Chapter 2

Basic Counting Rules[1]

2.1 THE PRODUCT RULE

Some basic counting rules underlie all of combinatorics. We summarize them in this chapter. The reader who is already familiar with these rules may wish to review them rather quickly. This chapter also introduces a widely used tool for proving that a certain kind of arrangement or pattern exists. In reading this chapter the reader already familiar with counting may wish to concentrate on the variety of applications that may not be as familiar, many of which are returned to in later chapters.

Example 2.1 Bit Strings and Binary Codes (Example 1.2 Revisited) Let us return to our binary code example (Example 1.2), and ask again how many letters of the alphabet can be encoded if there are exactly 2 bits. Let us get the answer by drawing a tree diagram. We do that in Figure 2.1. There are 4 possible strings of 2 bits, as we noted before. The reader will observe that there are 2 choices for the first bit, and for each of these choices, there are 2 choices for the second bit, and 4 is 2×2. ■

Example 2.2 DNA The total of all the genetic information of an organism is its *genome*. It is convenient to think of the genome as one long deoxyribonucleic acid (DNA) molecule. (The genome is actually made up of pieces of DNA representing the individual chromosomes.) The DNA (or chromosomes) is composed of a string of building blocks known as nucleotides. The genome size can be expressed in terms of the total number of nucleotides. Since DNA is actually double-stranded with the two strands held together by virtue of pairings between specific bases (a base being one of the three subcomponents of a nucleotide), genome sizes are usually

[1]This chapter was written by Helen Marcus-Roberts, Fred S. Roberts, and Barry A. Tesman.

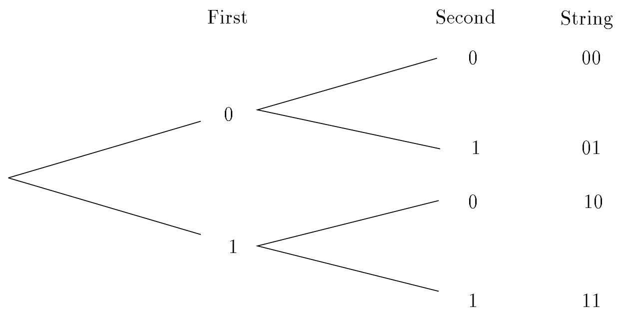

Figure 2.1: A tree diagram for counting the number of bit strings of length 2.

expressed in terms of base pairs (bp). Each base in a nucleotide is one of four possible chemicals: thymine, T; cytosine, C; adenine, A; guanine, G. The sequence of bases encodes certain genetic information. In particular, it determines long chains of amino acids which are known as proteins. There are 20 basic amino acids. A sequence of bases in a DNA molecule will encode one such amino acid. How long does a string of a DNA molecule have to be for there to be enough possible bases to encode 20 different amino acids? For example, can a 2-element DNA sequence encode for the 20 different basic amino acids? To answer this, we need to ask: How many 2-element DNA sequences are there? The answer to this question is again given by a tree diagram, as shown in Figure 2.2. We see that there are 16 possible 2-element DNA sequences. There are 4 choices for the first element, and for each of these choices, there are 4 choices for the second element; the reader will notice that 16 is 4×4. Notice that there are not enough 2-element sequences to encode for all 20 different basic amino acids. In fact, a sequence of 3 elements does the encoding in practice. A simple counting procedure has shown why at least 3 elements are needed. ∎

The two examples given above illustrate the following basic rule.

PRODUCT RULE: If something can happen in n_1 ways, *and* no matter how the first thing happens, a second thing can happen in n_2 ways, then the two things together can happen in $n_1 \times n_2$ ways. More generally, if something can happen in n_1 ways, *and* no matter how the first thing happens, a second thing can happen in n_2 ways, *and* no matter how the first two things happen, a third thing can happen in n_3 ways, *and* ..., then all the things together can happen in

$$n_1 \times n_2 \times n_3 \times \cdots$$

ways.

Returning to bit strings, we see immediately by the product rule that the number of strings of exactly 3 bits is given by $2 \times 2 \times 2 = 2^3 = 8$ since there are two choices for the first bit (0 or 1), and no matter how it is chosen, there are two choices for

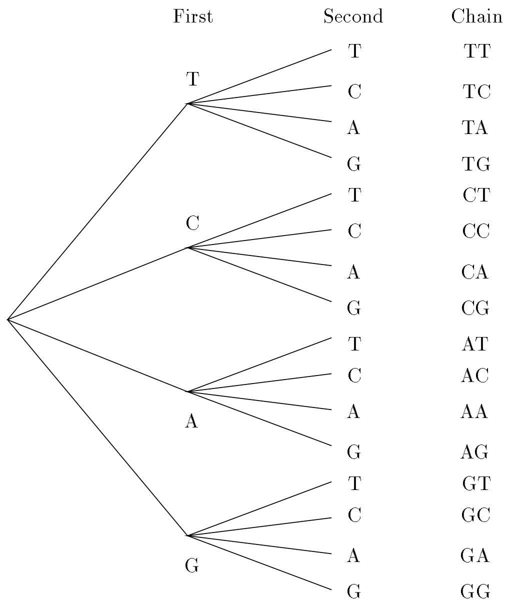

Figure 2.2: A tree diagram for counting the number of 2-element DNA sequences.

the second bit, and no matter how the first 2 bits are chosen, there are two choices for the third bit. Similarly, in the pipeline problem of Example 1.3, if there are 7 choices of pipe size for each of 3 links, there are

$$7 \times 7 \times 7 = 7^3 = 343$$

different possible networks. If there are 40 links, there are

$$7 \times 7 \times \cdots \times 7 = 7^{40}$$

different possible networks. Note that by our observations in Chapter 1, it is infeasible to count the number of possible pipeline networks by enumerating them (listing all of them). Some method of counting other than enumeration is needed. The product rule gives such a method. In the early part of this book, we shall be concerned with such simple methods of counting.

Next, suppose that A is a set of a objects and B is a set of b objects. Then the number of ways to pick one object from A and then one object from B is $a \times b$. This statement is a more precise version of the product rule.

To give one final example, the number of 3-element DNA sequences is

$$4 \times 4 \times 4 = 4^3 = 64.$$

That is why there are enough different 3-element sequences to encode for all 20 different basic amino acids; indeed, several different chains encode for the same

Table 2.1: The Number of Possible DNA Sequences for Various Organisms

Phylum	Genus and Species	Genome size (base pairs)	Number of possible sequences
Algae	*P. salina*	6.6×10^5	$4^{6.6\times10^5} > 10^{3.97\times10^5}$
Mycoplasma	*M. pneumoniae*	1.0×10^6	$4^{1.0\times10^6} > 10^{6.02\times10^5}$
Bacterium	*E. coli*	4.2×10^6	$4^{4.2\times10^6} > 10^{2.52\times10^6}$
Yeast	*S. cerevisiae*	1.3×10^7	$4^{1.3\times10^7} > 10^{7.82\times10^6}$
Slime mold	*D. discoideum*	5.4×10^7	$4^{5.4\times10^7} > 10^{3.25\times10^7}$
Nematode	*C. elegans*	8.0×10^7	$4^{8.0\times10^7} > 10^{4.81\times10^7}$
Insect	*D. melanogaster*	1.4×10^8	$4^{1.4\times10^8} > 10^{8.42\times10^7}$
Bird	*G. domesticus*	1.2×10^9	$4^{1.2\times10^9} > 10^{7.22\times10^8}$
Amphibian	*X. laevis*	3.1×10^9	$4^{3.1\times10^9} > 10^{1.86\times10^9}$
Mammal	*H. sapiens*	3.3×10^9	$4^{3.3\times10^9} > 10^{1.98\times10^9}$

Source: Lewin [2000].

amino acid—this is different from the situation in Morse code, where strings of up to 4 bits are required to encode for all 26 letters of the alphabet but not every possible string is used. In Section 2.9 we consider Gamow's [1954a,b] suggestion about which 3-element sequences encode for the same amino acid.

Continuing with DNA molecules, we see that the number of sequences of 4 bases is 4^4, the number with 100 bases is 4^{100}. How long is a full-fledged DNA molecule? Some answers are given in Table 2.1. Notice that in slime mold (*D. discoideum*), the genome has 5.4×10^7 bases or base pairs. Thus, the number of such sequences is

$$4^{5.4\times10^7},$$

which is greater than

$$10^{3.2\times10^7}.$$

This number is 1 followed by 3.2×10^7 zeros or 32 million zeros! It is a number that is too large to comprehend. Similar results hold for other organisms. By a simple counting of all possibilities, we can understand the tremendous possible variation in genetic makeup. It is not at all surprising, given the number of possible DNA sequences, that there is such an amazing variety in nature, and that two individuals are never the same. It should be noted once more that given the tremendous magnitude of the number of possibilities, it would not have been possible to count these possibilities by the simple expedient of enumerating them. It was necessary to develop rules or procedures for counting, which counted the number of possibilities without simply listing them. That is one of the three basic problems in combinatorics: developing procedures for counting without enumerating.

As large as the number of DNA sequences is, it has become feasible, in part due to the use of methods of combinatorial mathematics, to "sequence" and "map"

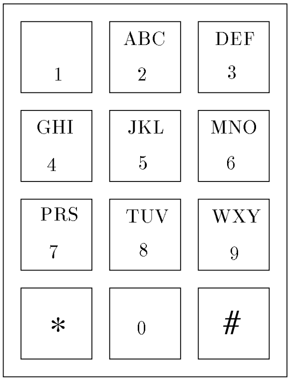

Figure 2.3: A telephone pad.

genomes of different organisms, including humans. A *gene* is a strip of DNA that carries the code for making a particular protein. “Mapping” the genome would require localizing each of its genes; “sequencing” it would require determining the exact order of the bases making up each gene. In humans, this involves approximately 100,000 genes, each with a thousand or more bases. For more on the use of combinatorial mathematics in mapping and sequencing the genome, see Clote and Backofen [2000], Congress of the United States [1988], Farach-Colton, *et al.* [1999], Gusfield [1997], Lander and Waterman [1995], Pevzner [2000], Setubal and Meidanis [1997], or Waterman [1995].

Example 2.3 Telephone Numbers At one time, a local telephone number was given by a sequence of two letters followed by five numbers. How many different telephone numbers were there? Using the product rule, one is led to the answer:

$$26 \times 26 \times 10 \times 10 \times 10 \times 10 \times 10 = 26^2 \times 10^5.$$

While the count is correct, it doesn't give a good answer, for two letters on the same place on the pad led to the same telephone number. The reader might wish to envision a telephone pad. (A rendering of one is given in Figure 2.3.) There are three letters on all digits, except that 1 and 0 have no letters. Hence, letters A, B, and C were equivalent; so were W, X, and Y; and so on. There were, in effect, only 8 different letters. The number of different telephone numbers was therefore

$$8^2 \times 10^5 = 6.4 \times 10^6.$$

Thus, there were a little over 6 million such numbers. In the 1950s and 1960s, most local numbers were changed to become simply seven-digit numbers, with the restriction that neither of the first two digits could be 0 or 1. The number of telephone numbers was still $8^2 \times 10^5$. Direct distance dialing was accomplished by adding a three-digit area code. The area code could not begin with a 0 or 1, and it had to have 0 or 1 in the middle. Using these restrictions, we compute that the number of possible telephone numbers was

$$8 \times 2 \times 10 \times 8^2 \times 10^5 = 1.024 \times 10^9.$$

Table 2.2: Two Switching Functions

Bit string x	$S(x)$	$T(x)$
00	1	0
01	0	0
10	0	1
11	1	1

That was enough to service over 1 billion customers. To service even more customers, direct distance dialing was changed to include a starting 1 as an 11th digit for long-distance calls. This freed up the restriction that an area code must have a 0 or 1 in the middle. The number of telephone numbers grew to

$$1 \times 8 \times 10 \times 10 \times 8^2 \times 10^5 = 5.12 \times 10^9.$$

With increasingly better technology, the telecommunications industry could boast that with the leading 1, there are no restrictions on the next 10 digits. Thus, there are now 10^{10} possible telephone numbers. However, demand continues to increase at a very fast pace (e.g., multiple lines, fax machines, cellular phones, pagers, etc.). What will we do when 10^{10} numbers are not enough? ∎

Example 2.4 Switching Functions Let B_n be the set of all bit strings of length n. A *switching function (Boolean function) of n variables* is a function that assigns to each bit string of length n a number 0 or 1. For instance, let $n = 2$. Then $B_2 = \{00, 01, 10, 11\}$. Two switching functions S and T defined on B_2 are given in Table 2.2. The problem of making a detailed design of a digital computer usually involves finding a practical circuit implementation of certain functional behavior. A computer device implements a switching function of two, three, or four variables. Now every switching function can be realized in numerous ways by an electrical network of interconnections. Rather than trying to figure out from scratch an efficient design for a given switching function, a computer engineer would like to have a catalog that lists, for every switching function, an efficient network realization. Unfortunately, this seems at first to be an impractical goal. For how many switching functions of n variables are there? There are 2^n elements in the set B_n by a generalization of Example 2.1. Hence, by the product rule, there are $2 \times 2 \times \cdots \times 2$ different n-variable switching functions, where there are 2^n terms in the product. In total, there are 2^{2^n} different n-variable switching functions. Even the number of such functions for $n = 4$ is 65,536, and the number grows astronomically fast. Fortunately, by taking advantage of symmetries, we can consider certain switching functions equivalent as far as what they compute is concerned. Then we need not identify the best design for every switching function; we need do it only for enough switching functions so that every other switching function is equivalent to one of those for which we have identified the best design. While the first computers were being built, a team of researchers at Harvard painstakingly enumerated all possible switching functions of 4 variables, and determined which were equivalent. They discovered that it was possible to reduce every switching function to one of 222 types (Harvard Computation Laboratory Staff [1951]). In Chapter 8 we show how to derive results such as this from a powerful theorem due to George Pólya. For a more detailed discussion of switching functions, see Deo [1974, Ch. 12], Harrison [1965], Hill and Peterson [1968], Kohavi [1970], Liu [1977], Muroga [1990], Pattavina [1998], Prather [1976], or Stone [1973]. ■

Example 2.5 Food Allergies An allergist sees a patient who often develops a severe upset stomach after eating. Certain foods are suspected of causing the problem: tomatoes, chocolate, corn, and peanuts. It is not clear if the problem arises because of one of these foods or a combination of them acting together. The allergist tells the patient to try different combinations of these foods to see whether there is a reaction. How many different combinations must be tried? Each food can be absent or present. Thus, there are $2 \times 2 \times 2 \times 2 = 2^4 = 16$ possible combinations. In principle, there are 2^{2^4} possible manifestations of food allergies based on these four foods; each possible combination of foods can either bring forth an allergic reaction or not. Each person's individual sensitivity to combinations of these foods corresponds to a switching function $S(x_1, x_2, x_3, x_4)$ where x_1 is 1 if there are tomatoes in the diet, x_2 is 1 if there is chocolate in the diet, x_3 is 1 if there is corn in the diet, x_4 is 1 if there are peanuts in the diet. For instance, a person who develops an allergic reaction any time tomatoes are in the diet or any time both corn and peanuts are in the diet would demonstrate the switching function S which has $S(1,0,0,0) = 1$, $S(1,1,0,0) = 1$, $S(0,0,1,1) = 1$, $S(0,1,1,0) = 0$, and so on. In practice, it is impossible to know the value of a switching function on all possible bit strings if the number of variables is large; there are just too many possible bit strings. Then the practical problem is to develop methods to guess the value of a switching function that is only partially defined. There is much recent work on this problem. See, for example, Boros, *et al.* [1995], Boros, Ibaraki, and Makino [1998], Crama, Hammer, and Ibaraki [1988], and Ekin, Hammer, or Kogan [2000]. Similar cause-and-effect problems occur in diagnosing failure of a complicated electronic system given a record of failures when certain components fail (we shall have more to say about this in Example 2.21) and in teaching a robot to maneuver in an area filled with obstacles where an obstacle might appear as a certain pattern of dark or light pixels and in some situations the pattern of pixels corresponds to an object and in others it does not. For other applications, see Boros, *et al.* [2000]. ■

EXERCISES FOR SECTION 2.1[2]

1. The population of Carlisle, Pennsylvania, is about 20,000. If each resident has three initials, is it true that there must be at least two residents with the same initials? Give a justification of your answer.

[2] *Note to reader:* In the exercises in Chapter 2, exercises after each section can be assumed to use techniques of some previous (nonoptional) section, not necessarily exactly the techniques just introduced. Also, there are additional exercises at the end of the chapter. Indeed, sometimes an exercise is included which does not make use of the techniques of the current section. To understand a new technique, one must understand when it does not apply as well as when it applies.

2. A library has 1,700,000 books, and the librarian wants to encode each using a codeword consisting of 3 letters followed by 3 numbers. Are there enough codewords to encode all 1,700,000 books with different codewords?
3. (a) Continuing with Exercise 7 of Chapter 1, compute the maximum number of strings of length at most 3 in a trinary code.

   (b) Repeat for length at most 4.

   (c) Repeat for length exactly 4, but beginning with a 0 or 1.
4. In our discussion of telephone numbers, suppose that we maintain the original restrictions on area code as in Example 2.3. Suppose that we lengthen the local phone number, allowing it to be any eight-digit number with the restriction that none of the first three digits can be 0 or 1. How many local phone numbers are there? How many phone numbers are there including area code?
5. If we want to use bit strings of length at most n to encode not only all 26 letters of the alphabet, but also all 10 decimal digits, what is the smallest number n that works? (What is n for Morse code?)
6. How many $m \times n$ matrices are there each of whose entries is 0 or 1?
7. A musical band has to have at least one member. It can contain at most one drummer, at most one pianist, at most one bassist, at most one lead singer, and at most two background singers. How many possible bands are there if we consider any two drummers indistinguishable, and the same holds true for the other categories, and hence call two bands the same if they have the same number of members of each category? Justify your answer.
8. How many nonnegative integers less than 1 million contain the digit 2?
9. Enumerate all switching functions of 2 variables.
10. If a function assigns 0 or 1 to each switching function of n variables, how many such functions are there?
11. A switching function S is called *self-dual* if the value S of a bit string is unchanged when 0's and 1's are interchanged. For instance, the function S of Table 2.2 is self-dual, but the function T of that table is not. How many self-dual switching functions of n variables are there?
12. (Stanat and McAllister [1977]) In some computers, an integer (positive or negative) is represented by using bit strings of length p. The last bit in the string represents the sign, and the first $p-1$ bits are used to encode the integer. What is the largest number of distinct integers that can be represented in this way for a given p? What if 0 must be one of these integers? (The sign of 0 is $+$ or $-$.)
13. (Stanat and McAllister [1977]) Every integer can be represented (nonuniquely) in the form $a \times 2^b$, where a and b are integers. The *floating-point representation* for an integer uses a bit string of length p to represent an integer by using the first m bits to encode a and the remaining $p-m$ bits to encode b, with the latter two encodings performed as described in Exercise 12.

