn, 151 of which are colored red (for Rodelio) and 149 of which are colored blue. The random sample is then a selection of 20 balls at random from the urn. The sample space U is the set of all possible random samples, so $|U| = C(300, 20)$. The event A that fewer than 5 of the voters in this sample support Rodelio is the number of ways to draw 20 balls such that the number of red balls is 0, 1, 2, 3, or 4. Therefore,

$$\begin{aligned} |A| = & C(149, 20) + C(151, 1) \cdot C(149, 19) + C(151, 2) \cdot C(149, 18) \\ & + C(151, 3) \cdot C(149, 17) + C(151, 4) \cdot C(149, 16). \end{aligned}$$

The desired probability is $P(A) = |A|/|U| \approx 0.0042$. ◇

What can we infer from the above calculation? Assuming that a majority—even the slightest possible majority—supports Rodelio, it is very unlikely that a poll of 20 will reveal as few as four supporters. If Rodelio were to obtain such an outcome from his poll, he should be very discouraged: either the result was extremely unlucky, or the supposition that a majority supports Rodelio is false. A result of 4 out of 20 in this poll is strong evidence that Rodelio is going to lose the election.

4.4.3 Expected Value

So far we have seen examples where probability quantifies how likely an event is to occur. Probability can also be used to make predictions about the value of a *random variable*.

Informally,¹ a random variable is a numerical result of a random experiment or process. We can extend our notation of probability to describe the probability that a random variable falls within a certain range. For example, $P(X = x)$ is the probability that the random variable X takes the particular value x .

1. We omit the formal definition of a random variable.

Example 4.42 Rolling two standard six-sided dice is a random experiment. The sum of the values on the two dice is a random variable X . In Example 4.36, we showed that $P(X = 8) = 5/36$. In Example 4.37, we showed that $P(X \leq 10) = 11/12$.

Example 4.43 Select a random passenger on the London Underground and measure the passenger's nose. Let X be the length of the nose in centimeters. Then X is a random variable, and $P(0 \leq X \leq 50) = 1$ (assuming nobody has a nose larger than 50 cm).

Suppose that we repeat a random experiment several times and record the value of a random variable each time. If we averaged these results, we would get an estimate of the probabilistic “average” of the random variable. For example, if we measured the noses of 15 randomly selected passengers on the London Underground and averaged these measurements, we would probably get a number that was somewhat representative of the population of all Underground passengers. If we were somehow able to measure the noses of everyone on the Underground, we would get the average, or *expected*, nose size of the population.

Definition 4.3 Let x_1, x_2, \dots, x_n be all of the possible values of a random variable X . Then X 's *expected value* $E(X)$ is the sum

$$\sum_{i=1}^n x_i \cdot P(X = x_i).$$

That is, $E(X) = x_1 \cdot P(X = x_1) + x_2 \cdot P(X = x_2) + \dots + x_n \cdot P(X = x_n)$.

Example 4.44 In the “Both Ways” version of Australia’s “Cash 3” lottery, you pick a three-digit number from 000 to 999, and a randomly chosen winning three-digit number is announced every evening at 5:55 p.m. If you pick a number with three distinct digits, you win \$580 if your number matches the winning number exactly, and you win \$80 if the digits of your number match the digits of the winning number, but in a different order. What is the expected value of the amount of money you win?

Solution: Let X be the amount of money you win. The possible values of X are 0, 580, and 80. Out of 1000 possible winning numbers, only one matches your number exactly, and five ($3! - 1$) have the same digits, but in a different order. Therefore your expected winnings are

$$\begin{aligned} E(X) &= 0 \cdot P(X = 0) + 580 \cdot P(X = 580) + 80 \cdot P(X = 80) \\ &= 0 \cdot \frac{994}{1000} + 580 \cdot \frac{1}{1000} + 80 \cdot \frac{5}{1000} \\ &= 0.98. \end{aligned}$$

Intuitively, this result means that if you play the lottery many times and average your winnings, in the long run you can expect the average to be around \$0.98. Given that it costs \$2 to play this game, playing the lottery is a losing venture (for you, but not for the Western Australia community, which receives the profits from Cash 3). \diamond

Exercises 4.4

1. A computer generates a random four-digit string in the symbols A, B, C, . . . , Z.
 - (a) How many such strings are possible?
 - (b) What is the probability that the random string contains no vowels (A, E, I, O, U)?
2. A multiple choice test consists of five questions, each of which has four choices. Each question has exactly one correct answer.
 - (a) How many different ways are there to fill out the answer sheet?
 - (b) How many ways are there to fill out the answer sheet so that four answers are correct and one is incorrect?
 - (c) William guesses randomly at each answer. What is the probability that he gets three or fewer questions correct?
3. A random number generator produces a sequence of 20 digits (0, 1, . . . , 9). What is the probability that the sequence contains at least one 3? (Hint: Consider the probability that it contains no 3's.)
4. Refer to Example 4.38. For each of the seven unlabeled intersections in Figure 4.11, find the probability that the customer passes through the intersection on the way to the bookstore.
5. If you roll two six-sided dice, what is the probability of rolling a 7?
6. If you roll two four-sided dice (numbered 1, 2, 3, and 4), what is the probability of rolling a 5?
7. If you roll a four-sided die and a six-sided die, which roll totals (2, 3, . . . , 10) have the highest probability?

246 Chapter 4 ■ Quantitative Thinking

8. Refer to Example 4.10. Assume that all the different possible license plates are equally likely.

- (a) What is the probability that a randomly chosen plate contains the number 9999?
- (b) What is the probability that a randomly chosen plate contains the substring HI? (For example, HI4321 or PHI786 are two ways HI might appear.)

9. The `tinyurl.com` service lets you alias a long URL to a shorter one. For example, you can ask the service to send people to

`http://www.expensive.edu/departments/math/hard/discrete/`

when they enter the URL `http://tinyurl.com/sd8k3` in a web browser.

- (a) How many different URL's of the form `http://tinyurl.com/*****` are possible, if `*****` can be any string of the characters `a, b, ..., z`, and `0, 1, ..., 9`? (Repeated characters are allowed.)
 - (b) How many different URL's of the form `http://tinyurl.com/*****` are possible, if `*****` must be a string consisting of three letters (`a, b, ..., z`) followed by two digits (`0, 1, ..., 9`)? (Repeats allowed.) For example, `ace44`, `cub98`.
 - (c) Suppose that an arbitrary string `*****` of letters and digits (as in question 9a) is chosen at random. What is the probability that this string contains no digits?
10. In a class of 11 boys and 9 girls, the teacher selects three students at random to write problems on the board. What is the probability that all the students selected are boys?
11. Refer to Example 4.40. An urn contains seven red balls, seven white balls, and seven blue balls, and sample of five balls is drawn at random without replacement.
- (a) Compute the probability that the sample contains four balls of one color and one of another color.
 - (b) Compute the probability that all of the balls in the sample are the same color.
 - (c) Compute the probability that the sample contains at least one ball of each color.

