

1

Counting

Stein
et al.

1.1 BASIC COUNTING

The Sum Principle

In this book, we introduce ideas through exercises. Trying to figure out how to do each exercise will help you understand the explanation that follows. Our first exercise illustrates the sum principle.

Exercise 1.1-1

The following loop is part of an implementation of selection sort, which sorts a list of items chosen from an ordered set (numbers, alphabet characters, words, etc.) into nondecreasing order.

```
(1) for  $i = 1$  to  $n - 1$ 
(2)     for  $j = i + 1$  to  $n$ 
(3)         if ( $A[i] > A[j]$ )
(4)             exchange  $A[i]$  and  $A[j]$ 
```

How many times is the comparison $A[i] > A[j]$ made in Line 3?

In Exercise 1.1-1, the segment of code from Lines 2–4 is executed $n - 1$ times, once for each value of i between 1 and $n - 1$, inclusive. The first time, it makes $n - 1$ comparisons. The second time, it makes $n - 2$ comparisons. The i th time, it makes $n - i$ comparisons. Thus, the total number of comparisons is

$$(n - 1) + (n - 2) + \cdots + 1. \quad (1.1)$$

This formula is not as important as the reasoning that led to it. To put the reasoning into a broadly applicable format, we use the language of sets to describe what we are doing. Think about the set S containing all comparisons made by the algorithm in Exercise 1.1-1, in which we divided set S into $n - 1$ pieces (i.e., smaller sets): the set S_1 of comparisons made when $i = 1$, the set S_2 of comparisons made when $i = 2$, and so on through

the set S_{n-1} of comparisons made when $i = n - 1$. We figured out the number of comparisons in each piece and then added the sizes of all the pieces to get the size of the set of all comparisons.

To describe a general version of this process, we now introduce some set-theoretic terminology. Two sets are called **disjoint** when they have no elements in common. Each set S_i described above is disjoint from each of the others because the comparisons made for one value of i are different from those made for another value of i . We say that the set of sets $\{S_1, \dots, S_m\}$ (above, m was $n - 1$) is a family of **mutually disjoint sets**, to mean that it is a family (set) of sets, any two of which are disjoint. With this language, we can state a general principle that explains what we did without making any specific reference to the problem we solved.

Principle 1.1 (Sum Principle)

The size of a union of a family of mutually disjoint finite sets is the sum of the sizes of the sets.

Thus, in effect, we used the sum principle to solve Exercise 1.1-1. We can also describe the sum principle using an algebraic notation. Let $|S|$ denote the size of the set S . For example, $|\{a, b, c\}| = 3$ and $|\{a, b, a\}| = 2$.¹ Using this notation, we can state the sum principle as follows. If $S_1, S_2, \dots, S_m$ are disjoint sets, then

$$|S_1 \cup S_2 \cup \dots \cup S_m| = |S_1| + |S_2| + \dots + |S_m|. \quad (1.2)$$

We can also use the standard notation for union, as follows, to avoid writing the dots that indicate left-out material (as was done in Equation 1.2). The union notation is used exactly as summation notation and is read as “the union from i equals 1 to m of S sub i .”

$$\left| \bigcup_{i=1}^m S_i \right| = \sum_{i=1}^m |S_i|.$$

When we can write a set S as a union of disjoint sets $S_1, S_2, \dots, S_k$, we say that we have **partitioned** S into the sets $S_1, S_2, \dots, S_k$ and that the sets $S_1,$

¹It may look strange to have $|\{a, b, a\}| = 2$, but an element either is or is not in a set. An element cannot be in a set multiple times. (This situation leads to the idea of multisets, which are introduced in Section 1.5.) This example emphasizes that the notation $\{a, b, a\}$ means the same thing as $\{a, b\}$. Why would someone even contemplate the notation $\{a, b, a\}$? Suppose we wrote $S = \{x | x \text{ is the first letter of Ann, Bob, or Alice}\}$. Explicitly following this description of S would lead us to first write down $\{a, b, a\}$ and then realize it equals $\{a, b\}$.

$S_2, \dots, S_k$ form a **partition** of S . Thus, $\{\{1\}, \{3, 5\}, \{2, 4\}\}$ is a partition of the set $\{1, 2, 3, 4, 5\}$, and the set $\{1, 2, 3, 4, 5\}$ can be partitioned into the sets $\{1\}, \{3, 5\}, \{2, 4\}$. However, it is clumsy to say we are partitioning a set into sets; instead, we call the sets S_i , into which we partition a set S , the **blocks** of the partition. Thus, the sets $\{1\}, \{3, 5\}, \{2, 4\}$ are the blocks of a partition of $\{1, 2, 3, 4, 5\}$. In this language, we can restate the sum principle as follows.