    (a) What is the largest number of distinct integers that can be represented using the floating-point notation for a given p?

    (b) Repeat part (a) if the floating-point representation is carried out in such a way that the leading bit for encoding the number a is 1.

    (c) Repeat part (a) if 0 must be included.

14. When acting on loan applications it can be concluded, based on historical records, that loan applicants having certain combinations of features can be expected to repay their loans and those who have other combinations of features cannot. As their main features, suppose that a bank uses:

    Marital Status: Married, Single (never married), Single (previously married).

    Past Loan: Previous default, No previous default.

    Employment: Employed, Unemployed (within 1 year), Unemployed (more than 1 year).

    (a) How many different loan applications are possible when considering these features?

    (b) How many manifestations of loan repayment/default are possible when considering these features?

2.2 THE SUM RULE

We turn now to the second fundamental counting rule. Consider the following example.

Example 2.6 Congressional Delegations There are 100 senators and 435 members of the House of Representatives. A delegation is being selected to see the President. In how many different ways can such a delegation be picked if it consists of one senator *and* one representative? The answer, by the product rule, is

$$100 \times 435 = 43{,}500.$$

What if the delegation is to consist of one member of the Senate *or* one member of the House? Then there are

$$100 + 435 = 535$$

possible delegations. This computation illustrates the second basic rule of counting, the sum rule. ■

SUM RULE: If one event can occur in n_1 ways and a second event in n_2 (different) ways, then there are $n_1 + n_2$ ways in which *either* the first event *or* the second event can occur (*but not both*). More generally, if one event can occur in n_1 ways, a second event can occur in n_2 (different) ways, a third event can occur in n_3 (still different) ways, ..., then there are

$$n_1 + n_2 + n_3 + \cdots$$

ways in which (exactly) one of the events can occur.

In Example 2.6 we have italicized the words "and" and "or." These key words usually indicate whether the sum rule or the product rule is appropriate. The word "and" suggests the product rule; the word "or" suggests the sum rule.

Example 2.7 Draft Picks A professional football team has two draft choices to make and has limited the choice to 3 quarterbacks, 4 linebackers, and 5 wide receivers. To pick a quarterback and linebacker there are $3 \times 4 = 12$ ways, by the product rule. How many ways are there to pick two players if they must play different positions? You can pick either a quarterback and linebacker, quarterback and wide receiver, or linebacker and wide receiver. There are, by previous computation, 12 ways of doing the first. There are 15 ways of doing the second (why?) and 20 ways of doing the third (why?). Hence, by the sum rule, the number of ways of choosing the two players from different positions is

$$12 + 15 + 20 = 47. \qquad \blacksquare$$

Example 2.8 Variables in BASIC and JAVA The programming language BASIC (standing for Beginner's All-Purpose Symbolic Instruction Code) dates back to 1964. Variable names in early implementations of BASIC could either be a letter, a letter followed by a letter, or a letter followed by a decimal digit, that is, one of the numbers $0, 1, \ldots, 9$. How many different variable names were possible? By the product rule, there were $26 \times 26 = 676$ and $26 \times 10 = 260$ names of the latter two kinds, respectively. By the sum rule, there were $26+676+260 = 962$ variable names in all.

The need for more variables was but one reason for more advanced programming languages. For example, the JAVA programming language, introduced in 1995, has variable name lengths ranging from 1 to 65,535 characters. Each character can be a letter (uppercase or lowercase), an underscore, a dollar sign, or a decimal digit except that the first character cannot be a decimal digit. By using the sum rule, we see that the number of possible characters is $26 + 26 + 1 + 1 + 10 = 64$ except for the first character which has only $64 - 10 = 54$ possibilities. Finally, by using the sum and product rules, we see that the number of variable names is

$$54 \cdot 64^{65,534} + 54 \cdot 64^{65,533} + \cdots + 54 \cdot 64 + 54.$$

This certainly allows for more than enough variables.[3] $\blacksquare$

In closing this section, let us restate the sum rule this way. Suppose that A and B are disjoint sets and we wish to pick exactly one element, picking it from A *or* from B. Then the number of ways to pick this element is the number of elements in A plus the number of elements in B.

EXERCISES FOR SECTION 2.2

1. How many bit strings have length 3, 4, or 5?
2. A committee is to be chosen from among 8 scientists, 7 psychics, and 12 clerics. If the committee is to have two members of different backgrounds, how many such committees are there?

[3] The value of just the first term in the sum, $54 \cdot 64^{65,534}$, is approximately $8.527 \times 10^{118,367}$. Arguably, there are at least on the order of 10^{80} atomic particles in the universe (e.g., see Dembski [1998]).

3. How many numbers are there which have five digits, each being a number in $\{1, 2, \ldots, 9\}$, and either having all digits odd or having all digits even?
4. Each American Express card has a 15-digit number for computer identification purposes. If each digit can be any number between 0 and 9, are there enough different account numbers for 10 million credit-card holders? Would there be if the digits were only 0 or 1?
5. How many 5-letter words either start with d or do not have the letter d?
6. In how many ways can we get a sum of 3 or a sum of 4 when two dice are rolled?
7. Suppose that a pipeline network is to have 30 links. For each link, there are 2 choices: The pipe may be any one of 7 sizes and any one of 3 materials. How many different pipeline networks are there?
8. How many DNA chains of length 3 have no C's at all or have no T's in the first position?

2.3 PERMUTATIONS

In combinatorics we frequently talk about n-element sets, sets consisting of n distinct elements. It is convenient to call these *n-sets.* A *permutation* of an n-set is an arrangement of the elements of the set in order. It is often important to count the number of permutations of an n-set.

Example 2.9 Job Interviews Three people, Ms. Jordan, Mr. Harper, and Ms. Gabler, are scheduled for job interviews. In how many different orders can they be interviewed? We can list all possible orders, as follows:

1. Jordan, Harper, Gabler
2. Jordan, Gabler, Harper
3. Harper, Jordan, Gabler
4. Harper, Gabler, Jordan
5. Gabler, Jordan, Harper
6. Gabler, Harper, Jordan

We see that there are 6 possible orders. Alternatively, we can observe that there are 3 choices for the first person being interviewed. For each of these choices, there are 2 remaining choices for the second person. For each of these choices, there is 1 remaining choice for the third person. Hence, by the product rule, the number of possible orders is

$$3 \times 2 \times 1 = 6.$$

Each order is a permutation. We are asking for the number of permutations of a 3-set, the set consisting of Jordan, Harper, and Gabler.

If there are 5 people to be interviewed, counting the number of possible orders can still be done by enumeration; however, that is rather tedious. It is easier to observe that now there are 5 possibilities for the first person, 4 remaining possibilities for the second person, and so on, resulting in

$$5 \times 4 \times 3 \times 2 \times 1 = 120$$

Table 2.3: Values of $n!$ for n from 0 to 10

n	0	1	2	3	4	5	6	7	8	9	10
$n!$	1	1	2	6	24	120	720	5,040	40,320	362,880	3,628,800

possible orders in all. ■

The computations of Example 2.9 generalize to give us the following result: The number of permutations of an n-set is given by

$$n \times (n-1) \times (n-2) \times \cdots \times 1 = n!$$

In Example 1.1 we discussed the number of orders in which to take 5 different drugs. This is the same as the number of permutations of a 5-set, so it is $5! = 120$. To see once again why counting by enumeration rapidly becomes impossible, we show in Table 2.3 the values of $n!$ for several values of n.

The number 25!, to give an example, is already so large that it is incomprehensible. To see this, note that

$$25! \approx 1.55 \times 10^{25}.$$

A computer checking 1 billion permutations per second would require almost half a billion years to look at 1.55×10^{25} permutations.[4] In spite of the result above, there are occasions where it is useful to enumerate all permutations of an n-set. In Section 2.16 we present an algorithm for doing so.

The number $n!$ can be approximated by computing $s_n = \sqrt{2\pi n}(n/e)^n$. The approximation of $n!$ by s_n is called *Stirling's approximation*. To see how good the approximation is, note that it approximates 5! as $s_5 = 118.02$ and 10! as $s_{10} =$ 3,598,600. (Compare these with the real values in Table 2.3.) The quality of the approximation is evidenced by the fact that the ratio of $n!$ to s_n approaches 1 as n approaches ∞ (grows arbitrarily large). (On the other hand, the difference $n! - s_n$ approaches ∞ as n approaches ∞.) For a proof, see an advanced calculus text such as Buck [1965].

EXERCISES FOR SECTION 2.3

1. List all permutations of

   (a) $\{1, 2, 3\}$ (b) $\{1, 2, 3, 4\}$

[4] To see why, note that there are approximately 3.15×10^7 seconds in a year. Thus, a computer checking 1 billion $= 10^9$ permutations per second can check $3.15 \times 10^7 \times 10^9 = 3.15 \times 10^{16}$ permutations in a year. Hence, the number of years required to check 1.55×10^{25} permutations is

$$\frac{1.55 \times 10^{25}}{3.15 \times 10^{16}} \approx 4.9 \times 10^8.$$

2. How many permutations of $\{1, 2, 3, 4, 5\}$ begin with 5?
3. How many permutations of $\{1, 2, \ldots, n\}$ begin with 1 and end with n?
4. Compute s_n and compare it to $n!$ if

   (a) $n = 4$ (b) $n = 6$ (c) $n = 8$

5. How many permutations of $\{1, 2, 3, 4\}$ begin with an odd number?
6. (a) How many permutations of $\{1, 2, 3, 4, 5\}$ have 2 in the second place?
   (b) How many permutations of $\{1, 2, \ldots, n\}$, $n \geq 3$, have 2 in the second place and 3 in the third place?
7. How many ways are there to rank five potential basketball recruits of different heights if the tallest one must be ranked first and the shortest one last?
8. (Cohen [1978])
   (a) In a six-cylinder engine, the even-numbered cylinders are on the left and the odd-numbered cylinders are on the right. A good firing order is a permutation of the numbers 1 to 6 in which right and left sides are alternated. How many possible good firing orders are there which start with a left cylinder?
   (b) Repeat for a $2n$-cylinder engine.
9. Ten job applicants have been invited for interviews, five having been told to come in the morning and five having been told to come in the afternoon. In how many different orders can the interviews be scheduled? Compare your answer to the number of different orders in which the interviews can be scheduled if all 10 applicants were told to come in the morning.

2.4 COMPLEXITY OF COMPUTATION

We have already observed that not all problems of combinatorics can be solved on the computer, at least not by enumeration. Suppose that a computer program implements an algorithm for solving a combinatorial problem. Before running such a program, we would like to know if the program will run in a "reasonable" amount of time and will use no more than a "reasonable" (or allowable) amount of storage or memory. The time or storage a program requires depends on the input. To measure how expensive a program is to run, we try to calculate a *cost function* or a *complexity function*. This is a function f that measures the cost, in terms of time required or storage required, as a function of the size n of the input problem. For instance, we might ask how many operations are required to multiply two square matrices of n rows and columns each. This number of operations is $f(n)$.

Usually, the cost of running a particular computer program on a particular machine will vary with the skill of the programmer and the characteristics of the machine. Thus there is a big emphasis in modern computer science on comparison of algorithms rather than programs, and on estimation of the complexity $f(n)$ of an algorithm, independent of the particular program or machine used to implement the algorithm. The desire to calculate complexity of algorithms is a major stimulus for the development of techniques of combinatorics.

Example 2.10 The Traveling Salesman Problem A salesman wishes to visit n different cities, starting and ending his business trip at the first city. He does not care in which order he visits the cities. What he does care about is to minimize the total cost of his trip. Assume that the cost of traveling from city i to city j is c_{ij}. The problem is to find an algorithm for computing the cheapest route, where the cost of a route is the sum of the c_{ij} for links used in the route. This is a typical combinatorial optimization problem.

For the traveling salesman problem, we shall be concerned with the enumeration algorithm: Enumerate all possible routes and calculate the cost of each route. We shall try to compute the complexity $f(n)$ of this algorithm, where n is the size of the input, that is, the number of cities. We shall assume that identifying a route and computing its cost is comparable for each route and takes 1 unit of time.

Now any route starting and ending at city 1 corresponds to a permutation of the remaining $n-1$ cities. Hence, there are $(n-1)!$ such routes, so $f(n) = (n-1)!$ units of time. We have already shown that this number can be extremely high. When n is 26 and $n-1$ is 25, we showed that $f(n)$ is so high that it is infeasible to perform this algorithm by computer. We return to the traveling salesman problem in Section 11.5. ■

It is interesting to note that the traveling salesman problem occurs in many guises. Examples 2.11 to 2.16 give some of the alternative forms in which this problem has arisen in practice.

Example 2.11 The Automated Teller Machine (ATM) Problem Your bank has many ATMs. Each day, a courier goes from machine to machine to make collections, gather computer information, and so on. In what order should the machines be visited in order to minimize travel time? This problem arises in practice at many banks. One of the first banks to use a traveling salesman algorithm to solve it, in the early days of ATMs, was Shawmut Bank in Boston.[5] ■

Example 2.12 The Phone Booth Problem Once a week, each phone booth in a region must be visited, and the coins collected. In what order should that be done in order to minimize travel time? ■

Example 2.13 The Problem of Robots in an Automated Warehouse The warehouse of the future will have orders filled by a robot. Imagine a pharmaceutical warehouse with stacks of goods arranged in rows and columns. An order comes in for 10 cases of aspirin, six cases of shampoo, eight cases of Band-Aids, and so on. Each is located by row, column, and height. In what order should the robot fill the order in order to minimize the time required? The robot needs to be programmed to solve a traveling salesman problem. (See Elsayed [1981] and Elsayed and Stern [1983].) ■

Example 2.14 A Problem of X-Ray Crystallography In x-ray crystallography, we must move a diffractometer through a sequence of prescribed angles. There

[5] This example is from Margaret Cozzens (personal communication).

is a cost in terms of time and setup for doing one move after another. How do we minimize this cost? (See Bland and Shallcross [1989].) ■

Example 2.15 Manufacturing In many factories, there are a number of jobs that must be performed or processes that must be run. After running process i, there is a certain setup cost before we can run process j; a cost in terms of time or money or labor of preparing the machinery for the next process. Sometimes this cost is small (e.g., simply making some minor adjustments) and sometimes it is major (e.g., requiring complete cleaning of the equipment or installation of new equipment). In what order should the processes be run to minimize total cost? (For more on this application, see Example 11.5 and Section 11.6.3.) ■

Example 2.16 Holes in Circuit Boards In 1993, Applegate, Bixby, Chvátal, and Cook (see http://www.cs.rutgers.edu/~chvatal/pcb3038.html) found the solution to the largest TSPLIB[6] traveling salesman problem solved up until that time. It had 3,038 cities and arose from a practical problem involving the most efficient order in which to drill 3,038 holes to make a circuit board (another traveling salesman problem application). For information about this, see Zimmer [1993].[7] ■

The traveling salesman problem is an example of a problem that has defied the efforts of researchers to find a "good" algorithm. Indeed, it belongs to a class of problems known as *NP-complete* or *NP-hard problems*, problems for which it is unlikely there will be a good algorithm in a very precise sense of the word *good*. We return to this point in Section 2.18, where we define NP-completeness briefly and define an algorithm to be a *good algorithm* if its complexity function $f(n)$ is bounded by a polynomial in n. Such an algorithm is called a *polynomial algorithm* (more precisely, a polynomial-time algorithm).

Example 2.17 Scheduling a Computer System[8] A computer center has n programs to run. Each program requires certain resources, such as a compiler, a number of processors, and an amount of memory per processor. We shall refer to the required resources as a *configuration* corresponding to the program. The conversion of the system from the ith configuration to the jth configuration has a cost associated with it, say c_{ij}. For instance, if two programs require a similar configuration, it makes sense to run them consecutively. The computer center would like to minimize the total costs associated with running the n programs. The fixed cost of running each program does not change with different orders of running the programs. The only things that change are the conversion costs c_{ij}. Hence, the center wants to find an order in which to run the programs such that the total

[6] The TSPLIB (http://www.iwr.uni-heidelberg.de/iwr/comopt/software/TSPLIB95/) is a library of 110 instances of the traveling salesman problem.

[7] All instances in the TSPLIB library have been solved. The largest instance of the traveling salesman problem in TSPLIB consists of a tour through 85,900 cities in a VLSI (Very Large-Scale Integration) application. For a survey about the computational aspects of the traveling salesman problem, see Applegate, *et al.* [1998]. See also the Traveling Salesman Problem home page (http://www.tsp.gatech.edu/index.html).

[8] This example is due to Stanat and McAllister [1977].

conversion costs are minimized. Similar questions arise in many scheduling problems in operations research. We discuss them further in Example 11.5 and Section 11.6.3. As in the traveling salesman problem, the algorithm of enumerating all possible orders of running the programs is infeasible, for it clearly has a computational complexity of $n!$. [Why $n!$ and not $(n-1)!$?] Indeed, from a formal point of view, this problem and the traveling salesman problem are almost equivalent—simply replace cities by configurations. Any algorithm for solving one of these problems is readily translatable into an algorithm for solving the other problem. It is one of the major motivations for using mathematical techniques to solve real problems that we can solve one problem and then immediately have techniques that are applicable to a large number of other problems, which on the surface seem quite different. ■

Example 2.18 Searching Through a File In determining computational complexity, we do not always know exactly how long a computation will take. For instance, consider the problem of searching through a list of n keys (identification numbers) and finding the key of a particular person in order to access that person's file. Now it is possible that the key in question will be first in the list. However, in the *worst case*, the key will be last on the list. The cost of handling the worst possible case is sometimes used as a measure of computational complexity called the *worst-case complexity*. Here $f(n)$ would be proportional to n. On the other hand, another perfectly appropriate measure of computational complexity is the *average* cost of handling a case, the *average-case complexity*. Assuming that all cases are equally likely, this is computed by calculating the cost of handling each case, summing up these costs, and dividing by the number of cases. In our example, the average-case complexity is proportional to $(n+1)/2$, assuming that all keys are equally likely to be the object of a search, for the sum of the costs of handling the cases is given by $1+2+\cdots+n$. Hence, using a standard formula for this sum, we have

$$f(n) = \frac{1}{n}(1+2+\cdots+n) = \frac{1}{n}\frac{n(n+1)}{2} = \frac{n+1}{2}.$$ ■

In Section 3.6 we discuss the use of binary search trees for storing files and argue that the computational complexity of finding a file with a given key can be reduced significantly by using a binary search tree.