12. An urn contains five red balls and seven blue balls. Four balls are drawn at random, without replacement.
 - (a) What is the probability that all four balls are red?
 - (b) What is the probability that two of the balls are red and two are blue?
13. In a class of 17 students, 3 are math majors. A group of four students is chosen at random.
 - (a) What is the probability that the group has no math majors?
 - (b) What is the probability that the group has at least one math major?
 - (c) What is the probability that the group has exactly two math majors?
14. Odalys sells eggs to restaurants. Before she sends a package of eggs to a customer, she selects five of the eggs in the package at random and checks to see if they are spoiled. She won't send the package if any of the eggs she tests are spoiled.
 - (a) Suppose the package contains 18 eggs, and half of them are spoiled. How likely is it that Odalys detects a spoiled egg?
 - (b) Suppose that a package contains 144 eggs, and half of them are spoiled. How likely is it that Odalys detects a spoiled egg?
 - (c) Suppose that a package contains 144 eggs, and 10 of them are spoiled. How likely is it that Odalys detects a spoiled egg?
 - (d) What seems to have a bigger effect on the probability of finding a spoiled egg: the size of the package or the percentage of spoiled eggs? Justify your answer.
- *15. A game warden catches 10 fish from a lake, marks them, and returns them to the lake. Three weeks later, the warden catches five fish, and discovers that two of them are marked.
 - (a) Let k be the number of fish in the lake. Find the probability (in terms of k) that two of five randomly selected fish are marked.
 - (b) What value of k will maximize this probability? (This sampling method is a way of estimating the number of fish in the lake.)
- *16. Ten cards are numbered 1 through 10. The cards are shuffled thoroughly and placed in a stack. What is the probability that the numbers on the top three cards are in ascending order?
17. Refer to Example 4.44. If you pick a three-digit number with two repeated digits (e.g., 797), the payouts are more: you win \$660 for an exact match and \$160 if the digits match the winning digits, but in a different order.

248 Chapter 4 ■ Quantitative Thinking

Show that this does not change the expected value of the amount of money you win.

18. An urn contains two red balls and five blue balls.
 - (a) Draw three balls at random from the urn, without replacement. Compute the expected number of red balls in your sample.
 - (b) Draw three balls at random from the urn, with replacement. Compute the expected number of red balls in your sample.
19. An urn contains three red balls, four white balls, and two black balls. Three balls are drawn from the urn at random without replacement. For each red ball drawn, you win \$10, and for each black ball drawn, you lose \$15. Let X represent your net winnings.
 - (a) Compute $P(X = 0)$.
 - (b) Compute $P(X < 0)$.
 - (c) Compute $E(X)$, your expected net winnings.
20. Refer to Exercise 7. Find the expected value of the roll total when rolling a four-sided die and a six-sided die.
21. Conduct the random experiment of flipping a coin five times. Let X be the number of heads.
 - (a) Compute $P(X > 3)$.
 - (b) Compute $E(X)$.
- *22. Conduct the random experiment of flipping a coin until you get heads, or until you have flipped the coin five times. Let X be the number of flips.
 - (a) Compute $P(X > 3)$.
 - (b) Compute $E(X)$.
23. Prove Theorem 4.10.

4.5 Counting Operations in Algorithms

Counting techniques are used in lots of ways: DNA arrangements, risk assessment, and manufacturing optimization, to name a few. In this section we will see how enumeration strategies apply to the study of algorithms. We will learn how to describe some very simple algorithms, and then we will count the operations done by an algorithm. This study will give us a new perspective on the counting problems in the previous sections: any time we can enumerate a set, we can write an algorithm to describe the set. Counting the elements of the set and analyzing the algorithm are fundamentally related.

4.5.1 Algorithms

An *algorithm* is a list of instructions for doing something. We use algorithms all the time; any time we follow an instruction manual or a recipe we are executing a well-defined sequence of operations in order to complete a certain task. Like a good instruction manual, a well-stated algorithm describes precisely what to do.

In mathematics and computer science, algorithms manipulate variables. A *variable* is an object of some type (e.g., integer, string, set) whose value can change. In mathematics, we usually think of variables as unknowns, like the symbol x in the equation $2x + 3 = 11$. In computer science, it is natural to think of variables as storage locations; a program changes the contents of various locations in the memory of a computer.

Algorithms are important for the modern study of almost any technical field. As computers have become commonplace, our understanding of the world has become more discrete. Our music is no longer pressed into the grooves of a vinyl record—it is encoded digitally on CD’s. Biologists can now understand living things by studying discrete genetic patterns. The hundreds of variables that influence the price of a commodity now fit neatly into a spreadsheet. Understanding how these things work requires some familiarity with algorithms. They’re not just for computer scientists anymore.

4.5.2 Pseudocode

In order to discuss algorithms, we need some way of describing them. To avoid having to learn the syntax of a specific programming language (e.g., C++, Java, Scheme), we will give informal descriptions of program operations as *pseudocode*.

The nice thing about pseudocode is that there aren’t very many rules. The variables can represent anything: numbers, strings, lists, or whatever is appropriate, considering the context. We change the values of the program variables using different commands, or *statements*. Note that we are using the word “statement” differently than we did in Chapter 1; in pseudocode, a statement is an instruction that does something, while in logic, a statement is a declarative sentence that is either true or false.

We’ll use the \leftarrow symbol to indicate the *assignment* of one variable to another. For example, the pseudocode statement

$$x \leftarrow y$$

means “set the variable x equal to the value of the variable y .” The arrow suggests the direction that the data is moving; in the context of computers, the value in storage location y is being copied into storage location x . Any preexisting data in x is lost (or “written over”) when this assignment statement is executed, but the value of y doesn’t change.

250 Chapter 4 ■ Quantitative Thinking

We can also use assignment statements to update the value of a single variable. For example, the statement

$$x \leftarrow x + 1$$

means “set x equal to the old value of x plus 1” or, more simply, “increment the value of x .”

In the simple examples that follow, we are going to want our algorithms to report information back to the user. This type of information is called *output*. For our purposes, the `print` statement will suffice; imagine that whatever follows the word `print` gets written to the computer screen. If we `print` a variable, then the output is the value of the variable. If we `print` something enclosed in quotes, then the output consists of the quoted text.

We also want our algorithms to be able to execute certain statements depending on whether some condition holds. For example, the instructions for parking a car might include the following statement.

If the car is facing uphill, then turn the wheels away from the curb.

The `if . . . then` statement is the pseudocode version of this construction. When the statement

```
if <condition> then <statement>
```

executes, the program first checks to see if the `<condition>` is true, and, if it is, the program executes the `<statement>`.

The following example illustrates these simple commands.

Example 4.45 Suppose x is some integer. Consider the following pseudocode statement.

```
print "Old value of x:" x
if x > 5 then x ← x + 3
print "New value of x:" x
```

If initially $x = 10$, then the program will print the following.

```
Old value of x: 10
New value of x: 13
```

However, if instead the initial value of x is 4, the output will be as follows.

```
Old value of x: 4
New value of x: 4
```

In the second case, the condition that the `if . . . then` statement checked ($4 > 5$) was false, so the value of the variable x was not changed.