Principle 1.2 (Sum Principle)

If a finite set S has been partitioned into blocks, then the size of S is the sum of the sizes of the blocks.

Abstraction

The process of figuring out a general principle that explains why a certain computation makes sense is an example of the mathematical process of **abstraction**. In this book, we don't give a precise definition of abstraction; instead, we provide examples of the process as we proceed. In a course in set theory, we would further abstract our work and derive the sum principle from the axioms of set theory. In a course in discrete mathematics, however, this level of abstraction is unnecessary. We simply use the sum principle as the basis of computations when it is convenient to do so. If our goal were to solve only Exercise 1.1-1, then our abstraction would have been almost a mindless exercise that complicated what was an "obvious" solution. However, the sum principle will prove to be useful in a variety of problems. Thus, the value of abstraction is that recognizing the abstract elements of a problem often helps us solve subsequent problems.

Summing Consecutive Integers

Returning to the problem in Exercise 1.1-1, it would be nice to find a simpler form for the sum given in Equation 1.1. We may write this sum as

$$\sum_{i=1}^{n-1} (n - i).$$

To avoid summing the values of $n - i$, we observe that the values we are summing are $n - 1, n - 2, \dots, 1$; so, we may write

$$\sum_{i=1}^{n-1} (n - i) = \sum_{i=1}^{n-1} i.$$

4 Chapter 1: Counting

A clever trick, usually attributed to Carl Friedrich Gauss, gives us a shorter formula for this sum:

$$\begin{array}{ccccccc} 1 & + & 2 & + & \cdots & + & (n-2) & + & (n-1) \\ + & (n-1) & + & (n-2) & + & \cdots & + & 2 & + & 1 \\ \hline & n & + & n & + & \cdots & + & n & + & n \end{array}$$

The sum below the horizontal line has $n - 1$ terms, each equal to n . Thus, the sum is $n(n - 1)$, or the sum of the two sums above the line. Because these sums are equal (identical except for being in reverse order), the sum below the line must be twice either sum above. Therefore, either of the sums above the line must be $n(n - 1)/2$. In other words, we may write

$$\sum_{i=1}^{n-1} (n - i) = \sum_{i=1}^{n-1} i = \frac{n(n - 1)}{2}.$$

This lovely trick is quite helpful in other similar situations involving a sum of variables. There are other ways to get the formula that don't use a trick. At the end of this section, after we analyze Exercise 1.1-2 and abstract the process used there, we will come back to this problem to see how we could have discovered this formula for ourselves without any tricks.

The Product Principle

Exercise 1.1-2

The following loop is part of a program that computes the product of two matrices. (You don't need to know how to find the product of two matrices to do this exercise.)

```
(1) for  $i = 1$  to  $r$ 
(2)   for  $j = 1$  to  $m$ 
(3)      $S = 0$ 
(4)     for  $k = 1$  to  $n$ 
(5)        $S = S + A[i, k] * B[k, j]$ 
(6)      $C[i, j] = S$ 
```

How many multiplications (expressed in terms of r , m , and n) does this pseudocode carry out in total among all the iterations of Line 5?

Exercise 1.1-3

Consider the following longer piece of pseudocode that sorts a list of numbers and then counts big gaps in the list. (For this exercise, a "big gap" is a place where a number in the list is more than twice the previous number.)

```

(1)  for  $i = 1$  to  $n - 1$ 
(2)      minval =  $A[i]$ 
(3)      minindex =  $i$ 
(4)      for  $j = i$  to  $n$ 
(5)          if ( $A[j] < \text{minval}$ )
(6)              minval =  $A[j]$ 
(7)              minindex =  $j$ 
(8)      exchange  $A[i]$  and  $A[\text{minindex}]$ 
(9)  bigjump = 0
(10) for  $i = 2$  to  $n$ 
(11)     if ( $A[i] > 2 * A[i - 1]$ )
(12)         bigjump = bigjump + 1

```

How many comparisons does the pseudocode make in Lines 5 and 11?

In Exercise 1.1-2, the program segment in Lines 4–5, which we call the “inner loop,” takes exactly n steps. Thus, it makes n multiplications, regardless of what the variables i and j are. The program segment in Lines 2–5 repeats the inner loop exactly m times, regardless of what i is. Therefore, this program segment makes n multiplications m times, or nm multiplications.