EXERCISES FOR SECTION 2.4

1. If a computer could consider 1 billion orders a second, how many years would it take to solve the computer configuration problem of Example 2.17 by enumeration if n is 25?
2. If a computer could consider 100 billion orders a second instead of just 1 billion, how many years would it take to solve the traveling salesman problem by enumeration if $n = 26$? (Does the improvement in computer speed make a serious difference in conclusions based on footnote 4 on page 26?)

3. Consider the problem of scheduling n legislative committees in order for meetings in n consecutive time slots. Each committee chair indicates which time slot is his or her first choice, and we seek to schedule the meetings so that the number of chairs receiving their first choice is as large as possible. Suppose that we solve this problem by enumerating all possible schedules, and for each we compute the number of chairs receiving their first choice. What is the computational complexity of this procedure? (Make an assumption about the number of steps required to compute the number of chairs receiving their first choice.)
4. Suppose that there are n phone booths in a region and we wish to visit each of them twice, but not in two consecutive times. Discuss the computational complexity of a naive algorithm for finding an order of visits that minimizes the total travel time.
5. Solve the traveling salesman problem by enumeration if $n = 4$ and the cost c_{ij} is given in the following matrix:

$$(c_{ij}) = \begin{pmatrix} - & 1 & 8 & 11 \\ 16 & - & 3 & 6 \\ 4 & 9 & - & 11 \\ 8 & 3 & 2 & - \end{pmatrix}.$$

6. Solve the computer system scheduling problem of Example 2.17 if $n = 3$ and the cost of converting from the ith configuration to the jth is given by

$$(c_{ij}) = \begin{pmatrix} - & 8 & 11 \\ 12 & - & 4 \\ 3 & 6 & - \end{pmatrix}.$$

7. Suppose that it takes 3×10^{-9} seconds to examine each key in a list. If there are n keys and we search through them in order until we find the right one, find

   (a) the worst-case complexity (b) the average-case complexity

8. Repeat Exercise 7 if it takes 3×10^{-11} seconds to examine each key.
9. (Hopcroft [1981]) Suppose that L is a collection of bit strings of length n. Suppose that A is an algorithm which determines, given a bit string of length n, whether or not it is in L. Suppose that A always takes 2^n seconds to provide an answer. Then A has the same worst-case and average-case computational complexity, 2^n. Suppose that $\widehat{L}$ consists of all bit strings of the form

$$x_1x_2\cdots x_nx_1x_2\cdots x_n,$$

where $x_1x_2\cdots x_n$ is in L. For instance, if $L = \{00, 10\}$, then $\widehat{L} = \{0000, 1010\}$. Consider the following Algorithm B for determining, given a bit string $y = y_1y_2\cdots y_{2n}$ of length $2n$, whether or not it is in $\widehat{L}$. First, determine if y is of the form $x_1x_2\cdots x_nx_1x_2\cdots x_n$. This is easy to check. Assume for the sake of discussion that it takes essentially 0 seconds to answer this question. If y is not of the proper form, stop and say that y is not in $\widehat{L}$. If y is of the proper form, check if the first n digits of y form a bit string in L.

   (a) Compute the worst-case complexity of Algorithm B.

   (b) Compute the average-case complexity of Algorithm B.

   (c) Do your answers suggest that average-case complexity might not be a good measure? Why?

2.5 r-PERMUTATIONS

Given an n-set, suppose that we want to pick out r elements and arrange them in order. Such an arrangement is called an *r-permutation of the n-set.* $P(n, r)$ will count the number of r-permutations of an n-set. For example, the number of 3-letter words without repeated letters can be calculated by observing that we want to choose 3 different letters out of 26 and arrange them in order; hence, we want $P(26, 3)$. Similarly, if a student has 4 experiments to perform and 10 periods in which to perform them (each experiment taking one period to complete), the number of different schedules he can make for himself is $P(10, 4)$. Note that $P(n, r) = 0$ if $n < r$: There are no r-permutations of an n-set in this case. In what follows, it will usually be understood that $n \geq r$.

To see how to calculate $P(n, r)$, let us note that in the case of the 3-letter words, there are 26 choices for the first letter; for each of these there are 25 remaining choices for the second letter; and for each of these there are 24 remaining choices for the third letter. Hence, by the product rule,

$$P(26, 3) = 26 \times 25 \times 24.$$

In the case of the experiment schedules, we have 10 choices for the first experiment, 9 for the second, 8 for the third, and 7 for the fourth, giving us

$$P(10, 4) = 10 \times 9 \times 8 \times 7.$$

By the same reasoning, if $n \geq r$,[9]

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

If $n > r$, this can be simplified as follows:

$$P(n, r) = \frac{[n \times (n-1) \times \cdots \times (n-r+1)] \times [(n-r) \times (n-r-1) \times \cdots \times 1]}{(n-r) \times (n-r-1) \times \cdots \times 1}.$$

Hence, we obtain the result

$$P(n, r) = \frac{n!}{(n-r)!}. \tag{2.1}$$

We have derived (2.1) under the assumption $n > r$. It clearly holds for $n = r$ as well. (Why?)

Example 2.19 CD Player We buy a brand new CD player with many nice features. In particular, the player has slots labeled 1 through 5 for five CDs which it plays in that order. If we have 24 CDs in our collection, how many different ways can we load the CD player's slots for our listening pleasure? There are 24 choices

[9] This formula even holds if $n < r$. Why?

for the first slot, 23 choices for the second, 22 choices for the third, 21 choices for the fourth, and 20 choices for the fifth, giving us

$$P(24,5) = 24 \times 23 \times 22 \times 21 \times 20.$$

Alternatively, using Equation (2.1), we see again that

$$P(24,5) = \frac{24!}{(24-5)!} = \frac{24!}{19!} = 24 \times 23 \times 22 \times 21 \times 20.$$

■

EXERCISES FOR SECTION 2.5

1. Find:

   (a) $P(3,2)$ (b) $P(5,3)$ (c) $P(8,5)$ (d) $P(1,3)$

2. Let $A = \{1,5,9,11,15,23\}$.

   (a) Find the number of sequences of length 3 using elements of A.

   (b) Repeat part (a) if no element of A is to be used twice.

   (c) Repeat part (a) if the first element of the sequence is 5.

   (d) Repeat part (a) if the first element of the sequence is 5 and no element of A is used twice.

3. Let $A = \{a,b,c,d,e,f,g,h\}$.

   (a) Find the number of sequences of length 4 using elements of A.

   (b) Repeat part (a) if no letter is repeated.

   (c) Repeat part (a) if the first letter in the sequence is b.

   (d) Repeat part (a) if the first letter is b and the last is d and no letters are repeated.

4. In how many different orders can we schedule the first five interviews if we need to schedule interviews with 20 job candidates?

5. If a campus telephone extension has four digits, how many different extensions are there with no repeated digits:

   (a) If the first digit cannot be 0?

   (b) If the first digit cannot be 0 and the second cannot be 1?

6. A typical combination[10] lock or padlock has 40 numbers on its dial, ranging from 0 to 39. It opens by turning its dial clockwise, then counterclockwise, then clockwise, stopping each time at specific numbers. How many different padlocks can a company manufacture?

---

[10] In Section 2.7 we will see that the term "combination" is not appropriate with regard to padlocks; "r-permutation" would be correct.

2.6 SUBSETS

Example 2.20 The Pizza Problem A pizza shop advertises that it offers over 500 varieties of pizza. The local consumer protection bureau is suspicious. At the pizza shop, it is possible to have on a pizza a choice of any combination of the following toppings:

pepperoni, mushrooms, peppers, olives, sausage,
anchovies, salami, onions, bacon.

Is the pizza shop telling the truth in its advertisements? We shall be able to answer this question with some simple applications of the product rule. ■

To answer the question raised in Example 2.20, let us consider the set $\{a, b, c\}$. Let us ask how many subsets there are of this set. The answer can be obtained by enumeration, and we find that there are 8 such subsets:

$$\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}.$$

The answer can also be obtained using the product rule. We think of building up a subset in steps. First, we think of either including element a or not. There are 2 choices. Then we either include element b or not. There are again 2 choices. Finally, we either include element c or not. There are again 2 choices. The total number of ways of building up the subset is, by the product rule,

$$2 \times 2 \times 2 = 2^3 = 8.$$

Similarly, the number of subsets of a 4-set is

$$2 \times 2 \times 2 \times 2 = 2^4 = 16,$$

and the number of subsets of an n-set is

$$\underbrace{2 \times 2 \times \cdots \times 2}_{n \text{ times}} = 2^n.$$

Do these considerations help with the pizza problem? We can think of a particular pizza as a subset of the set of toppings. Alternatively, we can think, for each topping, of either including it or not. Either way, we see that there are $2^9 = 512$ possible pizzas. Thus, the pizza shop has not advertised falsely.

EXERCISES FOR SECTION 2.6

1. Enumerate the 16 subsets of $\{a, b, c, d\}$.
2. A magazine subscription service deals with 35 magazines. A subscriber may order any number of them. The subscription service is trying to computerize its billing procedure and wishes to assign a different computer key (identification number) to two different people unless they subscribe to exactly the same magazines. How much storage is required; that is, how many different code numbers are needed?

3. If the pizza shop of Example 2.20 decides to always put onions and mushrooms on its pizzas, how many different varieties can the shop now offer?
4. Suppose that the pizza shop of Example 2.20 adds a new possible topping, sardines, but insists that each pizza either have sardines or have anchovies. How many possible varieties of pizza does the shop now offer?
5. If A is a set of 10 elements, how many nonempty subsets does A have?
6. If A is a set of 8 elements, how many subsets of more than one element does A have?
7. A *value function* on a set A assigns 0 or 1 to each subset of A.
   (a) If A has 3 elements, how many different value functions are there on A?
   (b) What if A has n elements?
8. In a simple game (see Section 2.15), every subset of players is identified as either winning or losing.
   (a) If there is no restriction on this identification, how many distinct simple games are there with 3 players?
   (b) With n players?

2.7 r-COMBINATIONS

An *r-combination of an n-set* is a selection of r elements from the set, which means that order does not matter. Thus, an r-combination is an r-element subset. $C(n, r)$ will denote the number of r-combinations of an n-set. For example, the number of ways to choose a committee of 3 from a set of 4 people is given by $C(4, 3)$. If the 4 people are Dewey, Evans, Grange, and Howe, the possible committees are

$$\{\text{Dewey, Evans, Grange}\}, \{\text{Howe, Evans, Grange}\},$$
$$\{\text{Dewey, Howe, Grange}\}, \{\text{Dewey, Evans, Howe}\}.$$

Hence, $C(4, 3) = 4$. We shall prove some simple theorems about $C(n, r)$. Note that $C(n, r)$ is 0 if $n < r$: There are no r-combinations of an n-set in this case. Henceforth, $n \geq r$ will usually be understood. It is assumed in all of the theorems in this section.

Theorem 2.1

$$P(n, r) = C(n, r) \times P(r, r).$$

Proof. An ordered arrangement of r objects out of n can be obtained by first choosing r objects [this can be done in $C(n, r)$ ways] and then ordering them [this can be done in $P(r, r) = r!$ ways]. The theorem follows by the product rule. Q.E.D.

Corollary 2.1.1

$$C(n, r) = \frac{n!}{r!(n-r)!}. \tag{2.2}$$

Proof.

$$C(n,r) = \frac{P(n,r)}{P(r,r)} = \frac{n!/(n-r)!}{r!/(r-r)!} = \frac{n!}{r!(n-r)!}. \qquad \text{Q.E.D.}$$

Corollary 2.1.2

$$C(n,r) = C(n, n-r).$$

Proof.

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

Q.E.D.

For an alternative "combinatorial" proof, see Exercise 20.

Note: The number

$$\frac{n!}{r!(n-r)!}$$

is often denoted by

$$\binom{n}{r}$$

and called a *binomial coefficient*. This is because, as we shall see below, this number arises in the binomial expansion (see Section 2.14). Corollary 2.1.2 states the result that

$$\binom{n}{r} = \binom{n}{n-r}.$$

In what follows we use $C(n,r)$ and $\binom{n}{r}$ interchangeably.

Theorem 2.2

$$C(n,r) = C(n-1, r-1) + C(n-1, r).$$

Proof. Mark one of the n objects with a $*$. The r objects can be selected either to include the object $*$ or not to include it. There are $C(n-1, r-1)$ ways to do the former, since this is equivalent to choosing $r-1$ objects out of the $n-1$ non-$*$ objects. There are $C(n-1, r)$ ways to do the latter, since this is equivalent to choosing r objects out of the $n-1$ non-$*$ objects. Hence, the sum rule yields the theorem. Q.E.D.

Note: This proof can be described as a "combinatorial" proof, i.e, relying on counting arguments. This theorem can also be proved by algebraic manipulation, using the formula (2.2). Here is such an "algebraic" proof.

Second Proof of Theorem 2.2.

$$\begin{aligned}
C(n-1,r-1)+C(n-1,r) &= \frac{(n-1)!}{(r-1)![(n-1)-(r-1)]!}+\frac{(n-1)!}{r![(n-1)-r]!}\\
&= \frac{(n-1)!}{(r-1)!(n-r)!}+\frac{(n-1)!}{r!(n-r-1)!}\\
&= \frac{r(n-1)!}{r!(n-r)!}+\frac{(n-r)(n-1)!}{r!(n-r)!}\\
&= \frac{r(n-1)!+(n-r)(n-1)!}{r!(n-r)!}\\
&= \frac{(n-1)![r+n-r]}{r!(n-r)!}\\
&= \frac{n!}{r!(n-r)!}\\
&= C(n,r).
\end{aligned}$$

Q.E.D.

Let us give some quick applications of our new formulas and our basic rules so far.

1. In the pizza problem (Example 2.20), the number of pizzas with exactly 3 different toppings is

$$C(9,3)=\frac{9!}{3!6!}=84.$$

2. The number of pizzas with at most 3 different toppings is, by the sum rule,

$$C(9,0)+C(9,1)+C(9,2)+C(9,3)=130.$$

3. If we have 6 drugs being tested in an experiment and we want to choose 2 of them to give to a particular subject, the number of ways in which we can do this is

$$C(6,2)=\frac{6!}{2!4!}=15.$$

4. If there are 7 possible meeting times and a committee must meet 3 times, the number of ways we can assign the meeting times is

$$C(7,3)=\frac{7!}{3!4!}=35.$$

5. The number of 5-member committees from a group of 9 people is

$$C(9,5)=126.$$

6. The number of 7-member committees from the U.S. Senate is
$$C(100, 7).$$

7. The number of delegations to the President consisting of 2 senators and 2 representatives is
$$C(100, 2) \times C(435, 2).$$

8. The number of 9-digit bit strings with 5 1's and 4 0's is
$$C(9, 5) = C(9, 4).$$
To see why, think of having 9 unknown digits and choosing 5 of them to be 1's (or 4 of them to be 0's).

A convenient method of calculating the numbers $C(n, r)$ is to use the array shown in Figure 2.4. The number $C(n, r)$ appears in the nth row, rth diagonal. Each element in a given position is obtained by summing the two elements in the row above it which are just to the left and just to the right. For example, $C(5, 2)$ is given by summing up the numbers 4 and 6, which are circled in Figure 2.4. The array of Figure 2.4 is called *Pascal's triangle*, after the famous French philosopher and mathematician Blaise Pascal. Pascal was one of the inventors of probability theory and discovered many interesting combinatorial techniques.

Why does Pascal's triangle work? The answer is that it depends on the relation
$$C(n, r) = C(n-1, r-1) + C(n-1, r). \tag{2.3}$$
This is exactly the relation that was proved in Theorem 2.2. The relation (2.3) is an example of a *recurrence relation*. We shall see many such relations later in the book, especially in Chapter 6, which is devoted entirely to this topic. Obtaining such relations allows one to reduce calculations of complicated numbers to earlier steps, and therefore allows the computation of these numbers in stages.

EXERCISES FOR SECTION 2.7

1. How many ways are there to choose 5 starters (independent of position) from a basketball team of 10 players?
2. How many ways can 7 award winners be chosen from a group of 50 nominees?
3. Compute:

   (a) $C(6, 3)$ (b) $C(7, 4)$ (c) $C(5, 1)$ (d) $C(2, 4)$
4. Find $C(n, 1)$.
5. Compute $C(5, 2)$ and check your answer by enumeration.
6. Compute $C(6, 2)$ and check your answer by enumeration.

	$r=0$	$r=1$	$r=2$	$r=3$	$r=4$	$r=5$	$r=6$
$n=0$	1						
$n=1$	1	1					
$n=2$	1	2	1				
$n=3$	1	3	3	1			
$n=4$	1	4	6	4	1		
$n=5$	1	5	10	10	5	1	
$n=6$	1	6	15	20	15	6	1

Figure 2.4: Pascal's triangle. The circled numbers are added to give $C(5,2)$.

7. Check by computation that:

   (a) $C(7,2) = C(7,5)$ (b) $C(6,4) = C(6,2)$

8. Extend Figure 2.4 by adding one more row.
9. Compute $C(5,3)$, $C(4,2)$, and $C(4,3)$ and verify that formula (2.3) holds.
10. Repeat Exercise 9 for $C(7,5)$, $C(6,4)$, and $C(6,5)$.
11. (a) In how many ways can 8 blood samples be divided into 2 groups to be sent to different laboratories for testing if there are 4 samples in each group? Assume that the laboratories are distinguishable.