4.5.3 Sequences of Operations

Example 4.46 Consider the following pseudocode segment. The notation `//` stands for a *comment*—descriptive information that doesn’t do anything.

```

y ← x + x + x + x + x      // line a
z ← y + y + y              // line b

```

First, five copies of x are added together and assigned to y , then three copies of y are added together and assigned to z . Line **a** requires four addition operations and line **b** requires two, so the total number of additions for this segment is 6.

This example is fairly simple, but it will be helpful to generalize it as our first counting principle for algorithms.

Addition Principle for Algorithms. If `statement1` requires m of a certain type of operation and `statement2` requires n operations, then the segment

```

statement1
statement2

```

requires $m + n$ operations.

Example 4.47 We can now revisit Example 4.2 from the perspective of algorithms. Let B be the set of breakfast customers and let L be the set of lunch customers. Let $|B| = 25$ and $|L| = 37$. Consider the following algorithm.

```

Serve everyone in B breakfast.
Serve everyone in L lunch.

```

By the addition principle, 62 meals were served.

4.5.4 Loops

Sometimes an algorithm needs to repeat the same process several times. For example, instructions for preparing to host a party might include the following statement.

```

For each guest that is invited to the party, prepare a name tag with the
guest’s name written on it.

```

The process of “preparing a name tag” must be repeated several times, once for each guest. In pseudocode, a *for-loop* is a convenient way to repeat a statement (or segment of statements) once for each element of some index set I .

Definition 4.4 Let I be a totally ordered finite set. Then the *for-loop*

```
for  $i \in I$  do
  statement $x$ 
```

will execute **statement _{x}** once for each $i \in I$. Each time **statement _{x}** is executed, the value of i moves to the next element in I , according to the total ordering.

For example, the segment

```
for  $i \in \{1, 2, \dots, n\}$  do
  print  $i + i + i$ 
```

will print out the first n multiples of three. Observe that the **print** statement is executed n times, and each time it does two additions, so the total number of additions performed by this segment is $2n$. Thus we see that a loop requires us to multiply when counting operations in algorithms.

Multiplication Principle for Algorithms. If **statement _{x}** requires m of a certain type of operation, then a loop that repeats **statement _{x}** n times requires mn operations.

Example 4.48 The following pseudocode segment adds up the first n natural numbers.

```
 $s \leftarrow 0$ 
for  $i \in \{1, 2, \dots, n\}$  do
   $s \leftarrow s + i$ 
```

Each time through the loop, i gets added to the value of s . The loop runs through the first n natural numbers, so when the loop finishes, s contains the sum of these numbers. This algorithm performs one $+$ operation each time through the loop, so the total number of “ $+$ ” operations is n , by the multiplication principle.

In the multiplication principle for algorithms, **statement _{x}** can be any kind of statement—in particular, it could be a loop. The next example illustrates this situation; the loops in this example are called *nested loops* because one is “nested” inside the other.

Example 4.49 Let $X = A \times B$, where A has m elements and B has n elements. The following pseudocode segment checks to see if $(a, b) \in X$. The \lceil and \lfloor symbols indicate that the bracketed lines together represent the statement that is inside the *for- i -loop*.

```
for  $i \in A$  do
   $\lceil$  for  $j \in B$  do
   $\lfloor$    if  $(i, j) = (a, b)$  then print "Found it."
```

How many comparisons does this segment perform?

Solution: The `if . . . then` statement performs a single comparison using the `=` sign. This is inside the j -loop, which executes n times, and the j -loop is inside the i -loop, which executes m times. Therefore this code segment performs mn comparisons. \diamond

Note that the `for`-loops in this example always run the same number of times, whether or not the pair (a, b) is found in X . This inefficiency points to a limitation of `for`-loops: there is no way to stop them from running for the predetermined number of iterations. A more clever algorithm would stop running once (a, b) is found. In Chapter 5, we'll see a different looping structure—a `while`-loop—with this capability.

There were several counting examples in Section 4.1 that used the addition and multiplication principles for counting sets. We can now revisit these examples from the point of view of algorithms.

Example 4.50 In Example 4.9 we counted the number of strings of at most length 3 that can be formed from a 26-symbol alphabet. Let the set A be this alphabet. The following algorithm prints out all 18,278 of these strings. Recall that if x and y are strings, their concatenation is written as xy .

```

for  $c \in A$  do
  print  $c$ 
for  $c \in A$  do
  for  $d \in A$  do
    print  $cd$ 
for  $c \in A$  do
  for  $d \in A$  do
    for  $e \in A$  do
      print  $cde$ 

```

Applying the addition and multiplication principles for algorithms is analogous to the solution of Example 4.9. The first `for- c` -loop prints one single-symbol string for every symbol in A , so it prints 26 strings. The second `for- c` -loop has another loop inside of it, which runs 26 times, so these nested loops print 26^2 strings. Finally, the third `for- c` -loop has a `for- d` -loop inside, which in turn has a `for e` loop inside of it. This nested trio therefore prints 26^3 strings. These three looping statements run in sequence, so by the addition principle, the total number of strings printed is $26 + 26^2 + 26^3 = 18,278$.

Example 4.51 In Example 4.6, we used a decision tree to determine that there are 12 designs of the form



254 Chapter 4 ■ Quantitative Thinking

where each square is colored either red, green, or blue, and no two adjacent squares are the same color. The following algorithm prints out strings in the symbols $A = \{\mathbf{R}, \mathbf{G}, \mathbf{B}\}$ corresponding to these 12 designs. Recall that the set difference $U \setminus X$ is another way of writing $U \cap X'$.

```

for  $c \in A$  do
  for  $d \in A \setminus \{c\}$  do
    for  $e \in A \setminus \{d\}$  do
      print  $cde$ 

```

These loops are very similar to the nested loops in Example 4.50. The only difference is that the index sets are modified so that d is drawn from a set that does not contain c , and likewise e comes from a set without d . This ensures that no two adjacent squares will be the same color. It also makes it easy to count the number of strings printed using the multiplication principle: $3 \cdot 2 \cdot 2$.

The following example first appeared in Section 4.2 as a counting problem. Let's revisit this example from an algorithmic point of view.

Example 4.52 Yifan has 26 refrigerator magnets in the shapes of the letters from A to Z. Write an algorithm to print all the different three-letter strings he can form with these letters.

Solution: The following algorithm will print out all the strings Yifan can form using his refrigerator magnets. The algorithm is very similar to the one used in Example 4.51. Let $A = \{\mathbf{A}, \mathbf{B}, \dots, \mathbf{Z}\}$.

```

for  $c \in A$  do
  for  $d \in A \setminus \{c\}$  do
    for  $e \in A \setminus \{c, d\}$  do
      print  $cde$ 

```

In each successive loop, the index sets get smaller by one element; this ensures that no character is repeated. The sizes of the index sets are 26, 25, and 24, so by the multiplication principle for algorithms, $26 \cdot 25 \cdot 24$ strings are printed. \diamond

4.5.5 Arrays

A for-loop is a good tool for running through every item in a list of data. The following definition gives a way of representing a list in pseudocode.