Why did we add in Exercise 1.1-1 but multiply here? We can answer this question using the abstract point of view we adopted in discussing Exercise 1.1-1. The algorithm in Exercise 1.1-2 performs a certain set of multiplications. For any given i , the set of multiplications performed in Lines 2–5 can be divided into the set S_1 of multiplications performed when $j = 1$, the set S_2 of multiplications performed when $j = 2$, and, in general, the set S_j of multiplications performed for any given value of j . Each set S_j consists of those multiplications that the inner loop carries out for a particular value of j , and there are exactly n multiplications in this set. Let T_i be the set of multiplications that our program segment carries out for a certain value of i . The set T_i is the union of the sets S_j . We use the standard notation for unions to write

$$T_i = \bigcup_{j=1}^m S_j.$$

By the sum principle, the size of the set T_i is the sum of the sizes of the sets S_j . A sum of m numbers, each equal to n , is mn . Stated as an equation,

$$|T_i| = \left| \bigcup_{j=1}^m S_j \right| = \sum_{j=1}^m |S_j| = \sum_{j=1}^m n = mn. \quad (1.3)$$

Thus, we multiplied because multiplication is repeated addition. From our solution, we extract a second principle that simply shortcuts the use of the sum principle.

Principle 1.3 (Product Principle)

The size of a union of m disjoint sets, each of size n , is mn .

We now complete our discussion of Exercise 1.1-2. Lines 2–5 are executed once for each value of i from 1 to r . A different i value is used each time those lines are executed; so, the set of multiplications in one execution is disjoint from the set of multiplications in any other. Thus, the set of all multiplications that the program carries out is a union of r disjoint sets T_i , each of which consists of mn multiplications. By the product principle, the set of all multiplications has size rmn . Therefore, the program carries out rmn multiplications.

Exercise 1.1-3 shows us that thinking about whether the sum or product principle is appropriate for a problem can help decompose the problem into easily solvable pieces. If we can decompose the problem into smaller pieces and solve the smaller pieces, then we may be able to either add or multiply solutions to smaller problems in order to solve the larger problem. In this exercise, the number of comparisons in the program fragment is the sum of the number of comparisons in the first loop (Lines 1–8) and the number of comparisons in the second loop (Lines 10–12). (What two disjoint sets are we talking about here?) Furthermore, the first loop makes $n(n+1)/2 - 1$ comparisons,² and the second loop has $n - 1$ comparisons. The number of comparisons made by the fragment is $n(n+1)/2 - 1 + n - 1 = n(n+1)/2 + n - 2$ comparisons.

Two-Element Subsets

There are often several ways to solve a problem. We originally solved Exercise 1.1-1 using the sum principle, but it is also possible to solve it using the product principle. Solving a problem two ways not only increases our confidence that we have found the correct solution, but it can also allow us to make new connections and yield valuable insight.

Consider the set of comparisons made by the entire execution of the code in Exercise 1.1-1. When $i = 1$, variable j takes on every value from 2 to

²To see why this is true, ask yourself first where the $n(n+1)/2$ comes from and then why we subtracted 1.

n . When $i = 2$, variable j takes on every value from 3 to n . Thus, for each two numbers i and j , we compare $A[i]$ and $A[j]$ exactly once in our loop. (The order in which we compare them depends on whether i or j is smaller.) Thus, the number of comparisons we make is the same as the number of two-element subsets of the set $\{1, 2, \dots, n\}$.³ In how many ways can we choose two elements from this set? If we choose a first and second element, there are n ways to choose a first element, and for each choice of the first element, there are $n - 1$ ways to choose a second element. Thus, the set of all such choices is the union of n sets of size $n - 1$, one set for each first element. It might appear that, by the product principle, there are $n(n - 1)$ ways to choose two elements from our set. However, what we have chosen is an **ordered pair**, or a pair of elements in which one comes first and the other comes second. For example, we could choose 2 first and 5 second to get the ordered pair $(2, 5)$, or we could choose 5 first and 2 second to get the ordered pair $(5, 2)$. Because each pair of distinct elements of $\{1, 2, \dots, n\}$ can be ordered in two ways, we get twice as many ordered pairs as two-element sets. Thus, because the number of ordered pairs is $n(n - 1)$, the number of two-element subsets of $\{1, 2, \dots, n\}$ is $n(n - 1)/2$. Therefore, the answer to Exercise 1.1-1 is $n(n - 1)/2$. This number comes up so often, it has its own name and notation: we call this number “ n choose 2” and denote it by $\binom{n}{2}$. To summarize, $\binom{n}{2}$ stands for the number of two-element subsets of an n -element set and equals $n(n - 1)/2$. Because one answer to Exercise 1.1-1 is $1 + 2 + \dots + (n - 1)$ and a second answer to Exercise 1.1-1 is $\binom{n}{2}$, we see that