    (b) In how many ways can 8 blood samples be divided into 2 groups to be sent to different laboratories for testing if there are 4 samples in each group? Assume that the laboratories are indistinguishable.

    (c) In how many ways can the 8 samples be divided into 2 groups if there is at least 1 item in each group? Assume that the laboratories are distinguishable.
12. A company is considering 6 possible new computer systems and its systems manager would like to try out at most 3 of them. In how many ways can the systems manager choose the systems to be tried out?
13. (a) In how many ways can 10 food items be divided into 2 groups to be sent to different laboratories for purity testing if there are 5 items in each group?

    (b) In how many ways can the 10 items be divided into 2 groups if there is at least 1 item in each group?
14. How many 8-letter words with no repeated letters can be constructed using the 26 letters of the alphabet if each word contains 3, 4, or 5 vowels?
15. How many odd numbers between 1000 and 9999 have distinct digits?
16. A fleet is to be chosen from a set of 7 different make foreign cars and 4 different make domestic cars. How many ways are there to form the fleet if:

    (a) The fleet has 5 cars, 3 foreign and 2 domestic?

(b) The fleet can be any size (except empty), but it must have equal numbers of foreign and domestic cars?

(c) The fleet has 4 cars and 1 of them must be a Chevrolet?

(d) The fleet has 4 cars, 2 of each kind, and a Chevrolet and Honda cannot both be in the fleet?

17. (a) A computer center has 9 different programs to run. Four of them use the language C++ and 5 use the language JAVA. The C++ programs are considered indistinguishable and so are the JAVA programs. Find the number of possible orders for running the programs if:

    i. There are no restrictions.

    ii. The C++ programs must be run consecutively.

    iii. The C++ programs must be run consecutively and the JAVA programs must be run consecutively.

    iv. The languages must alternate.

    (b) Suppose that the cost of switching from a C++ configuration to a JAVA configuration is 10 units, the cost of switching from a JAVA configuration to a C++ configuration is 5 units, and there is no cost to switch from C++ to C++ or JAVA to JAVA. What is the most efficient (least cost) ordering in which to run the programs?

    (c) Repeat part (a) if the C++ programs are all distinguishable from each other and so are the JAVA programs.

18. A certain company has 30 female employees, including 3 in the management ranks, and 150 male employees, including 12 in the management ranks. A committee consisting of 3 women and 3 men is to be chosen. How many ways are there to choose the committee if:

    (a) It includes at least 1 person of management rank of each gender?

    (b) It includes at least 1 person of management rank?

19. Consider the identity
$$\binom{n}{m}\binom{m}{k} = \binom{n}{k}\binom{n-k}{m-k}.$$

    (a) Prove this identity using an "algebraic" proof.

    (b) Prove this identity using a "combinatorial" proof.

20. Give an alternative "combinatorial" proof of Corollary 2.1.2 by using the definition of $C(n,r)$.

21. How would you find the sum $\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\cdots+\binom{n}{n}$ from Pascal's triangle? Do so for $n = 2$, 3, and 4. Guess at the answer in general.

22. Show that
$$\binom{n}{0}+\binom{n+1}{1}+\cdots+\binom{n+r}{r} = \binom{n+r+1}{r}.$$

23. Prove the following identity (using a combinatorial proof if possible). The identity is called *Vandermonde's identity.*

$$\binom{n+m}{r} = \binom{n}{0}\binom{m}{r} + \binom{n}{1}\binom{m}{r-1} + \binom{n}{2}\binom{m}{r-2} + \cdots + \binom{n}{r}\binom{m}{0}.$$

24. Following Cohen [1978], define $\left\langle {n \atop r} \right\rangle$ to be $\binom{n+r-1}{r}$. Show that

$$\left\langle {n \atop r} \right\rangle = \left\langle {n \atop r-1} \right\rangle + \left\langle {n-1 \atop r} \right\rangle$$

(a) using an algebraic proof (b) using a combinatorial proof

25. If $\left\langle {n \atop r} \right\rangle$ is defined as in Exercise 24, show that

$$\left\langle {n \atop r} \right\rangle = \frac{n}{r}\left\langle {n+1 \atop r-1} \right\rangle = \frac{n+r-1}{r}\left\langle {n \atop r-1} \right\rangle.$$

26. A sequence of numbers $a_0, a_1, a_2, \ldots, a_n$ is called *unimodal* if for some integer t, $a_0 \leq a_1 \leq \cdots \leq a_t$ and $a_t \geq a_{t+1} \geq \cdots \geq a_n$. (Note that the entries in any row of Pascal's triangle increase for awhile and then decrease and thus form a unimodal sequence.)

(a) Show that if $a_0, a_1, a_2, \ldots, a_n$ is unimodal, t is not necessarily unique.

(b) Show that if $n > 0$, the sequence $\binom{n}{0}, \binom{n}{1}, \binom{n}{2}, \ldots, \binom{n}{n}$ is unimodal.

(c) Show that the largest entry in the sequence in part (b) is $\binom{n}{\lfloor n/2 \rfloor}$ where $\lfloor x \rfloor$ is the greatest integer less than or equal to x.

2.8 PROBABILITY

The history of combinatorics is closely intertwined with the history of the theory of probability. The theory of probability was developed to deal with uncertain events, events that might or might not occur. In particular, this theory was developed by Pascal, Fermat, Laplace, and others in connection with the outcomes of certain gambles. In his *Théorie Analytique des Probabilités*, published in 1812, Laplace defined probability as follows: The *probability* of an event is the number of possible outcomes whose occurrence signals the event divided by the total number of possible outcomes. For instance, suppose that we consider choosing a 2-digit bit string at random. There are 4 such strings, 00, 01, 10, and 11. What is the probability that the string chosen has a 0? The answer is $\frac{3}{4}$, because 3 of the possible outcomes signal the event in question, that is, have a 0, and there are 4 possible outcomes in all. This definition of Laplace's is appropriate only if all the possible outcomes are equally likely, as we shall quickly observe.

Let us make things a little more precise. We shall try to formalize the notion of probability by thinking of an *experiment* that produces one of a number of possible outcomes. The set of possible outcomes is called the *sample space*. An *event* corresponds to a subset of the set of outcomes, that is, of the sample space; it corresponds to those outcomes that signal that the event has taken place. An event's *complement* corresponds to those outcomes that signal that the event has *not* taken place. Laplace's definition says that if E is an event in the sample space S and E^c is the complement of E, then

$$\text{probability of } E = \frac{n(E)}{n(S)} \text{ and probability of } E^c = \frac{n(S) - n(E)}{n(S)} = 1 - \frac{n(E)}{n(S)},$$

where $n(E)$ is the number of outcomes in E and $n(S)$ is the number of outcomes in S. Note that it follows that the probability of E is a number between 0 and 1.

Let us apply this definition to a gambling situation. We toss a die—this is the experiment. We wish to compute the probability that the outcome will be an even number. The sample space is the set of possible outcomes, $\{1, 2, 3, 4, 5, 6\}$. The event in question is the set of all outcomes which are even, that is, the set $\{2, 4, 6\}$. Then we have

$$\text{probability of even } = \frac{n(\{2,4,6\})}{n(\{1,2,3,4,5,6\})} = \frac{3}{6} = \frac{1}{2}.$$

Notice that this result would not hold unless all the outcomes in the sample space were equally likely. If we have a weighted die that always comes up 1, the probability of getting an even number is not $\frac{1}{2}$ but 0.[11]

Let us consider a family with two children. What is the probability that the family will have at least one boy? There are three possibilities for such a family: It can have two boys, two girls, or a boy and a girl. Let us take the set of these three possibilities as our sample space. The first and third outcomes make up the event "having at least one boy," and hence, by Laplace's definition,

$$\text{probability of having at least one boy } = \frac{2}{3}.$$

Is this really correct? It is not. If we look at families with two children, more than $\frac{2}{3}$ of them have at least one boy. That is because there are four ways to build up a family of two children: we can have first a boy and then another boy, first a girl and then another girl, first a boy and then a girl, or first a girl and then a boy. Thus, there are more ways to have a boy and a girl than there are ways to have two boys, and the outcomes in our sample space were not equally likely. However, the

[11] It could be argued that the definition of probability we have given is "circular" because it depends on the notion of events being "equally likely," which suggests that we already know how to measure probability. This is a subtle point. However, we can make comparisons of things without being able to measure them, e.g., to say that this person and that person seem equally tall. The theory of measurement of probability, starting with comparisons of this sort, is described in Fine [1973] and Roberts [1976, 1979].

outcomes BB, GG, BG, and GB, to use obvious abbreviations, are equally likely,[12] so we can take them as our sample space. Now the event "having at least one boy" has 3 outcomes in it out of 4, and we have

$$\text{probability of having at least one boy } = \frac{3}{4}.$$

We shall limit computations of probability in this book to situations where the outcomes in the sample space are equally likely. Note that our definition of probability applies only to the case where the sample space is finite. In the infinite case, the Laplace definition obviously has to be modified. For a discussion of the not-equally-likely case and the infinite case, the reader is referred to almost any textbook on probability theory, for instance Feller [1968], Parzen [1992], or Ross [1997].

Let us continue by giving several more applications of our definition. Suppose that a family is known to have 4 children. What is the probability that half of them are boys? The answer is not $\frac{1}{2}$. To obtain the answer we observe that the sample space is all sequences of B's and G's of length 4; a typical such sequence is BGGB. How many such sequences have exactly 2 B's? There are 4 positions, and 2 of these must be chosen for B's. Hence, there are $C(4,2)$ such sequences. How many sequences are there in all? By the product rule, there are 2^4. Hence,

$$\text{probability that half are boys } = \frac{C(4,2)}{2^4} = \frac{6}{16} = \frac{3}{8}.$$

The reader might wish to write out all 16 possible outcomes and note the 6 that signal the event having exactly 2 boys.

Next, suppose that a fair coin is tossed 5 times. What is the probability that there will be at least 2 heads? The sample space consists of all possible sequences of heads and tails of length 5, that is, it consists of sequences such as HHHTH, to use an obvious abbreviation. How many such sequences have at least 2 heads? The answer is that $C(5,2)$ sequences have exactly 2 heads, $C(5,3)$ have exactly 3 heads, and so on. Thus, the number of sequences having at least 2 heads is given by

$$C(5,2) + C(5,3) + C(5,4) + C(5,5) = 26.$$

The total number of possible sequences is $2^5 = 32$. Hence,

$$\text{probability of having at least two heads } = \frac{26}{32} = \frac{13}{16}.$$

Example 2.21 Reliability of Systems Imagine that a system has n components, each of which can work or fail to work. Let x_i be 1 if the ith component works and 0 if it fails. Let the bit string $x_1x_2\cdots x_n$ describe the system. Thus, the bit string 0011 describes a system with four components, with the first two failing

[12] Even this statement is not quite accurate, because it is slightly more likely to have a boy than a girl (see Cummings [1997]). Thus, the four events we have chosen are not exactly equally likely. For example, BB is more likely than GG. However, the assertion is a good working approximation.

Table 2.4: The Switching Function F That is 1 if and Only if Two or Three Components of a System Work

$x_1x_2x_3$	111	110	101	100	011	010	001	000
$F(x_1x_2x_3)$	1	1	1	0	1	0	0	0

and the third and fourth working. Since many systems have built-in redundancy, the system as a whole can work even if some components fail. Let $F(x_1x_2\cdots x_n)$ be 1 if the system described by $x_1x_2\cdots x_n$ works and 0 if it fails. Then F is a function from bit strings of length n to $\{0,1\}$, that is, an n-variable switching function (Example 2.4). For instance, suppose that we have a highly redundant system with three identical components, and the system works if and only if at least two components work. Then F is given by Table 2.4. We shall study other specific examples of functions F in Section 3.2.4 and Exercise 22, Section 13.3. Suppose that components in a system are equally likely to work or not to work.[13] Then any two bit strings are equally likely to be the bit string $x_1x_2\cdots x_n$ describing the system. Now we may ask: What is the probability that the system works, that is, what is the probability that $F(x_1x_2\cdots x_n)=1$? This is a measure of the *reliability* of the system. In our example, 4 of the 8 bit strings, 111, 110, 101, and 011, signal the event that $F(x_1x_2x_3)=1$. Since all bit strings are equally likely, the probability that the system works is $\frac{4}{8}=\frac{1}{2}$. For more on this approach to system reliability, see Karp and Luby [1983] and Barlow and Proschan [1975].

The theory of reliability of systems has been studied widely for networks of all kinds: electrical networks, computer networks, communication networks, and transportation routing networks. For a general reference on the subject of reliability of networks, see Hwang, Monma, and Roberts [1991] or Ball, Colbourn, and Provan [1995]. ■

Before closing this section, we observe that some common statements about probabilities of events correspond to operations on the associated subsets. Thus, we have:

Probability that event E does not occur is the probability of E^c.
Probability that event E or event F occurs is the probability of $E\cup F$.
Probability that event E and event F occur is the probability of $E\cap F$.

It is also easy to see from the definition of probability that

$$\text{probability of } E^c = 1 - \text{ probability of } E. \tag{2.4}$$

If E and F are disjoint,

$$\text{probability of } E\cup F = \text{ probability of } E + \text{probability of } F, \tag{2.5}$$

[13] In a more general analysis, we would first estimate the probability p_i that the ith component works.

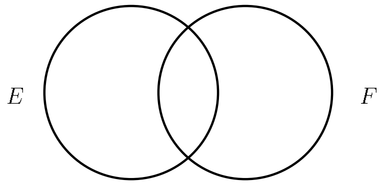

Figure 2.5: A Venn diagram related to Equation (2.6).

and, in general,

$$\begin{aligned}\text{probability of } E \cup F = &\ \text{probability of } E + \text{probability of } F\\ &- \text{ probability of } E \cap F.\end{aligned} \tag{2.6}$$

To see why Equation (2.6) is true, consider the Venn diagram in Figure 2.5. Notice that when adding the probability of E and the probability of F, we are adding the probability of the intersection of E and F twice. By subtracting the probability of the intersection of E and F from the sum of their probabilities, Equation (2.6) is obtained.

To illustrate these observations, let us consider the die-tossing experiment. Then the probability of not getting a 3 is 1 minus the probability of getting a 3; that is, it is $1 - \frac{1}{6} = \frac{5}{6}$. What is the probability of getting a 3 or an even number? Since $E = \{3\}$ and $F = \{2, 4, 6\}$ are disjoint, (2.5) implies it is probability of E plus probability of $F = \frac{1}{6} + \frac{3}{6} = \frac{2}{3}$. Finally, what is the probability of getting a number larger than 4 or an even number? The event in question is the set $\{2, 4, 5, 6\}$, which has probability $\frac{4}{6} = \frac{2}{3}$. Note that this is not the same as the probability of a number larger than 4 plus the probability of an even number $= \frac{2}{6} + \frac{3}{6} = \frac{5}{6}$. This is because $E = \{5, 6\}$ and $F = \{2, 4, 6\}$ are not disjoint, for $E \cap F = \{6\}$. Applying (2.6), we have probability of $E \cup F = \frac{2}{6} + \frac{3}{6} - \frac{1}{6} = \frac{2}{3}$, which agrees with our first computation.

Example 2.22 Food Allergies (Example 2.5 Revisited) In Example 2.5 we studied the switching functions associated with food allergies brought on by some combination of four foods: tomatoes, chocolate, corn, and peanuts. We saw that there are a total of $2^4 = 16$ possible food combinations. We considered the situation where a person develops an allergic reaction any time tomatoes are in the diet or corn and peanuts are in the diet. What is the probability of not having such an allergic reaction?

To find the probability in question, we first calculate the probability that there is a reaction. Note that there is a reaction if the foods present are represented by the bit string $(1, y, z, w)$ or the bit string $(x, y, 1, 1)$, where x, y, z, w are (binary) 0-1 variables. Since there are three binary variables in the first type and two binary variables in the second type, there are $2^3 = 8$ different bit strings of the first type and $2^2 = 4$ of the second type. If there was no overlap between the two types, then (2.5) would allow us to merely add probabilities. However, there is overlap when x, z, and w are all 1. In this case y could be 0 or 1. Thus, by (2.6), the probability of a food reaction is

$$\frac{8}{16} + \frac{4}{16} - \frac{2}{16} = \frac{10}{16} = \frac{5}{8}.$$

By (2.4), the probability of no reaction is $1 - \frac{5}{8} = \frac{3}{8}$. ■

In the Example 2.22, enumeration of the possible combinations would be an efficient solution technique. However, if as few as 10 foods are considered, then enumeration would begin to get unwieldy. Thus, the techniques developed and used in this section are essential to avoid enumeration.

EXERCISES FOR SECTION 2.8

1. Are the outcomes in the following experiments equally likely?
   (a) A citizen of California is chosen at random and his or her town of residence is recorded.
   (b) Two drug pills and three placebos (sugar pills) are placed in a container and one pill is chosen at random and its type is recorded.
   (c) A snowflake is chosen at random and its appearance is recorded.
   (d) Two fair dice are tossed and the sum of the numbers appearing is recorded.
   (e) A bit string of length 3 is chosen at random and the sum of its digits is observed.
2. Calculate the probability that when a die is tossed, the outcome will be:
   (a) An odd number
   (b) A number less than or equal to 2
   (c) A number divisible by 3
3. Calculate the probability that a family of 3 children has:
   (a) Exactly 2 boys
   (b) At least 2 boys
   (c) At least 1 boy and at least 1 girl
4. If black hair, brown hair, and blond hair are equally likely (and no other hair colors can occur), what is the probability that a family of 3 children has at least two blondes?
5. Calculate the probability that in four tosses of a fair coin, there are at most three heads.
6. Calculate the probability that if a DNA chain of length 5 is chosen at random, it will have at least four A's.
7. If a card is drawn at random from a deck of 52, what is the probability that it is a king or a queen?
8. Suppose that a card is drawn at random from a deck of 52, the card is replaced, and then another card is drawn at random. What is the probability of getting two kings?
9. If a bit string of length 4 is chosen at random, what is the probability of having at least three 1's?
10. What is the probability that a bit string of length 3, chosen at random, does not have two consecutive 0's?