Definition 4.5 An *array* is a sequence of variables $x_1, x_2, x_3, \dots, x_n$.

An array is simply a notational convenience for a list of variables of the same type. Sometimes it is helpful to think of an array as a contiguous block

of storage locations in a computer. For example, an array of ten real numbers might look like this:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
2.4	7.3	3.1	2.7	1.1	2.4	8.2	0.3	4.9	7.3

Notice that the order of the elements in an array matters, and an array can have duplicate entries.

Example 4.53 The following algorithm counts the number of duplicates in the array $x_1, x_2, x_3, \dots, x_n$, where $n > 1$.

```

t ← 0
for i ∈ {1, 2, 3, ..., n - 1} do
  ⌈ for j ∈ {i + 1, ..., n} do
    ⊥   if xi = xj then t ← t + 1

```

The variable t serves as a *counter*; its value gets incremented each time a duplicate is found. The loop indices are set up so that x_i always comes before x_j in the array. This avoids comparing the same pair of array elements more than once, while ensuring that every pair of elements gets compared.

The size of $\{i + 1, \dots, n\}$, the index set for j , depends on i . Therefore we can't use the multiplication principle to count the number of “=” comparisons, because the number of comparisons is different each time through the *for- i* -loop. The following table shows how the variables change in the case where $n = 4$ and the array has values $x_1 = 20$, $x_2 = 50$, $x_3 = 50$, and $x_4 = 18$.

x_1	x_2	x_3	x_4	t	i	j	Comparison
20	50	50	18	0	1	2	$20 \stackrel{?}{=} 50$
20	50	50	18	0	1	3	$20 \stackrel{?}{=} 50$
20	50	50	18	0	1	4	$20 \stackrel{?}{=} 18$
20	50	50	18	0	2	3	$50 \stackrel{?}{=} 50$
20	50	50	18	1	2	4	$50 \stackrel{?}{=} 18$
20	50	50	18	1	3	4	$50 \stackrel{?}{=} 18$

Such a table is called a *trace* of an algorithm. Notice that the only time that the counter t was incremented was after the two duplicates, x_2 and x_3 , were compared. Also notice that each pair of array elements was compared exactly once. The number of comparisons is evidently $6 = 3 + 2 + 1$ in this case. In general, this algorithm will make

$$(n - 1) + (n - 2) + \dots + 2 + 1$$

comparisons, by the addition principle.

Another way to count the number of comparisons is to notice that one comparison is done for each pair (x_i, x_j) , where $i < j$. By the reasoning in Exercise 7 of Section 4.3, the number of such pairs is the same as the number of ways to choose a set of two elements from the set $\{1, 2, \dots, n\}$, namely $C(n, 2)$. Recall that

$$C(n, 2) = \frac{n(n-1)}{2},$$

which indeed equals $(n-1) + (n-2) + \dots + 2 + 1$, by Exercise 2 of Section 3.2.

Example 4.53 works no matter what type of data is stored in the array x_1, x_2, \dots, x_n —these variables could represent images, or sets, or people. But often the elements of an array are elements of a set on which some total ordering is defined, like the integers (\leq) or English words (alphabetical order). In this situation it is possible to find the maximum item in the list with respect to the total ordering.

Example 4.54 Let x_1, x_2, \dots, x_n be an array whose elements can be compared by the total ordering \leq . Write an algorithm for computing the maximum element in the array. How many “<” comparisons does your algorithm require?
Solution: The natural way to find the maximum element is to go through the list and keep track of the largest element as we go.

```

m ← x1
for i ∈ {2, 3, ..., n} do
    if m < xi then m ← xi

```

The index set has $n - 1$ elements, so the algorithm makes $n - 1$ “<” comparisons, by the multiplication rule. \diamond

4.5.6 Sorting

Data is almost always easier to use if it is organized. For example, if we were working with an array containing a list of names, we would usually prefer to have the names listed alphabetically. In general, if the elements of an array can be compared by \leq , it is important to be able to rearrange the elements so that they are in order. A *sort* is an algorithm that guarantees that

$$x_1 \leq x_2 \leq x_3 \leq \dots \leq x_n$$

after the algorithm finishes. The next example gives a very simple algorithm for sorting an array.

Example 4.55 Let $x_1, x_2, x_3, \dots, x_n$ be an array whose elements can be compared by \leq . The following algorithm is called a *bubble sort*.

```

for  $i \in \{1, 2, \dots, n-1\}$  do
  for  $j \in \{1, 2, \dots, n-i\}$  do
    if  $x_j > x_{j+1}$  then swap  $x_j$  and  $x_{j+1}$ 

```

Step through this algorithm for the following list of four elements: $x_1 = 9$, $x_2 = 4$, $x_3 = 7$, and $x_4 = 1$.

Solution: The following table shows how the program variables change during execution. Each row of the table gives the value of the program variables before the if statement is executed. The last two columns show the result of each “>” comparison.

x_1	x_2	x_3	x_4	i	j	Comparison	Result
9	4	7	1	1	1	$x_1 > x_2$	yes
4	9	7	1	1	2	$x_2 > x_3$	yes
4	7	9	1	1	3	$x_3 > x_4$	yes
4	7	1	9	2	1	$x_1 > x_2$	no
4	7	1	9	2	2	$x_2 > x_3$	yes
4	1	7	9	3	1	$x_1 > x_2$	yes
1	4	7	9				

After the last comparison and swap, the loops are finished, and the list is in order. \diamond

The bubble sort is so named because the largest elements tend to “bubble” up to the end of the list, like the bubbles in a soft drink. Look back at the above trace and notice that the 9 bubbled to the end of the list when i was 1, then the 7 bubbled up next to it when i was 2, and so on.

Example 4.56 How many times does the bubble sort make the “>” comparison when sorting a list of n elements?

Solution: The if statement requires one comparison. The outside loop makes the inside loop execute $n - 1$ times, but each time the size of the index set for j gets smaller. Thus, the total number of comparisons is

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

as in Example 4.53. \diamond

Exercises 4.5

1. Look at pseudocode segment in Example 4.46. If $x = 3$ before this segment is executed, what is the value of z after execution?
2. Modify Example 4.48 so that it adds up the first n natural numbers using only $n - 1$ “+” operations. You may assume that $n \geq 1$.
3. Trace through the algorithm in Example 4.54 in the case when $n = 5$ and the array elements are $x_1 = 77$, $x_2 = 54$, $x_3 = 95$, $x_4 = 101$, and $x_5 = 62$.
4. Let x and n be integers greater than 1. Consider the following algorithm.

```

t ← 0, s ← 0
for i ∈ {1, 2, ..., x} do
  ⌈ for j ∈ {1, 2, ..., n} do
    t ← t + x
  ⊥ s ← s + t           // line A

```

The \lceil and \lfloor symbols and the indentation are important: they tell you which lines are inside which loop. So, for example, line A is inside the i -loop, but outside the j -loop.

- (a) If $x = 3$ and $n = 5$ initially, what is the value of s after this segment executes?
 - (b) Count the number of additions this segment performs. Your answer should be in terms of x and n .
5. How many words does each algorithm print? Explain your answers. Again, the indenting is important.