$$1 + 2 + \dots + (n - 1) = \binom{n}{2} = \frac{n(n - 1)}{2}.$$

Important Concepts, Formulas, and Theorems

1. *Set*. A *set* is a collection of objects. In a set, order is not important. Thus, the set $\{A, B, C\}$ is the same as the set $\{A, C, B\}$. An element either is or is not in a set; it cannot be in a set more than once, even if a description of a set names that element more than once.
2. *Disjoint*. Two sets are *disjoint* if they have no elements in common.
3. *Mutually disjoint sets*. A set of sets $\{S_1, \dots, S_n\}$ is a family of *mutually disjoint sets* if each two of the sets S_i are disjoint.
4. *Size of a set*. Given a set S , the size of S , denoted $|S|$, is the number of distinct elements in S .

³The relationship between the set of comparisons and the set of two-element subsets of $\{1, 2, \dots, n\}$ is an example of a bijection, an idea that we examine more in Section 1.2.

8 Chapter 1: Counting

5. *Sum principle.* The size of a union of a family of mutually disjoint sets is the sum of the sizes of the sets. In other words, if $S_1, S_2, \dots, S_n$ are disjoint sets, then

$$|S_1 \cup S_2 \cup \dots \cup S_n| = |S_1| + |S_2| + \dots + |S_n|.$$

To avoid the "dots" that indicate left-out material, we write

$$\left| \bigcup_{i=1}^n S_i \right| = \sum_{i=1}^n |S_i|.$$

6. *Partition of a set.* A partition of a set S is a set of mutually disjoint subsets (sometimes called *blocks*) of S whose union is S .
7. *Sum of first $n - 1$ numbers.*

$$\sum_{i=1}^{n-1} n - i = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}.$$

8. *Product principle.* The size of a union of m disjoint sets, each of size n , is mn .
9. *Two-element subsets.* The number of two-element subsets of an n -element set, denoted $\binom{n}{2}$, equals $n(n-1)/2$. $\binom{n}{2}$ is read as " n choose 2."

Problems

All problems with blue boxes have an answer or hint available at the end of the book.

1. The following segment of code is part of a program that uses insertion sort to sort a list A .

```
for i = 2 to n
    j = i
    while (j ≥ 2) and (A[j] < A[j - 1])
        exchange A[j] and A[j - 1]
    j = j - 1
```

What is the maximum number of times (considering all lists of n items that you could be asked to sort) the program makes the comparison $A[j] < A[j - 1]$? Describe as succinctly as you can those lists that require this number of comparisons.

2. Five schools are going to send their baseball teams to a tournament in which each team must play each other team exactly once. How many games are required?

3. In how many ways can you draw a first card and then a second card from a deck of 52 cards?
4. In how many ways can you draw two cards from a deck of 52 cards?
5. In how many ways can you draw a first, second, and third card from a deck of 52 cards?
6. In how many ways can a 10-person club select a president and a secretary-treasurer from among its members?
7. In how many ways can a 10-person club select a two-person executive committee from among its members?
8. In how many ways can a 10-person club select a president and a two-person executive advisory board from among its members (assuming that the president is not on the advisory board)?
9. Using the formula for $\binom{n}{2}$, it is straightforward to show that

$$n \binom{n-1}{2} = \binom{n}{2} (n-2).$$

However, this proof simply uses blind substitution and simplification. Find a more conceptual explanation of why this formula is true. (*Hint:* Think in terms of officers and committees in a club.)

10. If M is an m -element set and N is an n -element set, how many ordered pairs are there with the first member in M and the second member in N ?
11. The local ice cream shop sells ten different flavors of ice cream. How many different two-scoop cones are there? (Following your mother's rule that it all goes to the same stomach, a cone with a vanilla scoop on top of a chocolate scoop is considered the same as a cone with chocolate on top of vanilla.)
12. Suppose you decide to disagree with your mother in Problem 11—the order of the scoops does matter. How many different possible two-scoop cones are there?
13. Suppose on Day 1 you receive one penny, and, for $i > 1$, on Day i you receive twice as many pennies as you did on Day $i - 1$. How many pennies will you have on Day 20? How many will you have on Day n ? Can you justify your answer by using the sum or product principle?
14. The Pile High Deli offers a simple sandwich, consisting of your choice of one of five different kinds of bread; either butter, mayonnaise, or no spread; one of three different kinds of meat; and one of three different kinds of cheese, with the meat and cheese