11. Suppose that a system has four independent components, each of which is equally likely to work or not to work. Suppose that the system works if and only if at least three components work. What is the probability that the system works?
12. Repeat Exercise 11 if the system works if and only if the fourth component works and at least two of the other components work.
13. A medical lab can operate only if at least one licensed x-ray technician is present and at least one phlebotomist. There are three licensed x-ray technicians and two phlebotomists, and each worker is equally likely to show up for work on a given day or to stay home. Assuming that each worker decides independently whether or not to come to work, what is the probability that the lab can operate?
14. Suppose that we have 10 different pairs of gloves. From the 20 gloves, 4 are chosen at random. What is the probability of getting at least one pair?
15. Use rules (2.4)–(2.6) to calculate the probability of getting, in six tosses of a fair coin:

    (a) Two heads or three heads (b) Two heads or two tails

    (c) Two heads or a head on the first toss

    (d) An even number of heads or at least nine heads

    (e) An even number of heads and a head on the first toss

16. Use the definition of probability to verify rules:

    (a) (2.4) (b) (2.5) (c) (2.6)

17. Repeat the problem in Example 2.22 when allergic reactions occur only in diets:

    (a) Containing either tomatoes and corn or chocolate and peanuts

    (b) Containing either tomatoes or all three other foods

2.9 SAMPLING WITH REPLACEMENT

In the National Hockey League (NHL), a team can either win (W), lose (L), or lose in overtime (OTL) each of its games. In an 82-game schedule, how many different seasons[14] can a particular team have? By the product rule, the answer is 3^{82}. There are three possibilities for each of the 82 games: namely, W, L, or OTL. We say that we are *sampling with replacement*. We are choosing an 82-permutation out of a 3-set, {W, L, OTL}, but with replacement of the elements in the set after they are drawn. Equivalently, we are allowing repetition. Let $P^R(m,r)$ be the number of r-permutations of an m-set, with replacement or repetition allowed. Then the product rule gives us

$$P^R(m,r) = m^r. \tag{2.7}$$

The number $P(m,r)$ counts the number of r-permutations of an m-set if we are sampling without replacement or repetition.

[14] Do not confuse "seasons" with "records." Records refer to the final total of wins, losses, and ties while seasons counts the number of different ways that each record could be attained.

We can make a similar distinction in the case of r-combinations. Let $C^R(m,r)$ be the number of r-combinations of an m-set if we sample with replacement or repetition. For instance, the 4-combinations of a 2-set $\{a,b\}$ if replacement is allowed are given by

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

Thus, $C^R(2,4)=5$. We now state a formula for $C^R(m,r)$.

Theorem 2.3

$$C^R(m,r) = C(m+r-1,r).$$

We shall prove Theorem 2.3 at the end of this section. Here, let us illustrate it with some examples.

Example 2.23 The Chocolate Shoppe Suppose that there are three kinds of truffles available at a chocolate shoppe: cherry (c), orange (o), and vanilla (v). The store allows a customer to design a box of chocolates by choosing a dozen truffles. How many different truffle boxes are there? We can think of having a 3-set, $\{c,o,v\}$, and picking a 12-combination from it, with replacement. Thus, the number of truffle boxes is

$$C^R(3,12) = C(3+12-1,12) = C(14,12) = 91. \qquad \blacksquare$$

Example 2.24 DNA Strings: Gamow's Encoding In Section 2.1 we studied DNA strings on the alphabet $\{\mathrm{A},\mathrm{C},\mathrm{G},\mathrm{T}\}$ and the minimum length required for such a string to encode for an amino acid. We noted that there are 20 different amino acids, and showed in Section 2.1 that there are only 16 different DNA strings of length 2, so a string of length at least 3 is required. But there are $4^3 = 64$ different strings of length 3. Gamow [1954a,b] suggested that there was a relationship between amino acids and the rhombus-shaped "holes" formed by the bases in the double helix structure of DNA. Each rhombus (see Figure 2.6) consists of 4 bases, with one base located at each corner of the rhombus. We will identify each rhombus with its 4-base sequence $xyzw$ that starts at the top of the rhombus and continues clockwise around the rhombus. For example, the rhombus in Figure 2.6 would be written GTTA. Due to base pairing in DNA, the fourth base, w, in the sequence is always fixed by the second base, y, in the sequence. If the second base is T, then the fourth base is A (and vice versa), or if the second base is G, then the fourth base is C (and vice versa).

Gamow proposed that rhombus $xyzw$ encodes the same amino acid as (a) $xwzy$ and (b) $zyxw$. Thus, GTTA, GATT, TTGA, and TAGT would all encode the same amino acid. If Gamow's suggestion were correct, how many amino acids could be encoded using 4-base DNA rhombuses? In the 4-base sequence $xyzw$ there are only 2 choices for the y-w pair: A-T (or equivalently, T-A) and G-C (or equivalently, C-G). Picking the other two bases is an application of Theorem 2.3. We have $m=4$ objects, we wish to choose $r=2$ objects, with replacement, and order doesn't matter. This can be done in

$$C(m+r-1,r) = C(5,2) = 10$$

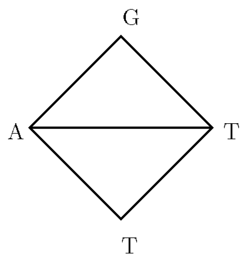

Figure 2.6: A 4-base DNA rhombus.

Table 2.5: Choosing a Sample of r Elements from a Set of m Elements

Order counts?	Repetition allowed?	The sample is called:	Number of ways to choose the sample	Reference
No	No	r-combination	$C(m,r) = \dfrac{m!}{r!(m-r)!}$	Corollary 2.1.1
Yes	No	r-permutation	$P(m,r) = \dfrac{m!}{(m-r)!}$	Eq. (2.1)
No	Yes	r-combination with replacement	$C^R(m,r) = C(m+r-1,r)$	Theorem 2.3
Yes	Yes	r-permutation with replacement	$P^R(m,r) = m^r$	Eq. (2.7)

different ways. Thus, it would be possible to encode $2 \times 10 = 20$ different amino acids using Gamow's 4-base DNA rhombuses which is precisely the correct number. Unfortunately, it was later discovered that this is not the way the coding works. See Golomb [1962] for a discussion. See also Griffiths, *et al.* [1996]. ■

Our discussion of sampling with and without replacement is summarized in Table 2.5.

Example 2.25 Voting Methods In most elections in the United States, a number of candidates are running for an office and each registered voter may vote for the candidate of his or her choice. The winner of the election is the candidate with the highest vote total. (There could be multiple winners in case of a tie, but then tie-breaking methods could be used.) This voting method is called *plurality voting.* Suppose that 3 juniors are running for student class president of a class of 400 students. How many different results are possible if everyone votes? By "different results" we are referring to the number of different "patterns" of vote totals obtained by the 3 candidates. A *pattern* is a sequence (a_1, a_2, a_3) where a_i is the number of votes obtained by candidate i, $i = 1, 2, 3$. Thus, $(6, 55, 339)$ is different from $(55, 339, 6)$, and $(6, 55, 338)$ is not possible. [We will make no distinction among the voters (i.e., who voted for whom), only in the vote totals for each candidate.] Again, this is an application of Theorem 2.3. We have $m = 3$ objects and we wish to choose $r = 400$ objects, with replacement (obviously). This can be done in

$$C(m + r - 1, r) = C(402, 400) = 80{,}601$$

different ways. This answer assumes that each voter voted. Exercise 10 addresses the question of vote totals when not all voters necessarily vote.

Another voting method, called *cumulative voting*, can be used in elections where more than one candidate needs to be elected. This is the case in many city council, board of directors, and school board elections. (Cumulative voting was used to elect the Illinois state legislature from 1870 to 1980.) With cumulative voting, voters cast as many votes as there are open seats to be filled and they are not limited to giving all of their votes to a single candidate. Instead, they can put multiple votes on one or more candidates. In such an election with p candidates, q open seats, and r voters, a total of qr votes are possible. The winners, analogous to the case of plurality voting, are the candidates with the q largest vote totals. Again consider the school situation of 3 candidates and 400 voters. However, now suppose that the students are not voting to elect a junior class president but two co-presidents. Under the cumulative voting method, how many different vote totals are possible? If, as in the plurality example above, each voter is required to vote for at least one candidate, then at least 400 votes and at most $2 \cdot 400 = 800$ votes must be cast. Consider the case of j votes being cast where $400 \leq j \leq 800$. By Theorem 2.3, there are

$$C^R(3, j) = \binom{3+j-1}{3-1} = \binom{2+j}{2}$$

different vote totals. Since j can range from 400 to 800, using the sum rule, there are a total of

$$\binom{2+400}{2} + \binom{2+401}{2} + \cdots + \binom{2+800}{2} = 75{,}228{,}001$$

different vote totals. Cumulative voting with votes not required and other voting methods are addressed in the exercises. For a general introduction to the methods and mathematics of voting see Aumann and Hart [1998], Brams [1994], Brams and Fishburn [1983], Farquharson [1969], or Kelly [1987]. ∎

Proof of Theorem 2.3.[15] Suppose that the m-set has elements $a_1, a_2, \ldots, a_m$. Then any sample of r of these objects can be described by listing how many a_1's are in it, how many a_2's, and so on. For instance, if $r = 7$ and $m = 5$, typical samples are $a_1a_1a_2a_3a_4a_4a_5$ and $a_1a_1a_1a_2a_4a_5a_5$. We can also represent these samples by putting a vertical line after the last a_i, for $i = 1, 2, \ldots, m-1$. Thus, these two samples would be written as $a_1a_1 \mid a_2 \mid a_3 \mid a_4a_4 \mid a_5$ and $a_1a_1a_1 \mid a_2 \mid\mid a_4 \mid a_5a_5$, where in the second case we have two consecutive vertical lines since there is no a_3. Now if we use this notation to describe a sample of r objects, we can omit the subscripts. For instance, $aa \mid aa \mid\mid\mid aaa$ represents $a_1a_1 \mid a_2a_2 \mid\mid\mid a_5a_5a_5$. Then the number of samples of r objects is just the number of different arrangements of r letters a and $m-1$ vertical lines. Such an arrangement has $m+r-1$ elements, and we determine the arrangement by choosing r positions for the a's. Hence, there are $C(m+r-1, r)$ such arrangements. Q.E.D.

[15] The proof may be omitted.

EXERCISES FOR SECTION 2.9

1. If replacement is allowed, find all:

   (a) 5-permutations of a 2-set
   (b) 2-permutations of a 3-set
   (c) 5-combinations of a 2-set
   (d) 2-combinations of a 3-set

2. Check your answers in Exercise 1 by using Equation (2.7) or Theorem 2.3.
3. If replacement is allowed, compute the number of:

   (a) 7-permutations of a 3-set
   (b) 7-combinations of a 4-set

4. In how many ways can we choose eight concert tickets if four concerts are available?
5. In how many different ways can we choose 12 microwave desserts if 5 different varieties are available?
6. Suppose that a codeword of length 8 consists of letters A, B, or C or digits 0 or 1, and cannot start with 1. How many such codewords are there?
7. How many DNA chains of length 6 have at least one of each base T, C, A, and G? Answer this question under the following assumptions:

   (a) Only the number of bases of a given type matter.
   (b) Order matters.

8. In an 82-game NHL season, how many different final records[16] are possible:

   (a) If a team can either win, lose, or overtime lose each game?
   (b) If overtime losses are not possible?

9. The United Soccer League in the United States has a shootout if a game is tied at the end of regulation. So there are wins, shootout wins, losses, or shootout losses. How many different 12-game seasons are possible?
10. Calculate the number of different vote totals, using the plurality voting method (see Example 2.25), when there are m candidates and n voters and each voter need not vote.
11. Calculate the number of different vote totals, using the cumulative voting method (see Example 2.25), when there are m candidates, n voters, l open seats, and each voter need not vote.

2.10 OCCUPANCY PROBLEMS[17]

2.10.1 The Types of Occupancy Problems

In the history of combinatorics and probability theory, problems of placing *balls* into *cells* or *urns* have played an important role. Such problems are called *occupancy problems.* Occupancy problems have numerous applications. In classifying

[16] See footnote on page 47.

[17] For a quick reading of this section, it suffices to read Section 2.10.1.

Table 2.6: The Distributions of Two Distinguishable Balls to Three Distinguishable Cells

		Distribution								
		1	2	3	4	5	6	7	8	9
	1	ab			a	a		b	b	
Cell	2		ab		b		a	a		b
	3			ab		b	b		a	a

types of accidents according to the day of the week in which they occur, the balls are the types of accidents and the cells are the days of the week. In cosmic-ray experiments, the balls are the particles reaching a Geiger counter and the cells are the counters. In coding theory, the possible distributions of transmission errors on k codewords are obtained by studying the codewords as cells and the errors as balls. In book publishing, the possible distributions of misprints on k pages are obtained by studying the pages as cells and the balls as misprints. In the study of irradiation in biology, the light particles hitting the retina correspond to balls, the cells of the retina to the cells. In coupon collecting, the balls correspond to particular coupons, the cells to the types of coupons. We shall return in various places to these applications. See Feller [1968, pp. 10–11] for other applications.

In occupancy problems, it makes a big difference whether or not we regard two balls as distinguishable and whether or not we regard two cells as distinguishable. For instance, suppose that we have two distinguishable balls, a and b, and three distinguishable cells, 1, 2, and 3. Then the possible distributions of balls to cells are shown in Table 2.6. There are nine distinct distributions. However, suppose that we have two indistinguishable balls. We can label them both a. Then the possible distributions to three distinguishable cells are shown in Table 2.7. There are just six of them. Similarly, if the cells are not distinguishable but the balls are, distributions 1–3 of Table 2.6 are considered the same: two balls in one cell, none in the others. Similarly, distributions 4–9 are considered the same: two cells with one ball, one cell with no balls. There are then just two distinct distributions. Finally, if neither the balls nor the cells are distinguishable, then distributions 1–3 of Table 2.7 are considered the same and distributions 4–6 are as well, so there are two distinct distributions.

It is also common to distinguish between occupancy problems where the cells are allowed to be empty and those where they are not. For instance, if we have two distinguishable balls and two distinguishable cells, then the possible distributions are given by Table 2.8. There are four of them. However, if no cell can be empty, there are only two, distributions 3 and 4 of Table 2.8.

The possible cases of occupancy problems are summarized in Table 2.9. The notation and terminology in the fourth column, which has not yet been defined, will be defined below. We shall now discuss the different cases.

Table 2.7: The Distributions of Two Indistinguishable Balls to Three Distinguishable Cells

		Distribution 1	2	3	4	5	6
Cell	1	aa			a	a	
	2		aa		a		a
	3			aa		a	a

Table 2.8: The Distributions of Two Distinguishable Balls to Two Distinguishable Cells

		Distribution 1	2	3	4
Cell	1	ab		a	b
	2		ab	b	a

2.10.2 Case 1: Distinguishable Balls and Distinguishable Cells

Case 1a is covered by the product rule: There are k choices of cells for each ball. If $k = 3$ and $n = 2$, we get $k^n = 9$, which is the number of distributions shown in Table 2.6. Case 1b is discussed in Section 2.10.4.

2.10.3 Case 2: Indistinguishable Balls and Distinguishable Cells[18]

Case 2a follows from Theorem 2.3, for we have the following result.

Theorem 2.4 The number of ways to distribute n indistinguishable balls into k distinguishable cells is $C(k+n-1, n)$.

Proof. Suppose that the cells are labeled $C_1, C_2, \ldots, C_k$. A distribution of balls into cells can be summarized by listing for each ball the cell into which it goes. Then, a distribution corresponds to a collection of n cells with repetition allowed. For instance, in Table 2.7, distribution 1 corresponds to the collection $\{C_1, C_1\}$ and distribution 5 to the collection $\{C_1, C_3\}$. If there are four balls, the collection $\{C_1, C_2, C_3, C_3\}$ corresponds to the distribution that puts one ball into cell C_1, one ball into cell C_2, and two balls into cell C_3. Because a distribution corresponds to a collection $C_{i_1}, C_{i_2}, \ldots, C_{i_n}$, the number of ways to distribute the balls into cells is the same as the number of n-combinations of the k-set $\{C_1, C_2, \ldots, C_k\}$ in which repetition is allowed. This is given by Theorem 2.3 to be $C(k+n-1, n)$. Q.E.D.

[18] The rest of Section 2.10 may be omitted.

Table 2.9: Classification of Occupancy Problems

	Distinguished balls?	Distinguished cells?	Can cells be empty?	Number of ways to place n balls into k cells:
Case 1				
1a	Yes	Yes	Yes	k^n
1b	Yes	Yes	No	$k!S(n,k)$
Case 2				
2a	No	Yes	Yes	$C(k+n-1,n)$
2b	No	Yes	No	$C(n-1,k-1)$
Case 3				
3a	Yes	No	Yes	$S(n,1)+S(n,2)+\cdots+S(n,k)$
3b	Yes	No	No	$S(n,k)$
Case 4				
4a	No	No	Yes	Number of partitions of n into k or fewer parts
4b	No	No	No	Number of partitions of n into exactly k parts

Theorem 2.4 is illustrated by Table 2.7. We have $k = 3, n = 2$, and $C(k+n-1,n) = C(4,2) = 6$. The result in case 2b now follows from the result in case 2a. Given n indistinguishable balls and k distinguishable cells, we first place one ball in each cell. There is one way to do this. It leaves $n-k$ indistinguishable balls. We wish to place these into k distinguishable cells, with no restriction as to cells being nonempty. By Theorem 2.4 this can be done in

$$C(k+(n-k)-1, n-k) = C(n-1, k-1)$$

ways. We now use the product rule to derive the result for case 2b of Table 2.9. Note that $C(n-1,k-1)$ is 0 if $n<k$. There is no way to assign n balls to k cells with at least one ball in each cell.

2.10.4 Case 3: Distinguishable Balls and Indistinguishable Cells

Let us turn next to case 3b. Let $S(n,k)$ be defined to be the number of ways to distribute n distinguishable balls into k indistinguishable cells with no cell empty. The number $S(n,k)$ is called a *Stirling number of the second kind.*[19] In Section 5.5.3 we show that

$$S(n,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^i\binom{k}{i}(k-i)^n. \tag{2.8}$$

[19] A Stirling number of the first kind exists and is found in other contexts. See Exercise 24 of Section 3.4.