- (a)

```

for i ∈ {1, 2, ..., 9} do
  ⌈ for j ∈ {1, 2, ..., 6} do
    ⌈ for k ∈ {1, 2, 3} do
      ⊥ ⊥ print "Cubs Win"

```
- (b)

```

for i ∈ {1, 2, ..., 9} do
  ⌈ for j ∈ {1, 2, ..., 6} do
    print "Sox Win"
    for k ∈ {1, 2, 3} do
      ⊥ print "Sox Win"

```

6. Let $n > 1$. Consider the following pseudocode segment.

```

for  $i \in \{1, 2, \dots, 10\}$  do
  ▮ statementA
    for  $j \in \{1, 2, \dots, n\}$  do
      statementB
      for  $k \in \{1, 2, 3, 4\}$  do
        ▮ for  $l \in \{1, 2, \dots, n\}$  do
          ▮ statementC

```

- Which statement (A , B , or C) gets executed the most number of times?
- Suppose that statement A requires $3n$ comparison operations, B requires n^2 comparisons, and C requires 30 comparisons. How many total comparisons does the entire pseudocode segment require?

7. Consider the following pseudocode segment.

```

 $x \leftarrow 3$ 
for  $i \in \{1, 2, \dots, n\}$  do
  ▮ for  $j \in \{1, 2, \dots, n\}$  do
     $x \leftarrow x + 5$ 
    for  $k \in \{1, 2, 3, 4, 5\}$  do
      ▮  $x \leftarrow x + k + 1$ 

```

- How many times is the “+” operation executed?
- What is the value of x after this segment runs?

8. Consider the following algorithm.

```

 $x \leftarrow 1$ 
for  $i \in \{1, 2, 3\}$  do
  ▮ for  $j \in \{1, 2, 3, 4\}$  do
     $x \leftarrow x + x$ 
    for  $k \in \{1, 2, 3, 4, 5\}$  do
      ▮  $x \leftarrow x + 1$ 
      ▮  $x \leftarrow x + 5$ 

```

- Count the number of + operations done by this algorithm.
- What is the value of x after the algorithm finishes?

9. Let $n > 3$. Consider the following pseudocode segment.

```

for  $i \in \{1, 2, \dots, n - 3\}$  do
   $w \leftarrow w + z + 10$ 
  for  $j \in \{1, 2, \dots, n\}$  do
     $z \leftarrow y + y + y + 30$ 
    for  $k \in \{1, 2, \dots, n^2\}$  do
      for  $l \in \{1, 2, \dots, n\}$  do
         $y \leftarrow x + l$ 
        for  $m \in \{1, 2, \dots, n\}$  do
           $x \leftarrow w + y + z$ 

```

How many “+” operations does this algorithm perform?

10. Interpret the following algorithm in the context of rolling two six-sided dice. What do the counters e and s count?

```

 $s \leftarrow 0$ 
 $e \leftarrow 0$ 
for  $i \in \{1, 2, 3, 4, 5, 6\}$  do
  for  $j \in \{1, 2, 3, 4, 5, 6\}$  do
    if  $i + j = 7$  then  $e \leftarrow e + 1$ 
   $s \leftarrow s + 1$ 

```

11. Write a pseudocode algorithm that runs through all possible outcomes when a four-sided die and a six-sided die are cast and prints all the ways of having a roll total of 8.
12. Write a pseudocode algorithm to compute the product of the first n positive integers. How many multiplications does your algorithm perform?
13. Write an algorithm in pseudocode that will print out all possible Illinois license plates, according to the description in Example 4.10.
14. Write an algorithm in pseudocode that will print out the first 500,000 license plates in a given district in India, according to the description in Exercise 6 of Section 4.1.
15. Trace through the bubble sort algorithm for the following data set: $x_1 = 5$, $x_2 = 4$, $x_3 = 2$, $x_4 = 1$, $x_5 = 3$.
16. The number of **swap** statements executed by a bubble sort varies depending on the initial state of the array. Under what circumstances will the bubble sort make zero **swaps**?

17. Write a pseudocode algorithm to compute the following sum.

$$1 + 1 \cdot 2 + 1 \cdot 2 \cdot 3 + \cdots + 1 \cdot 2 \cdots n$$

How many multiplications does your algorithm perform?

18. Let x_1, x_2, \dots, x_n be an array. Consider the following algorithm.

```

for  $i \in \{1, 2, \dots, \lfloor n/2 \rfloor\}$  do
   $t \leftarrow x_i$ 
   $x_i \leftarrow x_{n-i+1}$ 
   $x_{n-i+1} \leftarrow t$ 

```

- (a) How many “ \leftarrow ” operations does this algorithm perform?
- (b) What does this algorithm do to the array?
19. Let x_1, x_2, \dots, x_n be an array of integers. Write a pseudocode algorithm that will compute the probability that a randomly chosen element of this array is odd. (You may use a statement like “if k is odd then ...” in your algorithm.)
20. Write a pseudocode algorithm that will print out all three-letter palindromes with symbols from the set $A = \{A, B, \dots, Z\}$. How many such palindromes are there?
21. Write a pseudocode algorithm that will print out all strings of four symbols from the set $A = \{A, B, \dots, Z\}$ such that no symbol is repeated. How many such strings are there?
22. Recall Example 4.11. Write a pseudocode algorithm that prints out all allowable colorings of the vertices a, b, c , and d as a four-symbol string using the symbols in $C = \{R, G, B, V\}$. Use the two disjoint cases given in the solution to Example 4.11: when b and d are the same color, and when b and d are different colors.
23. Repeat the previous problem using the three disjoint cases suggested in Exercise 12 of Section 4.1: using two different colors, using three different colors, and using four different colors.

4.6 Estimation

So far, we have usually been able to come up with an exact answer for counting problems. Some of our examples, such as rearranging the letters in a word or drawing balls from an urn, might seem somewhat distant from real-world situations. However, in many applications—and especially in algorithms—

enumeration problems are too complicated to consider all the cases and get an exact count. Often, all we need is an estimate. In this section, we will practice estimating answers to counting problems, concentrating on the kind of answers that help solve discrete problems in computer science.

4.6.1 Growth of Functions

In most discrete counting problems, the answer depends on some natural number n . We have seen several examples: the number of binary strings with n digits, the number of ways to choose three things from a set of n things, the number of socks needed to guarantee getting n of the same kind. The answer to these problems is a function

$$f: \mathbf{N} \longrightarrow \mathbf{R}$$

This function usually maps into the integers, but for the purposes of estimation we use the codomain \mathbf{R} to allow for the possibility of noninteger values. Sometimes we will also think of f as being defined on the domain \mathbf{R} instead of \mathbf{N} , but for most discrete applications, we only evaluate f at integer values.

Example 4.57 Suppose that a certain networking algorithm must run through all the different possible groups of three computers (i.e., subsets of size 3) in a network of n computers. Give the number of groups as a function of n .