To illustrate this result, let us consider the case $n = 2, k = 2$. Then

$$S(n,k) = S(2,2) = \frac{1}{2}[2^2 - 2\cdot 1^2 + 0] = 1.$$

There is only one distribution of two distinguishable balls a and b to two indistinguishable cells such that each cell has at least one ball: one ball in each cell.

The result in case 3a now follows from the result in case 3b by the sum rule. For to distribute n distinguishable balls into k indistinguishable cells with no cell empty, either one cell is not empty or two cells are not empty or .... The result in case 1b now follows also, since putting n distinguishable balls into k distinguishable cells with no cells empty can be accomplished by putting n distinguishable balls into k indistinguishable cells with no cells empty [which can be done in $S(n,k)$ ways] and then labeling the cells (which can be done in $k!$ ways). For instance, if $k = n = 2$, then by our previous computation, $S(2,2) = 1$. Thus, the number of ways to put two distinguishable balls into two distinguishable cells with no cells empty is $2!S(2,2) = 2$. This is the observation we made earlier from Table 2.8.

2.10.5 Case 4: Indistinguishable Balls and Indistinguishable Cells

To handle cases 4a and 4b, we define a *partition* of a positive integer n to be a collection of positive integers that sum to n. For instance, the integer 5 has the partitions

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

Note that $\{3,2\}$ is considered the same as $\{2,3\}$. We are interested only in what integers are in the collection, not in their order. The number of ways to distribute n indistinguishable balls into k indistinguishable cells is clearly the same as the number of ways to partition the integer n into at most k parts. This gives us the result in case 4a of Table 2.9. For instance, if $n = 5$ and $k = 3$, there are five possible partitions, all but the first two listed above. If $n = 2$ and $k = 3$, there are two possible partitions, $\{1,1\}$ and $\{2\}$. This corresponds in Table 2.7 to the two distinct distributions: two cells with one ball in each or one cell with two balls in it. The result in case 4b of Table 2.9 follows similarly: The number of ways to distribute n indistinguishable balls into k indistinguishable cells with no cell empty is clearly the same as the number of ways to partition the integer n into exactly k parts. To illustrate this, if $n = 2$ and $k = 3$, there is no way.

We will explore partitions of integers briefly in the exercises and return to them in the exercises of Sections 5.3 and 5.4, where we approach them using the method of generating functions. For a detailed discussion of partitions, see most number theory books, for instance, Niven [1991] or Hardy and Wright [1980]. See also Berge [1971] or Riordan [1980].

2.10.6 Examples

We now give a number of examples, applying the results of Table 2.9. The reader should notice that whether or not balls or cells are distinguishable is often a matter of judgment, depending on the interpretation and in what we are interested.

Example 2.26 Hospital Deliveries Suppose that 80 babies are born in the month of September in a hospital and we record the day each baby is born. In how many ways can this event occur? The babies are the balls and the days are the cells. If we do not distinguish between 2 babies but do distinguish between days, we are in case 2, $n = 80, k = 30$, and the answer is given by $C(109, 80)$. The answer is given by $C(79, 29)$ if we count only the number of ways this can happen with each day having at least 1 baby. If we do not care about what day a particular number of babies is born but only about the number of days in which 2 babies are born, the number in which 3 are born, and so on, we are in case 4 and we need to consider partitions of the integer 80 into 30 or fewer parts. ■

Example 2.27 Coding Theory In coding theory, messages are first encoded into coded messages and then sent through a transmission channel. The channel may be a telephone line or radio wave. Due to noise or weak signals, errors may occur in the received codewords. The received codewords must then be decoded into the (hopefully) original messages. (An introduction to cryptography with an emphasis on coding theory is contained in Chapter 10.)

In monitoring the reliability of a transmission channel, suppose that we keep a record of errors. Suppose that 100 coded messages are sent through a transmission channel and 30 errors are made. In how many ways could this happen? The errors are the balls and the codewords are the cells. It seems reasonable to disregard the distinction between errors and concentrate on whether more errors occur during certain time periods of the transmission (because of external factors or a higher load period). Then codewords are distinguished. Hence, we are in case 2, and the answer is given by $C(129, 30)$. ■

Example 2.28 Gender Distribution Suppose that we record the gender of the first 1000 people to get a degree in computer science at a school. The people correspond to the balls and the two genders are the cells. We certainly distinguish cells. If we distinguish individuals, that is, if we distinguish between individual 1 being male and individual 2 being male, for example, then we are in case 1. However, if we are interested only in the number of people of each gender, we are in case 2. In the former case, the number of possible distributions is 2^{1000}. In the latter case, the number of possible distributions is given by $C(1001, 1000) = 1001$. ■

Example 2.29 Auditions A director has called back 24 actors for 8 different touring companies of a "one-man" Broadway show. (More than one actor may be chosen for a touring company in case of the need for a stand-in.) The actors correspond to the balls and the touring companies to the cells. If we are interested only in the actors who are in the same touring company, we can consider the balls distinguishable and the cells indistinguishable. Since each touring company needs at least one actor, no cell can be empty. Thus we are in case 3. The number of possible distributions is given by $S(24,8)$. ■

Example 2.30 Statistical Mechanics In statistical mechanics, suppose that we have a system of t particles. Suppose that there are p different states or levels (e.g., energy levels), in which each of the particles can be. The state of the system is described by giving the distribution of particles to levels. In all, if the particles are distinguishable, there are p^t possible distributions. For instance, if we have 4 particles and 3 levels, there are $3^4 = 81$ different arrangements. One of these has particle 1 at level 1, particle 2 at level 3, particle 3 at level 2, and particle 4 at level 3. Another has particle 1 at level 2, particle 2 at level 1, and particles 3 and 4 at level 3. If we consider any distribution of particles to levels to be equally likely, then the probability of any given arrangement is $1/p^t$. In this case we say that the particles obey the *Maxwell-Boltzmann statistics.* Unfortunately, apparently no known physical particles exhibit these Maxwell-Boltzmann statistics; the p^t different arrangements are not equally likely. It turns out that for many different particles, in particular photons and nuclei, a relatively simple change of assumption gives rise to an empirically accurate model. Namely, suppose that we consider the particles as indistinguishable. Then we are in case 2: Two arrangements of particles to levels are considered the same if the same number of particles is assigned to the same level. Thus, the two arrangements described above are considered the same, as they each assign one particle to level 1, one to level 2, and two to level 3. By Theorem 2.4, the number of distinguishable ways to arrange t particles into p levels is now given by $C(p+t-1,t)$. If we consider any distribution of particles to levels to be equally likely, the probability of any one arrangement is

$$\frac{1}{C(p+t-1,t)}.$$

In this case, we say that the particles satisfy the *Bose-Einstein statistics.* A third model in statistical mechanics arises if we consider the particles indistinguishable but add the assumption that there can be no more than two particles at a given level. Then we get the *Fermi-Dirac statistics* (see Exercise 21). See Feller [1968] or Parzen [1992] for a more detailed discussion of all the cases we have described. ■

EXERCISES FOR SECTION 2.10

Note to the reader: When it is unclear whether balls or cells are distinguishable, you should state your interpretation, give a reason for it, and then proceed.

1. Write down all the distributions of:

   (a) 3 distinguishable balls a, b, c into 2 distinguishable cells $1, 2$

   (b) 4 distinguishable balls a, b, c, d into 2 distinguishable cells $1, 2$

(c) 2 distinguishable balls a, b into 4 distinguishable cells $1, 2, 3, 4$

(d) 3 indistinguishable balls a, a, a into 2 distinguishable cells $1, 2$

(e) 4 indistinguishable balls a, a, a, a into 2 distinguishable cells $1, 2$

(f) 2 indistinguishable balls a, a into 4 distinguishable cells $1, 2, 3, 4$

2. In Exercise 1, which of the distributions are distinct if the cells are indistinguishable?
3. Use the results of Table 2.9 to compute the number of distributions in each case in Exercise 1 and check the result by comparing the distributions you have written down.
4. Repeat Exercise 3 if the cells are indistinguishable.
5. Use the results of Table 2.9 to compute the number of distributions with no empty cell in each case in Exercise 1. Check the result by comparing the distributions you have written down.
6. Repeat Exercise 5 if the cells are indistinguishable.
7. Find all partitions of:

   (a) 4 (b) 7 (c) 8

8. Find all partitions of:

   (a) 9 into four or fewer parts (b) 11 into three or fewer parts

9. Compute:

   (a) $S(n, 0)$ (b) $S(n, 1)$ (c) $S(n, 2)$
   (d) $S(n, n-1)$ (e) $S(n, n)$

10. In checking the work of a proofreader, we look for 5 kinds of misprints in a textbook. In how many ways can we find 12 misprints?
11. In Exercise 10, suppose that we do not distinguish the types of misprints but we do keep a record of the page on which a misprint occurred. In how many different ways can we find 25 misprints in 75 pages?
12. In Example 2.27, suppose that we pinpoint 30 kinds of errors and we want to find out whether these errors tend to appear together, not caring in which codeword they appear together. In how many ways can we find 30 kinds of errors in 100 codewords if each kind of error is known to appear exactly once in some codeword?
13. An elevator with 9 passengers stops at 5 different floors. If we are interested only in the passengers who get off together, how many possible distributions are there?
14. If lasers are aimed at 5 tumors, how many ways are there for 10 lasers to hit? (You do not have to assume that each laser hits a tumor.)
15. A Geiger counter records the impact of 6 different kinds of radioactive particles over a period of time. How many ways are there to obtain a count of 30?
16. Find the number of ways to distribute 10 customers to 7 salesmen so that each salesman gets at least 1 customer.
17. Find the number of ways to pair off 10 students into lab partners.
18. Find the number of ways to assign 6 jobs to 4 workers so that each job gets a worker and each worker gets at least 1 job.

19. Find the number of ways to partition a set of 20 elements into exactly 4 subsets.
20. In Example 2.30, suppose that there are 8 photons and 4 energy levels, with 2 photons at each energy level. What is the probability of this occurrence under the assumption that the particles are indistinguishable (the Bose-Einstein case)?
21. Show that in Example 2.30, if particles are indistinguishable but no two particles can be at the same level, then there are $C(p, t)$ possible arrangements of t particles into p levels. (Assume that $t \leq p$.)
22. (a) Show by a combinatorial argument that
$$S(n,k) = kS(n-1,k) + S(n-1,k-1).$$
    (b) Use the result in part (a) to describe how to compute Stirling numbers of the second kind by a method similar to Pascal's triangle.
    (c) Apply your result in part (b) to compute $S(6,3)$.
23. Show by a combinatorial argument that
$$S(n+1,k) = C(n,0)S(0,k-1) + C(n,1)S(1,k-1) + \cdots + C(n,n)S(n,k-1).$$
24. (a) If order counts in a partition, then $\{3,2\}$ is different from $\{2,3\}$. Find the number of partitions of 5 if order matters.
    (b) Find the number of partitions of 5 into exactly 2 parts where order matters.
    (c) Show that the number of partitions of n into exactly k parts where order matters is given by $C(n-1,k-1)$.
25. The *Bell number* B_n is the number of partitions of a set of n elements into nonempty, indistinguishable cells. Note that
$$B_n = S(n,0) + S(n,1) + \cdots + S(n,n).$$
Show that
$$B_n = \binom{n-1}{0}B_0 + \binom{n-1}{1}B_1 + \cdots + \binom{n-1}{n-1}B_{n-1}.$$

2.11 MULTINOMIAL COEFFICIENTS

2.11.1 Occupancy Problems with a Specified Distribution

In this section we consider the occupancy problem of distributing n distinguishable balls into k distinguishable cells. In particular, we consider the situation where we distribute n_1 balls into the first cell, n_2 into the second cell, ..., n_k into the kth cell. Let
$$C(n; n_1, n_2, \ldots, n_k)$$
denote the number of ways this can be done. This section is devoted to the study of the number $C(n; n_1, n_2, \ldots, n_k)$, which is sometimes also written as
$$\binom{n}{n_1, n_2, \ldots, n_k}$$
and called the *multinomial coefficient.*

Example 2.31 Campus Registration The university registrar's office is having a problem. It has 11 new students to squeeze into 4 sections of an introductory course: 3 in the first, 4 each in the second and third, and 0 in the fourth (that section is already full). In how many ways can this be done? The answer is $C(11; 3, 4, 4, 0)$. Now there are $C(11, 3)$ choices for the first section; for each of these there are $C(8, 4)$ choices for the second section; for each of these there are $C(4, 4)$ choices for the third section; for each of these there are $C(0, 0)$ choices for the fourth section. Hence, by the product rule, the number of ways to assign sections is

$$\begin{aligned} C(11; 3, 4, 4, 0) &= C(11, 3) \times C(8, 4) \times C(4, 4) \times C(0, 0) \\ &= \frac{11!}{3!8!} \times \frac{8!}{4!4!} \times \frac{4!}{4!0!} \times \frac{0!}{0!0!} = \frac{11!}{3!4!4!}, \end{aligned}$$

since $0! = 1$. Of course, $C(0, 0)$ always equals 1, so the answer is equivalent to $C(11; 3, 4, 4)$. Additionally, $C(4, 4) = 1$, so the answer is also equivalent to $C(11, 3) \times C(8, 4)$. The reason for this is that once the 3 students for the first section and 4 students for the second section have been chosen, there is only one way to choose the remaining 4 for the third section.

Note that if section assignments for 11 students are made at random, there are 4^{11} possible assignments: For each student, there are 4 choices of section. Hence, the probability of having 3 students in the first section, 4 each in the second and third sections, and 0 in the fourth is given by

$$\frac{C(11; 3, 4, 4, 0)}{4^{11}}.$$

In general, suppose that $\Pr(n; n_1, n_2, \ldots, n_k)$ denotes the probability that if n balls are distributed at random into k cells, there will be n_i balls in cell i, $i = 1, 2, \ldots, k$. Then

$$\Pr(n; n_1, n_2, \ldots, n_k) = \frac{C(n; n_1, n_2, \ldots, n_k)}{k^n}.$$

(Why?) Note that when calculating the multinomial coefficient, the acknowledgment of empty cells does not affect the calculation. This is because

$$C(n; n_1, n_2, \ldots, n_j, 0, 0, \ldots, 0) = C(n; n_1, n_2, \ldots, n_j).$$

However, the probability of a multinomial distribution *is* affected by empty cells as the denominator is based on the number of cells, both empty and nonempty.

Continuing with our example, suppose that suddenly, spaces in the fourth section become available. The registrar's office now wishes to put 3 people each into the first, second, and third sections, and 2 into the fourth. In how many ways can this be done? Of the 11 students, 3 must be chosen for the first section; of the remaining 8 students, 3 must be chosen for the second section; of the remaining 5 students, 3 must be chosen for the third section; finally, the remaining 2 must be put into the fourth section. The total number of ways of making the assignments is

$$\begin{aligned} C(11; 3, 3, 3, 2) &= C(11, 3) \times C(8, 3) \times C(5, 3) \times C(2, 2) \\ &= \frac{11!}{3!8!} \times \frac{8!}{3!5!} \times \frac{5!}{3!2!} \times \frac{2!}{2!0!} = \frac{11!}{3!3!3!2!}. \end{aligned}$$ ■

Let us derive a formula for $C(n; n_1, n_2, \ldots, n_k)$. By reasoning analogous to that used in Example 2.31,

$$\begin{aligned}
C(n; n_1, n_2, \ldots, n_k) &= C(n, n_1) \times C(n-n_1, n_2) \times C(n-n_1-n_2, n_3) \times \cdots \\
&\quad \times C(n-n_1-n_2-\cdots-n_{k-1}, n_k) \\
&= \frac{n!}{n_1!(n-n_1)!} \times \frac{(n-n_1)!}{n_2!(n-n_1-n_2)!} \times \frac{(n-n_1-n_2)!}{n_3!(n-n_1-n_2-n_3)!} \\
&\quad \times \cdots \times \frac{(n-n_1-n_2-\cdots-n_{k-1})!}{n_k!(n-n_1-n_2-\cdots-n_k)!} \\
&= \frac{n!}{n_1!n_2!\cdots n_k!(n-n_1-n_2-\cdots-n_k)!}.
\end{aligned}$$

Since $n_1 + n_2 + \cdots + n_k = n$, and since $0! = 1$, we have the following result.

Theorem 2.5

$$C(n; n_1, n_2, \ldots, n_k) = \frac{n!}{n_1!n_2!\cdots n_k!}.$$

We now give several illustrations and applications of this theorem.

1. The number of 4-digit bit strings consisting of three 1's and one 0 is $C(4; 3, 1)$: Out of four places, we choose three for the digit 1 and one for the digit 0. Hence, the number of such strings is given by

$$C(4; 3, 1) = \frac{4!}{3!1!} = 4.$$

   The four such strings are 1110, 1101, 1011, and 0111.

2. The number of 5-digit numbers consisting of two 2's, two 3's, and one 1 is

$$C(5; 2, 2, 1) = \frac{5!}{2!2!1!} = 30.$$

3. Notice that $C(n; n_1, n_2) = C(n, n_1)$. Why should this be true?

4. An NHL hockey season consists of 82 games. The number of ways the season can end in 41 wins, 27 losses, and 14 overtime losses is

$$C(82; 41, 27, 14) = \frac{82!}{41!27!14!}.$$

5. RNA is a messenger molecule whose links are defined from DNA. An RNA chain has, at each link, one of four bases. The possible bases are the same as those in DNA (Example 2.2), except that the base uracil (U) replaces the base thymine (T). How many possible RNA chains of length 6 are there consisting of 3 cytosines (C) and 3 adenines (A)? To answer this question, we think of

6 positions in the chain, and of dividing these positions into four sets, 3 into the C set, 3 into the A set, and 0 into the T and G sets. The number of ways this can be done is given by

$$C(6;3,3,0,0) = C(6;3,3) = 20.$$

6. There are 4^6 possible RNA chains of length 6. The probability of obtaining one with 3 C's and 3 A's if the RNA chain were produced at random (i.e., all possibilities equally likely) would be

$$\Pr(6;3,3,0,0) = \frac{C(6;3,3,0,0)}{4^6} = \frac{C(6;3,3)}{4^6} = \frac{20}{4096} = \frac{5}{1024} \approx .005.$$

Note that this is not $\Pr(6;3,3)$.

7. The number of 10-link RNA chains consisting of 3 A's, 2 C's, 2 U's, and 3 G's is

$$C(10;3,2,2,3) = 25{,}200.$$

8. The number of RNA chains as described in the previous example which end in AAG is

$$C(7;1,2,2,2) = 630,$$

since there are now only the first 7 positions to be filled, and two of the A's and one of the G's are already used up. Notice how knowing the end of a chain can reduce dramatically the number of possible chains. In the next section we see how, by judicious use of various enzymes which decompose RNA chains, we might further limit the number of possible chains until, by a certain amount of detective work, we can uncover the original RNA chain without actually observing it.

2.11.2 Permutations with Classes of Indistinguishable Objects

Applications 1, 2, and 5–8 suggest the following general notion: Suppose that there are n objects, n_1 of type 1, n_2 of type 2, ..., n_k of type k, with $n_1 + n_2 + \cdots + n_k = n$. Suppose that objects of the same type are indistinguishable. The number of distinguishable permutations of these objects is denoted $P(n; n_1, n_2, \ldots, n_k)$. We use the word "distinguishable" here because we assume that objects of the same type are indistinguishable. For instance, suppose that $n = 3$ and there are two type 1 objects, a and a, and one type 2 object, b. Then there are $3! = 6$ permutations of the three objects, but several of these are indistinguishable. For example, baa in which the first of the two a's comes second is indistinguishable from baa in which the second of the two a's comes second. There are only three distinguishable permutations, baa, aba, and aab.

Theorem 2.6

$$P(n; n_1, n_2, \ldots, n_k) = C(n; n_1, n_2, \ldots, n_k).$$

Proof. We have n positions or places to fill in the permutation, and we assign n_1 of these to type 1 objects, n_2 to type 2 objects, and so on. Q.E.D.

We return to permutations with classes of indistinguishable objects in Section 2.13.

EXERCISES FOR SECTION 2.11

1. Compute:

   (a) $C(7; 2, 2, 2, 1)$ (b) $C(9; 3, 3, 3)$ (c) $C(8; 1, 2, 2, 2, 1)$
   (d) $\Pr(6; 2, 2, 2)$ (e) $\Pr(10; 2, 1, 1, 2, 4)$ (f) $\Pr(8; 4, 2, 2, 0, 0)$
   (g) $P(9; 6, 1, 2)$ (h) $P(7; 3, 1, 3)$ (i) $P(3; 1, 1, 1)$

2. Find $C(n; 1, 1, 1, \ldots, 1)$.
3. Find:

   (a) $P(n; 1, n-1)$ (b) $\Pr(n; 1, n-1)$ (c) $\Pr(n; 1, n-1, 0)$

4. A code is being written using the five symbols $+$, $\natural$, $\bowtie$, ∇, and $\otimes$.
   (a) How many 10-digit codewords are there that use exactly 2 of each symbol?
   (b) If an 10-digit codeword is chosen at random, what is the probability that it will use exactly 2 of each symbol?
5. In a kennel that is short of space, 12 dogs must be put into 3 cages, 4 in cage 1, 5 in cage 2, and 3 in cage 3. In how many ways can this be done?
6. A code is being written using three symbols, a, b, and c.
   (a) How many 7-digit codewords can be written using exactly 4 a's, 1 b, and 2 c's?
   (b) If a 7-digit codeword is chosen at random, what is the probability that it will use exactly 4 a's, 1 b, and 2 c's?
7. A code is being written using the five digits 1, 2, 3, 4, and 5.
   (a) How many 15-digit codewords are there that use exactly 3 of each digit?
   (b) If a 15-digit codeword is chosen at random, what is the probability that it will use exactly 3 of each digit?
8. How many RNA chains have the same makeup of bases as the chain

   UGCCAUCCGAC?

9. (a) How many different "words" can be formed using all the letters of the word *excellent*?
   (b) If a 9-letter "word" is chosen at random, what is the probability that it will use all the letters of the word *excellent*?

10. How many ways are there to form a sequence of 10 letters from 4 a's, 4 b's, 4 c's, and 4 d's if each letter must appear at least twice?

11. How many distinguishable permutations are there of the symbols $a, a, a, a, a, b, c, d, e$ if no two a's are adjacent?

12. Repeat Exercise 20 of Section 2.10 under the assumption that the particles are distinguishable (the Maxwell-Boltzmann case).

13. (a) Suppose that we distinguish 5 different light particles hitting the retina. In how many ways could these be distributed among three cells, with three hitting the first cell and one hitting each of the other cells?

    (b) If we know there are 5 different light particles distributed among the three cells, what is the probability that they will be distributed as in part (a)?

14. Suppose that 35 radioactive particles hit a Geiger counter with 50 counters. In how many different ways can this happen with all but the 35th particle hitting the first counter?

15. Suppose that 6 people are invited for job interviews.

    (a) How many different ways are there for 2 of them to be interviewed on Monday, 2 on Wednesday, and 2 on Saturday?

    (b) Given the 6 interviews, what is the probability that the interviews will be distributed as in part (a) if the 6 people are assigned to days at random?

    (c) How many ways are there for the interviews to be distributed into 3 days, 2 per day?

16. Suppose that we have 4 elements. How many distinguishable ways are there to assign these to 4 distinguishable sets, 1 to each set, if the elements are:

    (a) a, b, c, d? (b) a, b, b, b? (c) a, a, b, b? (d) a, b, b, c?

2.12 COMPLETE DIGEST BY ENZYMES[20]

Let us consider the problem of discovering what a given RNA chain looks like without actually observing the chain itself (RNA chains were introduced in Section 2.11). Some enzymes break up an RNA chain into fragments after each G link. Others break up the chain after each C or U link. For example, suppose that we have the chain

CCGGUCCGAAAG.

Applying the G enzyme breaks the chain into the following fragments:

G fragments: CCG, G, UCCG, AAAG.

[20] This section may be omitted without loss of continuity. The material here is not needed again until Section 11.4.4. However, this section includes a detailed discussion of an applied topic, and we always include it in our courses.

We then know that these are the fragments, but we do not know in what order they appear. How many possible chains have these four fragments? The answer is that $4! = 24$ chains do: There is one chain corresponding to each of the different permutations of the fragments. One such chain (different from the original) is the chain

UCCGGCCGAAAG.

Suppose that we next apply the U, C enzyme, the enzyme that breaks up a chain after each C link or U link. We obtain the following fragments:

U, C fragments: C, C, GGU, C, C, GAAAG.

Again, we know that these are the fragments, but we do not know in what order they appear. How many chains are there with these fragments? One is tempted to say that there are 6! chains, but that is not right. For example, if the fragments were

C, C, C, C, C, C,

there would not be 6! chains with these fragments, but only one, the chain

CCCCCC.

The point is that some of the fragments are indistinguishable. To count the number of distinguishable chains with the given fragments, we note that there are six fragments. Four of these are C fragments, one is GGU, and one is GAAAG. Thus, by Theorem 2.6, the number of possible chains with these as fragments is

$$P(6;4,1,1) = C(6;4,1,1) = \frac{6!}{4!1!1!} = 30.$$

Actually, this computation is still a little off. Notice that the fragment GAAAG among the U, C fragments could not have appeared except as the terminal fragment because it does not end in U or C. Hence, we know that the chain ends

GAAAG.

There are five remaining U, C fragments: C, C, C, C, and GGU. The number of chains (beginning segments of chains) with these as fragments is

$$C(5;4,1) = 5.$$

The possible chains are obtained by adding GAAAG to one of these 5 beginning chains. The possibilities are

CCCCGGUGAAAG
CCCGGUCGAAAG
CCGGUCCGAAAG
CGGUCCCGAAAG
GGUCCCCGAAAG.

We have not yet combined our knowledge of both G and U, C fragments. Can we learn anything about the original chain by using our knowledge of both? Which of the 5 chains that we have listed has the proper G fragments? The first does not, for it would have a G fragment CCCCG, which does not appear among the fragments when the G enzyme is applied. A similar analysis shows that only the third chain,

CCGGUCCGAAAG,

has the proper set of G fragments. Hence, we have recovered the initial chain from among those that have the given U, C fragments.

This is an example of recovery of an RNA chain given a *complete enzyme digest*, that is, a split up after every G link and another after every U or C link. It is remarkable that we have been able to limit the large number of possible chains for any one set of fragments to only one possible chain by considering both sets of fragments. This result is more remarkable still if we consider trying to guess the chain knowing just its bases but not their order. Then we have

$$C(12; 4, 4, 3, 1) = 138{,}600 \text{ possible chains!}$$

Let us give another example. Suppose we are told that an RNA chain gives rise to the following fragments after complete digest by the G enzyme and the U, C enzyme:

G fragments: UG, ACG, AC
U, C fragments: U, GAC, GAC.

Can we discover the original chain? To begin with, we ask again whether or not the U, C fragments tell us which part of the chain must come last. The answer is that, in this case, they do not. However, the G fragments do: AC could only have arisen as a G fragment if it came last. Hence, the two remaining G fragments can be arranged in any order, and the possible chains with the given G fragments are

UGACGAC and ACGUGAC.

Now the latter chain would give rise to AC as one of the U, C fragments. Hence, the former must be the correct chain.

It is not always possible to recover the original RNA chain completely knowing the G fragments and U, C fragments. Sometimes the complete digest by these two enzymes is ambiguous in the sense that there are two RNA chains with the same set of G fragments and the same set of U, C fragments. We ask the reader to show this as an exercise (Exercise 8).

The "fragmentation stratagem" described in this section was used by R. W. Holley and his co-workers at Cornell University (Holley, *et al.* [1965]) to determine the first nucleic acid sequence. The method is not used anymore and indeed was used only for a short time before other, more efficient, methods were adopted. However, it has great historical significance and illustrates an important role for mathematical methods in biology. Nowadays, by the use of radioactive marking and high-speed computer analysis, it is possible to sequence long RNA chains rather quickly.

The reader who is interested in more details about complete digests by enzymes should read Hutchinson [1969], Mosimann [1968], or Mosimann, *et al.* [1966]. We return to this problem in Section 11.4.4.

EXERCISES FOR SECTION 2.12

1. An RNA chain has the following fragments after being subjected to complete digest by G and U, C enzymes:

   G fragments: CUG, CAAG, G, UC
   U, C fragments: C, C, U, AAGC, GGU.

   (a) How many RNA chains are there with these G fragments?

   (b) How many RNA chains are there with these U, C fragments?

   (c) Find *all* RNA chains that have these G and U, C fragments.

2. In Exercise 1, find the number of RNA chains with the same bases as those of the chains with the given G fragments.
3. Repeat Exercise 1 for the following G and U, C fragments:

   G fragments: G, UCG, G, G, UU
   U, C fragments: GGGU, U, GU, C.

4. In Exercise 3, find the number of RNA chains with the same bases as those of the chains with the given G fragments.
5. Repeat Exercise 1 for the following G and U, C fragments:

   G fragments: G, G, CC, CUG, G
   U, C fragments: GGGC, U, C, GC.

6. In Exercise 5, find the number of RNA chains with the same bases as those of the chains with the given G fragments.
7. A bit string is broken up after every 1 and after every 0. The resulting pieces (not necessarily in proper order) are as follows:

   break up after 1: 0, 001, 01, 01
   break up after 0: 0, 10, 0, 10, 10.

   (a) How many bit strings are there which have these pieces after breakup following each 1?

   (b) After each 0?

   (c) Find all bit strings having both of these sets of pieces.

8. Find an RNA chain which is ambiguous in the sense that there is another chain with the same G fragments and the same U, C fragments. (Can you find one with six or fewer links?)
9. What is the shortest possible RNA chain that is ambiguous in the sense of Exercise 8?
10. Can a bit string be ambiguous if it is broken up as in Exercise 7? Why?

2.13 PERMUTATIONS WITH CLASSES OF INDISTINGUISHABLE OBJECTS REVISITED

In Sections 2.11 and 2.12 we encountered the problem of counting the number of permutations of a set of objects in which some of the objects were indistinguishable. In this section we develop an alternative procedure for counting in this situation.

Example 2.32 "Hot Hand" A basketball player has observed that of his 10 shots attempted in an earlier game, 4 were made and 6 were missed. However, all 4 made shots came first. The basketball player's observation can be abbreviated as

XXXXOOOOOO,

where X stands for a made shot and O for a missed one. Is this observation a coincidence, or does it suggest that the player had a "hot hand"? A hot hand assumes that once a player makes a shot, he or she has a higher-than-average chance of making the next shot. (For a detailed analysis of the hot hand phenomenon, see Tversky and Gilovich [1989].) To answer this question, let us assume that there is no such thing as having a hot hand, that is, that a shot is no more (or less) likely to go in when it follows a made (or missed) shot. Thus, let us assume that the made shots occur at random, and each shot has the same probability of being made, independent of what happens to the other shots. It follows that all possible orderings of 4 made shots and 6 missed shots are equally likely.[21] How many such orderings are there? The answer, to use the notation of Section 2.11, is

$$P(10; 6, 4) = C(10; 6, 4) = \frac{10!}{4!6!} = 210.$$

To derive this directly, note that there are 10 positions and we wish to assign 4 of these to X and 6 to O. Thus, the number of such orderings is $C(10; 6, 4) = 210$. If all such orderings are equally likely, the probability of seeing the specific arrangement

XXXXOOOOOO

is 1 out of 210. Of course, this is the probability of seeing any one given arrangement. What is more interesting to calculate is the probability of seeing 4 made shots together out of 10 shots in which exactly 4 are made. In how many arrangements of 10 shots, 4 made and 6 missed, do the 4 made ones occur together? To answer this, let us consider the 4 made shots as one unit X*. Then we wish to consider the number of orders of 1 X* and 6 O's. There are

$$C(7; 1, 6) = \frac{7!}{1!6!} = 7$$

[21] It does not follow that all orderings of 10 made and missed shots are equally likely. For instance, even if making a shot occurs at random, if making a shot is very unlikely, then the sequence OOOOOOOOOO is much more likely than the sequence XXXXXXXXXX.

such orders. These correspond to the orders

$$
\begin{array}{ll}
\text{XXXXOOOOOO} & (\text{which is X}^*\text{OOOOOO}) \\
\text{OXXXXOOOOO} & (\text{which is OX}^*\text{OOOOO}) \\
\text{OOXXXXOOOO} & (\text{which is OOX}^*\text{OOOO}) \\
\text{OOOXXXXOOO} & (\text{which is OOOX}^*\text{OOO}) \\
\text{OOOOXXXXOO} & (\text{which is OOOOX}^*\text{OO}) \\
\text{OOOOOXXXXO} & (\text{which is OOOOOX}^*\text{O}) \\
\text{OOOOOOXXXX} & (\text{which is OOOOOOX}^*).
\end{array}
$$

The probability of seeing 4 made shots and 6 missed shots with all the made shots together, given that there are 4 made shots and 6 missed ones, is therefore 7 out of 210, or $\frac{1}{30}$. This is quite small. Hence, since seeing all the made shots together is unlikely, we expect that perhaps this is not a random occurrence, and there is such a thing as a hot hand. (We return to the question of hot hand in Example 6.8.)

Before leaving this example, it is convenient to repeat the calculation of the number of ways of ordering 4 made shots and 6 missed ones. Suppose we label the shots so that they are distinguishable:

$$\text{X}_a, \text{X}_b, \text{X}_c, \text{X}_d, \text{O}_a, \text{O}_b, \text{O}_c, \text{O}_d, \text{O}_e, \text{O}_f.$$

There are 10! permutations of these 10 labels. For each such permutation, we can reorder the 4 X's arbitrarily; there are 4! such reorderings. Each reordering gives rise to an ordering which is considered the same as far as we are concerned. Similarly, we can reorder the 6 O's in 6! ways. Thus, groups of $4! \times 6!$ orderings are the same, and each permutation corresponds to $4! \times 6!$ similar ones. The number of indistinguishable permutations is

$$\frac{10!}{4!6!}. \qquad \blacksquare$$

The reasoning in Example 2.32 generalizes to give us another proof of the result (Theorem 2.6) that $P(n; n_1, n_2, \ldots, n_k) = C(n; n_1, n_2, \ldots, n_k)$.

EXERCISES FOR SECTION 2.13

1. Suppose a researcher observes that of 12 petri dishes, 4 have growths and 8 do not. The 4 dishes with growths are next to each other. Assuming that 4 of the 12 petri dishes have growths and that all orderings of these dishes are equally likely, what is the probability that the 4 dishes with growths will all be next to each other?
2. A market researcher observes that of 11 cars on a block, 4 are foreign and 7 are domestic. The 4 foreign cars are next to each other. Assuming that 4 of the 11 cars are foreign and that all orderings of these 11 cars are equally likely, what is the probability that the 4 foreign cars are next to each other?
3. Suppose a forester observes that some trees are sick (S) and some well (W), and that of 13 trees in a row, the first 5 are S and the last 8 are W.

(a) Assuming that sickness occurs at random and all trees have the same probability of being sick, independent of what happens to the other trees, what is the probability of observing the given order?

(b) What is the probability that out of 13 trees, 5 sick and 8 well, the 5 sick ones occur together?

4. Suppose a forester observes that some trees are sick (S), some well (W), and some questionable (Q). Assuming that of 30 trees, 10 are sick or questionable, what is the probability that these 10 appear consecutively? Assume that all sequences of sick, well, and questionable with 10 sick or questionable are equally likely.

5. Suppose that of 11 houses lined up in one block, 6 are infested with termites.

   (a) In how many ways can the presence or absence of termites occur so that these 6 houses are next to each other?

   (b) In how many ways can this occur so that none of these 6 houses are next to each other?

   (c) In how many ways can we schedule an order of visits that go to two of these houses in which there are no termites?

   (d) In how many ways can we schedule an order of visits that go to two of these houses if at most one house that we visit can have termites?

6. In how many different orders can a couple produce 9 children with 5 boys so that a boy comes first and all 4 girls are born consecutively?

7. How many distinct ways are there to seat 8 people around a circular table? (Clarify what "distinct" means here.)

8. If an RNA chain of length 4 is chosen at random, what is the probability that it has:

   (a) At least three consecutive C's? (b) At least two consecutive C's?

   (c) A consecutive AG? (d) A consecutive AUC?