Solution: Let $f(n)$ be the number of subsets of three computers in a network of n computers. Using the Selection Principle, there are

$$f(n) = \binom{n}{3} = \frac{n!}{3!(n-3)!} = \frac{n(n-1)(n-2)}{3!}$$

different subsets of size 3. ◇

Example 4.58 Suppose that a different networking algorithm must run through all the different possible (unordered) pairs of computers in a network of n computers. Give the number of pairs as a function of n .

Solution: Let $g(n)$ be the number of subsets of two computers in a network of n computers. Using the Selection Principle, there are

$$g(n) = \binom{n}{2} = \frac{n!}{2!(n-2)!} = \frac{n(n-1)}{2}$$

different subsets of size 2. ◇

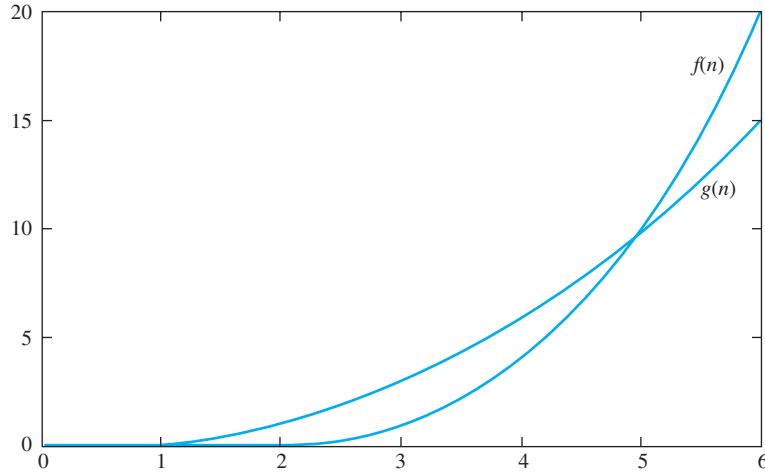


Figure 4.12 Plots of $f(n)$ and $g(n)$ for $0 \leq n \leq 6$.

How do these two algorithms compare? More specifically, which one will run faster? Assuming that the running time depends mainly on the number of groups each algorithm must run through, the functions $f(n)$ and $g(n)$ should indicate the relative speed of the two algorithms. Figure 4.12 shows a plot of $f(n)$ and $g(n)$ vs. n , for $0 \leq n \leq 6$. The two curves are fairly similar, suggesting that the running times of these two algorithms should be comparable for small networks.

However, the story is quite different for larger networks. Figure 4.13 shows a plot of the same two functions over the interval $0 \leq n \leq 50$. Notice that the $f(n)$ curve grows much more steeply than $g(n)$ over this range. Therefore, on larger networks, we would expect the algorithm on triples to run noticeably slower than the algorithm on pairs.

The algorithms in Examples 4.57 and 4.58 illustrate a phenomenon that should seem plausible: differences in algorithm performance will be more noticeable when the algorithms are applied to big problems. For this reason, it is important to have a way of classifying and ranking functions of n based on their behavior for large values of n . This is the idea behind big- \mathcal{O} (“big-oh”), big- Ω (“big-omega”), and big- Θ (“big-theta”) notation.

For the remainder of this section, let’s assume for simplicity that all the functions we consider map from domain \mathbf{N} to codomain \mathbf{R}^+ , the positive real numbers, unless noted otherwise.

264 Chapter 4 ■ Quantitative Thinking

Definition 4.6 Let $f: \mathbf{N} \rightarrow \mathbf{R}^+$ be a function. Then $\mathcal{O}(f)$ is the set of all functions g such that

$$g(n) \leq Kf(n)$$

for some constant $K > 0$, and for all $n \geq N$ for some $N > 0$. If $g \in \mathcal{O}(f)$, we also say that “ g is big-oh of f .”

In other words, $\mathcal{O}(f)$ is the set of all functions that are bounded above by some constant multiple of $f(n)$ for large values of n . Figure 4.13 illustrates this situation graphically; the following example shows how to apply the definition algebraically.

Example 4.59 Show that $g \in \mathcal{O}(f)$, where

$$f(n) = \frac{n(n-1)(n-2)}{6} \quad \text{and} \quad g(n) = \frac{n(n-1)}{2}.$$

Solution: By Definition 4.6, we must show that there are constants K and N such that $g(n) \leq Kf(n)$ for all $n \geq N$. In order to establish this inequality, we

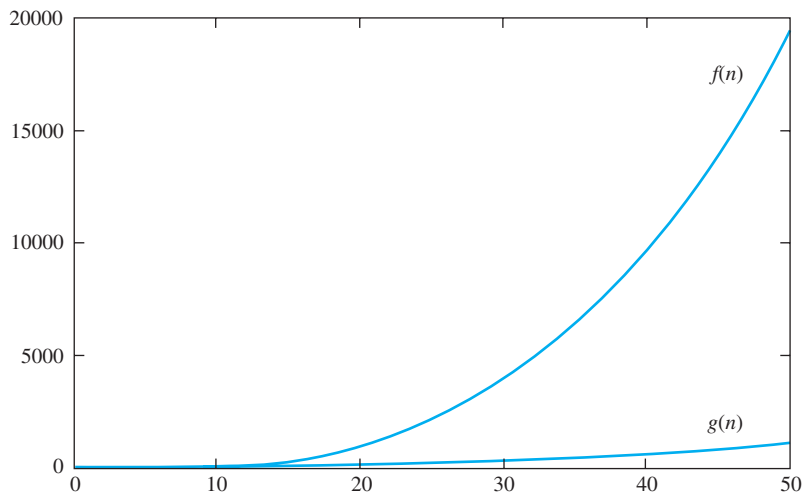


Figure 4.13 Plots of $f(n)$ and $g(n)$ for $0 \leq n \leq 50$.

construct a sequence of (in)equalities as follows:

$$\begin{aligned} g(n) &= \frac{n(n-1)}{2} \\ &\leq \frac{n(n-1)(n-2)}{2} \quad \text{for } n \geq 3 \\ &= 3 \cdot \frac{n(n-1)(n-2)}{6}. \end{aligned}$$

So $g(n) \leq 3f(n)$ for all $n \geq 3$, as required. \diamond

Notice that the second line of the above sequence holds because $n-2 \geq 1$ for $n \geq 3$. In general, to establish a “ \leq ” inequality, start with an equation you know to be true, and think of ways to make the right-hand side bigger.

In addition to big- \mathcal{O} , there is a similar definition for functions that are bounded below by a multiple of f .

Definition 4.7 Let $f: \mathbf{N} \rightarrow \mathbf{R}^+$ be a function. Then $\Omega(f)$ is the set of all functions g such that

$$g(n) \geq Kf(n)$$

for some constant $K > 0$, and for all $n \geq N$ for some $N > 0$. If $g \in \Omega(f)$, we also say that “ g is big- ω of f .”

Think of big- \mathcal{O} and big- Ω as ways of comparing the long-term behavior of functions. They both stipulate that one function is at least as big as the other, up to a constant multiple K , for sufficiently large values of n . The two definitions are symmetric, as the following theorem shows.

Theorem 4.11 Let $f, g: \mathbf{N} \rightarrow \mathbf{R}^+$. Then $f \in \mathcal{O}(g) \Leftrightarrow g \in \Omega(f)$.

Proof Suppose $f \in \mathcal{O}(g)$. Then there exist positive numbers K and N such that $f(n) \leq Kg(n)$ for all $n \geq N$. Let $K' = 1/K$. Then for all $n \geq N$, $g(n) \geq K'f(n)$, so $g \in \Omega(f)$. The proof of the converse is almost the same and is left as an exercise. \square

Big- Θ notation combines big- \mathcal{O} and big- Ω to form an equivalence relation on the set of all functions $\mathbf{N} \rightarrow \mathbf{R}^+$.

Definition 4.8 Let $f: \mathbf{N} \rightarrow \mathbf{R}^+$ be a function. The *big-theta class* $\Theta(f)$ is the set of all functions g such that

$$K_1f(n) \leq g(n) \leq K_2f(n)$$

for some positive constants K_1, K_2 , and for all $n \geq N$ for some $N > 0$. In other words, $\Theta(f) = \mathcal{O}(f) \cap \Omega(f)$. If $g \in \Theta(f)$, we also say that “ g is big-theta of f ” or “ g is order f .”

Observe that $f \in \Theta(g)$ if and only if $g \in \Theta(f)$. In this case we say that f and g are of the same order, and we often choose to write $\Theta(f) = \Theta(g)$. Functions of the same order grow in roughly the same way as n increases. For the purposes of estimation, big- Θ encapsulates just enough information to make useful comparisons between functions.²

Example 4.60 See Example 4.55. Show that the number of comparisons required to do a bubble sort on a list of n items is in $\Theta(n^2)$.

Solution: In Example 4.56, we calculated that the number of comparisons performed by a bubble sort is $n(n-1)/2$. Since

$$\begin{aligned} \frac{n(n-1)}{2} &= \frac{1}{2} \cdot (n^2 - n) \\ &\leq \frac{1}{2} \cdot n^2 \quad \text{for } n \geq 1, \end{aligned}$$

we conclude that $n(n-1)/2 \in \mathcal{O}(n^2)$. Furthermore, the calculation

$$\begin{aligned} \frac{n(n-1)}{2} &= \frac{1}{2} \cdot (n^2 - n) \\ &\geq \frac{1}{2} \cdot \left(n^2 - \frac{n^2}{2} \right) \quad \text{for } n \geq 2 \\ &= \frac{1}{4} \cdot n^2 \end{aligned}$$

shows that $n(n-1)/2 \in \Omega(n^2)$. Therefore, $n(n-1)/2 \in \Theta(n^2)$. \diamond

In other words, using estimation, we say that “the number of comparisons done by the bubble sort is order n^2 .”

4.6.2 Estimation Targets

It is an exercise for you to show that big- Θ defines an equivalence relation on the set of all functions $\mathbf{N} \rightarrow \mathbf{R}^+$. The Θ -classes are the equivalence classes determined by this relation. The next theorem lists some commonly used equivalence class representatives.

² Unfortunately, people often confuse big- \mathcal{O} with big- Θ . Many practitioners will read “ $f \in \mathcal{O}(g)$ ” as “ f is order g ” or write “ $f \in \mathcal{O}(g)$ ” when they mean “ $f \in \Theta(g)$.” Beware. It is also common to see the notation “ $f = \Theta(g)$ ” in place of “ $f \in \Theta(g)$.”

Theorem 4.12 *The following functions of n represent different Θ -classes. Furthermore, the functions are listed according to how fast they grow: if f comes before g on the list, then $f \in \mathcal{O}(g)$.*

1. 1, the constant function $f(n) = 1$.
2. $\log_2 n$
3. n^p for $0 < p < 1$.
4. n
5. $n \log_2 n$
6. n^p for $1 < p < \infty$.
7. 2^n
8. $n!$

Cases (3) and (6) represent continuums of Θ -classes: if $0 < p < q$, then $n^p \in \mathcal{O}(n^q)$ and $\Theta(n^p) \neq \Theta(n^q)$.

We'll skip proving this theorem, but you will find some parts of the proof left as exercises.

The functions 1, $\log_2 n$, n , n^p , $n \log_2 n$, 2^n , $n!$ are called *estimation targets*. When we want to estimate the growth of a function, we will try to compare it to one of these target functions.

Example 4.61 Give a big- Θ estimate of $\log_2 n^{7n+1}$.

Solution: Using the identity $\log_b(a^k) = k \log_b a$, we obtain

$$\begin{aligned} \log_2 n^{7n+1} &= (7n+1) \log_2 n \\ &\leq 8n \log_2 n, \text{ for } n \geq 1 \end{aligned}$$

and similarly,

$$\begin{aligned} \log_2 n^{7n+1} &= (7n+1) \log_2 n \\ &\geq 7n \log_2 n, \text{ for } n \geq 1, \end{aligned}$$

so $\log_2 n^{7n+1} \in \Theta(n \log_2 n)$. ◇

Two functions in the same Θ -class grow at approximately the same rate, for large values of n . If these two functions describe the size of two tasks (in terms of input size n), then these tasks will “scale” up approximately equally well. We will explore this idea more carefully when we study the complexity of discrete processes in Chapter 5.

4.6.3 Properties of Big- Θ

There are several properties of Θ -classes that help make finding big- Θ estimates easier. The first property is the observation that constant multiples don't change the Θ -class.

Theorem 4.13 *Let $f: \mathbf{N} \rightarrow \mathbf{R}^+$ be a function, and let $k > 0$ be a constant. Then $kf(n) \in \Theta(f(n))$.*

Proof Exercise. □

The next theorem says that the big- Θ estimate of a sum is determined by the biggest summand.

Theorem 4.14 *Let $f, g: \mathbf{N} \rightarrow \mathbf{R}^+$ be functions, and suppose that $g(n) \in \mathcal{O}(f(n))$. Then $f(n) + g(n) \in \Theta(f(n))$.*

Proof Since g takes only positive values, $f(n) + g(n) \geq f(n)$, so $f(n) + g(n) \in \Omega(f(n))$. Since $g(n) \in \mathcal{O}(f(n))$, there are positive constants K and N such that

$$g(n) \leq Kf(n)$$

for all $n \geq N$. We can assume $K \geq 1$, for if it weren't, we could increase it, and the inequality $g(n) \leq Kf(n)$ would still hold. Therefore, $f(n) \leq Kf(n)$, and

$$f(n) + g(n) \leq Kf(n) + Kf(n) = 2Kf(n),$$

so $f(n) + g(n) \in \mathcal{O}(f(n))$. Therefore, $f(n) + g(n) \in \Theta(f(n))$. □

This theorem is a mathematical version of a rule from chemistry. In a chemical reaction involving several different stages, the rate-determining step is the slowest step. The same thing happens when we estimate the growth of functions.

A consequence of Theorem 4.14 is that the Θ -class of a polynomial is determined by its highest-degree term.

Corollary 4.4 *Let $f(n) = a_0 + a_1n + a_2n^2 + \cdots + a_pn^p$ and suppose all of the coefficients $a_i \geq 0$ and $a_p > 0$. Then $f \in \Theta(n^p)$.*

Actually, a stronger statement holds: the coefficients a_0, a_1, \dots, a_{p-1} need not be positive. It is left as an exercise to verify this fact.

Example 4.62 Give a big- Θ estimate of the number of strings with ten or fewer letters that can be formed using n symbols if no symbol can be repeated.

Solution: Not counting the empty string, there are 10 cases to consider, and they are all arrangement problems. So we need an estimate for

$$f(n) = P(n, 1) + P(n, 2) + \cdots + P(10, n).$$

Each term is a polynomial in n ; as a product:

$$P(n, r) = n(n-1)(n-2) \cdots (n-r+1).$$

There are r factors in this product, so the highest-degree term in $P(n, r)$ is n^r . Therefore, the highest-degree term in the polynomial $f(n)$ is n^{10} , so $f(n) \in \Theta(n^{10})$. \diamond

Exercises 4.6

1. Give a big- Θ estimate of $\log_2(n^3)$ in terms of an estimation target. Use Definition 4.8 to justify your answer. (Hint: $\log_b(a^k) = k \log_b a$.)
2. Let $b > 1$. Show that $\log_b n \in \Theta(\log_2 n)$. (Hint: $\log_b n = \frac{\log_2 n}{\log_2 b}$.)
3. Find a big- Θ estimate for each function using an estimation target.
 - (a) $3n^5 + 4n^2 + 17$.
 - (b) $(12n + 17)^{23}$.
 - (c) $n \log_2 n + n!$.
 - (d) $n \log_2 n + n$.
 - (e) $\log_2(n^{10})$.
4. Let $f_1, g_1, f_2, g_2: \mathbf{N} \rightarrow \mathbf{R}^+$ be functions such that $g_1 \in \Theta(f_1)$ and $g_2 \in \Theta(f_2)$. Use Definition 4.8 to verify the following identities.
 - (a) $g_1 + g_2 \in \Theta(f_1 + f_2)$.
 - (b) $g_1 \cdot g_2 \in \Theta(f_1 \cdot f_2)$.

This exercise shows that big- Θ notation respects addition and multiplication: you can multiply (or add) first, and then estimate, or you can estimate first, and then multiply (or add).

270 Chapter 4 ■ Quantitative Thinking

5. Use the fact that big- Θ notation respects multiplication to give big- Θ estimates of the following. First estimate each factor, then multiply. Use estimation targets when possible.

(a) $(10n + 100) \log_{10} n$.

(b) $(4\sqrt{n} + 1)(\sqrt{n} + 10)$.

(c) $(7n^3 + 3n^2 + 2n + 9)(9n^7 + 3n^5 + 2n + 4)$.

(d) $(3n^5 + 7n^8) \log_2 n^3$.

6. Let $p, q \in \mathbf{N}$ with $0 < p < q$. Show that $n^p \in \mathcal{O}(n^q)$ using Definition 4.6.

7. Let $m, b > 0$. Use Definition 4.6 to show that the linear function $l(n) = mn + b$ is in $\Theta(n)$.

8. Prove that $2^{n-1} \in \Theta(2^n)$.

9. Show that $2^n \in \mathcal{O}(10^n)$.

10. In this exercise, we show that $2^n \notin \Omega(10^n)$. Suppose, to the contrary, that $2^n \in \Omega(10^n)$. By Definition 4.7, there are positive constants K and N such that $2^n \geq K \cdot 10^n$ for $n \geq N$. Therefore

$$\frac{1}{K} \geq 5^n$$

for all $n \geq N$. Explain why this is a contradiction.

11. Show that $2^n \in \mathcal{O}(n!)$.

12. Prove by contradiction that $2^n \notin \Omega(n!)$.

13. Finish the proof of the assertion at the end of Theorem 4.12 by showing that $\Theta(n^p) \neq \Theta(n^q)$ for $0 < p < q$. (Hint: Use a proof by contradiction: suppose, to the contrary, that $n^p \in \Omega(n^q)$.)

14. Let $k > 0$ be a constant. Think of k as a constant function $f(n) = k$. Show that $k \in \Theta(1)$.

15. Prove Theorem 4.13

16. Prove Corollary 4.4.

*17. Consider Corollary 4.4. Show that the result is still true even if negative coefficients are allowed, that is, if we remove the restriction that $a_i \geq 0$ for $i = 1, 2, \dots, p - 1$.

18. Let X be a set with n elements, $n > 3$. Determine the sizes of each of the following sets in terms of n , and give a big- Θ estimate for each answer.

(a) $\mathcal{P}(X)$ the power set of X .

- (b) The set of all one-to-one correspondences $X \rightarrow X$.
- (c) The set of all subsets of X of size 3.
- (d) The set of all strings x_1x_2 , with $x_i \in X$.
19. Find a big- Θ estimate for the number ways to choose a set of five or fewer elements from a set of size n .
20. Let $n > 1$. Consider the following pseudocode segment.

```

w ← 1
for i ∈ {1, 2, ..., n} do
  ⌈ w ← w · 10
  for j ∈ {1, 2, ..., 2n} do
    w ← w · w
  for k ∈ {1, 2, ..., n5} do
    ⌈ for l ∈ {1, 2, ..., n} do
      ⌊         ⌊ w ← l · w

```

- (a) How many multiplications does this algorithm perform? Show how you arrive at your answer.
- (b) Give a big- Θ estimate for your answer to part (a). Use one of the estimation targets.
21. A certain algorithm processes a list of n elements. Suppose that **Subroutine_a** requires $n^2 + 2n$ operations and **Subroutine_b** requires $3n^3 + 7$ operations. Give a big- Θ estimate for the number of operations performed by the following pseudocode segment.

```

for i ∈ {1, 2, ..., n} do
  Subroutinea
  Subroutineb

```

22. Consider a list of n names. Suppose that an algorithm on this list consists of the following tasks, and the big- Θ estimate of the number of operations in each step is as shown in the table.

	Task	Big- Θ
1.	Rearrange each name, putting last name first.	n
2.	Sort the list of names.	$n \log_2 n$
3.	Search the list for “Knuth, Donald.”	$\log_2 n$

Give a big- Θ estimate for the number of operations performed by the entire algorithm. Justify your answer with a theorem from this section.

23. Finish the proof of Theorem 4.11 by proving that $g \in \Omega(f) \Rightarrow f \in \mathcal{O}(g)$.

272 Chapter 4 ■ Quantitative Thinking

24. Explain why the relation on functions $\mathbf{N} \rightarrow \mathbf{R}^+$ defined by

$$f R g \Leftrightarrow f \in \mathcal{O}(g)$$

is not a partial ordering.

25. Explain why the relation on functions $\mathbf{N} \rightarrow \mathbf{R}^+$ defined by

$$f R g \Leftrightarrow f \in \mathcal{O}(g)$$

is not an equivalence relation.

26. Show that the relation on functions $\mathbf{N} \rightarrow \mathbf{R}^+$ defined by

$$f R g \Leftrightarrow f \in \Theta(g)$$

is an equivalence relation.