9. How many bit strings of length 21 have every 1 followed by 0 and have seventeen 0's and four 1's?

10. How many RNA chains of length 20 have five A's, four U's, five C's, and six G's and have every C followed by G?

11. There are 20 people whose records are stored in order in a file. We want to choose 4 of these at random in performing a survey, making sure not to choose two consecutive people. In how many ways can this be done? (*Hint:* Either we choose the last person or we do not.)

2.14 THE BINOMIAL EXPANSION

Suppose that 6-position license plates are being made with each of the six positions being either a number or a letter. Referring back to the sum and product rules, there are

$$(10 + 26)^6 = 2{,}176{,}782{,}336$$

possible license plates. Generalizing, suppose that the license plates need to be n digits with each position being either one of a things or one of b things. Then there are

$$(a+b)^n$$

license plates. As an application of the ideas considered in this chapter, let us develop a useful formula for $(a+b)^n$.

Theorem 2.7 (Binomial Expansion) For $n \geq 0$,

$$(a+b)^n = \sum_{k=0}^{n} C(n,k) a^k b^{n-k} = \sum_{k=0}^{n} \binom{n}{k} a^k b^{n-k}.$$

Proof.[22] Note that

$$(a+b)^n = \underbrace{(a+b)(a+b)\cdots(a+b)}_{n \text{ times}}.$$

In multiplying out, we pick one term from each factor $(a+b)$. Hence, we only obtain terms of the form $a^k b^{n-k}$. To find the coefficient of $a^k b^{n-k}$, note that to obtain $a^k b^{n-k}$, we need to choose k of the terms from which to choose a. This can be done in $\binom{n}{k}$ ways. Q.E.D.

In particular, we have

$$\begin{aligned}
(a+b)^2 &= \binom{2}{0}a^2 + \binom{2}{1}ab + \binom{2}{2}b^2 = a^2 + 2ab + b^2 \\
(a+b)^3 &= \binom{3}{0}a^3 + \binom{3}{1}a^2b + \binom{3}{2}ab^2 + \binom{3}{3}b^3 = a^3 + 3a^2b + 3ab^2 + b^3 \\
(a+b)^4 &= \binom{4}{0}a^4 + \binom{4}{1}a^3b + \binom{4}{2}a^2b^2 + \binom{4}{3}ab^3 + \binom{4}{4}b^4 \\
&= a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4.
\end{aligned}$$

The reader might wish to compare the coefficients here with those numbers appearing in Pascal's triangle (Figure 2.4). Do you notice any similarities?

It is not hard to generalize the binomial expansion of Theorem 2.7 to an expansion of

$$(a+b+c)^n$$

and more generally of

$$(a_1 + a_2 + \cdots + a_k)^n.$$

We leave the generalizations to the reader (Exercises 5 and 8).

Let us give a few applications of the binomial expansion here. The coefficient of x^{20} in the expansion of $(1+x)^{30}$ is obtained by taking $a = 1$ and $b = x$ in Theorem 2.7. We are seeking the coefficient of $1^{10}x^{20}$, that is, $C(30,10)$.

[22] For an alternative proof, see Exercise 12.

Theorem 2.8 For $n \geq 0$,

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = 2^n.$$

Proof. Note that

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

Hence, by the binomial expansion with $a = b = 1$,

$$2^n = \sum_{k=0}^{n} \binom{n}{k} 1^k 1^{n-k}.$$

Q.E.D.

Another way of looking at Theorem 2.8 is the following. The number 2^n counts the number of subsets of an n-set. Also, the left-hand side of Theorem 2.8 counts the number of 0-element subsets of an n-set plus the number of 1-element subsets plus ... plus the number of n-element subsets. Each subset is counted once and only once in this way.

Theorem 2.9 For $n > 0$,

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

Proof.

$$0 = (1-1)^n = (-1+1)^n = \sum_{k=0}^{n} \binom{n}{k} (-1)^k (1)^{n-k}.$$

Q.E.D.

Corollary 2.9.1 For $n > 0$,

$$\binom{n}{0} + \binom{n}{2} + \cdots = \binom{n}{1} + \binom{n}{3} + \cdots.$$

The interpretation of this corollary is that the number of ways to select an even number of objects from n equals the number of ways to select an odd number.

EXERCISES FOR SECTION 2.14

1. Write out:

   (a) $(x+y)^5$ (b) $(a+2b)^3$ (c) $(2u+3v)^4$

2. Find the coefficient of x^{11} in the expansion of:

   (a) $(1+x)^{16}$ (b) $(2+x)^{14}$ (c) $(2x+4y)^{11}$

3. What is the coefficient of x^9 in the expansion of $(1+x)^{12}(1+x)^4$?
4. What is the coefficient of x^8 in the expansion of $(1+x)^{10}(1+x)^6$?

5. Find a formula for $(a+b+c)^n$.
6. Find the coefficient of $a^2b^2c^2$ in the expansion of $(a+b+c)^6$.
7. Find the coefficient of xyz^3 in the expansion of $(x+y+z)^5$.
8. Find a formula for $(a_1+a_2+\cdots+a_k)^n$.
9. What is the coefficient of a^3bc^2 in the expansion of $(a+b+c)^6$?
10. What is the coefficient of xy^2z^2w in the expansion of $(x+y+z+2w)^6$?
11. What is the coefficient of $a^3b^2cd^6$ in the expansion of $(a+5b+2c+2d)^{12}$?
12. Prove Theorem 2.7 by induction on n.
13. Find $\binom{12}{0}+\binom{12}{2}+\binom{12}{4}+\binom{12}{6}+\binom{12}{8}+\binom{12}{10}+\binom{12}{12}$.
14. Prove that the number of even-sized subsets of an n-set equals 2^{n-1}.
15. A bit string has *even parity* if it has an even number of 1's. How many bit strings of length n have even parity?
16. Find:
    (a) $\sum_{k=0}^{n} 2^k\binom{n}{k}$ (b) $\sum_{k=0}^{n} 4^k\binom{n}{k}$
    (c) $\sum_{k=0}^{n} x^k\binom{n}{k}$ (d) $\sum_{k=2}^{n} k(k-1)\binom{n}{k}$
    (e) $\sum_{k=1}^{n} k\binom{n}{k}$ [*Hint:* Differentiate the expansion of $(x+1)^n$ and set $x=1$.]
17. Show that:
    (a) $\sum_{k=1}^{n} k\binom{n}{k}2^{k-1}2^{n-k} = n(4^{n-1})$ (b) $\sum_{k=1}^{n} k\binom{n}{k}2^{n-k} = n(3^{n-1})$

2.15 POWER IN SIMPLE GAMES[23]

2.15.1 Examples of Simple Games

In this section we apply to the analysis of multiperson games some of the counting rules described previously. Now in modern applied mathematics, a game has come to mean more than just Monopoly, chess, or poker. It is any situation where a group of players is competing for different rewards or payoffs. In this sense, politics is a game, the economic marketplace is a game, the international bargaining arena is a game, and so on. We shall take this broad view of games here.

Let us think of a game as having a set I of n players. We are interested in possible cooperation among the players and, accordingly, we study *coalitions* of players, which correspond to subsets of the set I. We concentrate on *simple games*,

[23]This section may be omitted without loss of continuity. The formal prerequisites for this section are Sections 2.1–2.8.

games in which each coalition is either winning or losing. We can define a simple game by giving a value function v which assigns the number 0 or 1 to each coalition $S \subseteq I$, with $v(S)$ equal to 0 if S is a losing coalition and 1 if S is a winning coalition. It is usually assumed in game theory that a subset of a losing coalition cannot be winning, and we shall make that assumption. It is also usually assumed that for all S, either S or $I - S$ is losing. We shall assume that.

Very important examples of simple games are the weighted majority games. In a *weighted majority game*, there are n players, player i has v_i votes, and a coalition is winning if and only if it has at least q votes. We denote this game by

$$[q; v_1, v_2, \ldots, v_n].$$

The assumption that either S or $I - S$ loses places restrictions on the allowable $q, v_1, v_2, \ldots, v_n$. For instance, $[3; 4, 4]$ does not satisfy this requirement. Weighted majority games arise in corporations, where the players are the stockholders and a stockholder has one vote for each share owned. Most legislatures are weighted majority games of the form $[q; 1, 1, \ldots, 1]$, where each player has one vote. However, some legislatures give a legislator a number of votes corresponding to the population of his or her district. For example, in 1964 the Nassau County, New York, Board of Supervisors was the weighted majority game $[59; 31, 31, 21, 28, 2, 2]$ (Banzhaf [1965]).

Another example is the Council of the European Union. This body, made up of 27 member states, legislates for the Union. In most of its cases, the Council decides by a "qualified majority vote" from its member states carrying the following weights:

Member Countries	Votes
Germany, France, Italy, and the United Kingdom	29
Spain and Poland	27
Romania	14
The Netherlands	13
Belgium, Czech Republic, Greece, Hungary, and Portugal	12
Austria, Bulgaria, and Sweden	10
Denmark, Ireland, Lithuania, Slovakia, and Finland	7
Cyprus, Estonia, Latvia, Luxembourg, and Slovenia	4
Malta	3
Total	345

At least 255 votes are required for a qualified majority. [In addition, a majority of member states (and in some cases a two-thirds majority) must approve for a qualified majority to be reached. We will not consider this criterion.] Therefore, this weighted majority game is

$$[255; 29, 29, 29, 29, 27, 27, 14, 13, 12, 12, 12, 12, 12, 10, 10, 10, 7, 7, 7, 7, 7, 4, 4, 4, 4, 4, 3].$$

Perhaps the most elementary weighted majority game is the game $[2; 1, 1, 1]$. In this game there are three players, each having one vote, and a simple majority of the players forms a winning coalition. Thus, the winning coalitions are the sets

$$\{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}.$$

Another example of a simple game is the U.N. Security Council. Here there are 15 players: 5 permanent members (China, France, Russia, the United Kingdom, the United States) and 10 nonpermanent members. For decisions on substantive matters, a coalition is winning if and only if it has all 5 permanent members, since they have veto power,[24] and at least 4 of the 10 nonpermanent members. (Decisions on procedural matters are made by an affirmative vote of at least nine of the 15 members.) It is interesting to note that for substantive decisions the Security Council can be looked at as a weighted majority game. Consider the game

$$[39; 7, 7, 7, 7, 7, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],$$

where the first five players correspond to the permanent members. The winning coalitions in this game are exactly the same as those in the Security Council, as is easy to check. Hence, even though weighted votes are not explicitly assigned in the Security Council, it can be considered a weighted majority game. (The reader might wish to think about how to obtain numbers, such as 39, 7, and 1, which translate a simple game into a weighted majority game.)

A similar situation arises for the Australian government. In making national decisions, 6 states and the federal government play a role. In effect, a measure passes if and only if it has the support of at least 5 states or at least 2 states and the federal government. As is easy to see, this simple game corresponds (in the sense of having the same winning coalitions) to the game $[5; 1, 1, 1, 1, 1, 1, 3]$, where the seventh player is the federal government.

Not every simple game is a weighted majority game. A bicameral legislature is an example (see Exercise 5).

2.15.2 The Shapley-Shubik Power Index

We shall be concerned with measuring the *power* of a player in a simple game: his or her ability to maneuver into a winning coalition. Note first that power is not necessarily proportional to the number of votes a player has. For example, compare the two games $[2; 1, 1, 1]$ and $[51; 49, 48, 3]$. In each game, there are 3 players, and any coalition of 2 or more players wins. Thus, in both games, player 3 is in the same winning coalitions, and hence has essentially the same power. These two games might be interpreted as a legislature with 3 parties. In the first legislature, there are 3 equal parties and 2 out of 3 must go along for a measure to pass. In the second legislature, there are 2 large parties and a small party. However, assuming that party members vote as a bloc, it is still necessary to get 2 out of 3 parties to go along to pass a measure, so in effect the third small party has as much power as it does in the first legislature.

There have been a number of alternative approaches to the measurement of power in simple games. These include power indices proposed by Banzhaf [1965], Coleman [1971], Deegan and Packel [1978], Johnston [1978], and Shapley and Shubik [1954]. We shall refer to the Banzhaf and Coleman power indices in Section 5.7.

---

[24] This is the rule of "great power unanimity."

The former has come to be used in the courts in "one-person, one-vote" cases. Here we concentrate on the Shapley-Shubik power index, introduced in its original form by Shapley [1953] and in the form we present by Shapley and Shubik [1954]. (In its more general original form, it is called the *Shapley value*.) For a survey of the literature on the Shapley-Shubik index, see Shapley [1981]. For a survey of the literature on power measures, see Lucas [1983].

Let us think of building up a coalition by adding one player at a time until we reach a winning coalition. The player whose addition throws the coalition over from losing to winning is called *pivotal*. More formally, let us consider any permutation of the players and call a player i *pivotal* for that permutation if the set of players preceding i is losing, but the set of players up to and including player i is winning. For example, in the game $[2; 1, 1, 1]$, player 2 is pivotal in the permutation 1, 2, 3 and in the permutation 3, 2, 1. The *Shapley-Shubik power index* p_i for player i in a simple game is defined as follows:

$$p_i = \frac{\text{number of permutations of the players in which } i \text{ is pivotal}}{\text{number of permutations of the players}}.$$

If we think of one permutation of the players being chosen at random, the Shapley-Shubik power index for player i is the probability that player i is pivotal. In the game $[2; 1, 1, 1]$, for example, there are three players and hence 3! permutations. Each player is pivotal in two of these. For example, as we have noted, player 2 is pivotal in 1, 2, 3 and 3, 2, 1. Hence, each player has power $2/3! = 1/3$. In the game $[51; 49, 48, 3]$, player 1 is pivotal in the permutations 2, 1, 3 and 3, 1, 2. For in the first he or she brings in enough votes to change player 2's 48 votes into 97 and in the second he or she brings in enough votes to change player 3's 3 votes to 52. Thus,

$$p_1 = \frac{2}{3!} = \frac{1}{3}.$$

Similarly, player 2 is pivotal in permutations 1, 2, 3 and 3, 2, 1, and player 3 in permutations 1, 3, 2 and 2, 3, 1. Hence,

$$p_2 = p_3 = \frac{1}{3}.$$

Thus, as anticipated, the small third party has the same power as the two larger parties.

In the game $[51; 40, 30, 15, 15]$, the possible permutations are shown in Table 2.10, and the pivotal player in each is circled. Player 1 is pivotal 12 times, so his or her power is $12/4! = 1/2$. Players 2, 3, and 4 are each pivotal 4 times, so they each have power $4/4! = 1/6$.

Let us compute the Shapley-Shubik power index for the Australian government, the game $[5; 1, 1, 1, 1, 1, 1, 3]$. In this and the following examples, the enumeration of all permutations is not the most practical method for computing the Shapley-Shubik power index. We proceed by another method. The federal government (player 7) is pivotal in a given permutation if and only if it is third, fourth, or fifth. By symmetry, we observe that the federal government is picked in the ith position

Table 2.10: All Permutations of the Players in the Game $[51; 40, 30, 15, 15]$, with Pivotal Player Circled

1 2 3 4	2 1 3 4	3 1 2 4	4 1 2 3
1 2 4 3	2 1 4 3	3 1 4 2	4 1 3 2
1 3 2 4	2 3 1 4	3 2 1 4	4 2 1 3
1 3 4 2	2 3 4 1	3 2 4 1	4 2 3 1
1 4 2 3	2 4 1 3	3 4 1 2	4 3 1 2
1 4 3 2	2 4 3 1	3 4 2 1	4 3 2 1

in a permutation of the 7 players in exactly 1 out of every 7 permutations. Hence, it is picked third, fourth, or fifth in exactly 3 out of every 7 permutations. Thus, the probability that the federal government is pivotal is 3/7; that is,

$$p_7 = \frac{3}{7}.$$

We can also see this by observing that the number of permutations of the seven players in which the federal government is third is 6!, for we have to order the remaining players. Similarly, the number of permutations in which the federal government is fourth (or fifth) is also 6!. Thus,

$$p_7 = \frac{3 \cdot 6!}{7!} = \frac{3}{7}.$$

Now it is easy to see that

$$p_1 + p_2 + \cdots + p_7 = 1.$$

It is always the case that

$$\sum_i p_i = 1.$$

(Why?) Hence,

$$p_1 + p_2 + p_3 + p_4 + p_5 + p_6 = 1 - \frac{3}{7} = \frac{4}{7}.$$

Since by symmetry

$$p_1 = p_2 = \cdots = p_6,$$

each of these numbers is equal to

$$\frac{4}{7} \div 6 = \frac{2}{21}.$$

Thus, although the federal government has only 3 times the number of votes of a state, it has $4\frac{1}{2}$ times the power (9/21 vs. 2/21).

2.15.3 The U.N. Security Council

Let us turn next to the Shapley-Shubik power index for the U.N. Security Council. Let us fix a nonpermanent player i. Player i is pivotal in exactly those permutations in which all permanent players precede i and exactly three nonpermanent players precede i. How many such permutations are there? To find such a permutation, we first choose the three nonpermanent players who precede i; for each such choice, we order all eight players who precede i (the five permanent and three nonpermanent ones); for each choice and ordering, we order the remaining six nonpermanent players who follow i. The number of ways to make the first choice is $C(9,3)$, the number of ways to order the preceding players is $8!$, and the number of ways to order the following players is $6!$. Thus, by the product rule, the number of permutations in which i is pivotal is given by

$$C(9,3) \times 8! \times 6! = \frac{9!}{3!6!} \times 8! \times 6! = \frac{9!8!}{3!}.$$

Thus, since the total number of permutations of the 15 players is $15!$,

$$p_i = \frac{9!8!}{3!} \div 15! \approx .001865.$$

It follows that the sum of the powers of the nonpermanent players is 10 times this number; that is, it is .01865. Thus, since all the powers add to 1, the sum of the powers of the permanent players is .98135. There are five permanent players, each of whom, by symmetry, has equal power. It follows that each has power

$$p_j = \frac{.98135}{5} = .1963.$$

Hence, permanent members have more than 100 times the power of nonpermanent members. (The idea of calculating power in the U.N. Security Council and other legislative bodies in this manner was introduced in Shapley and Shubik [1954].)

2.15.4 Bicameral Legislatures

To give a more complicated example, suppose that we have a bicameral legislature with n_1 members in the first house and n_2 members in the second house.[25] Suppose that a measure can pass only if it has a majority in each house of the legislature, and suppose for simplicity that n_1 and n_2 are both odd. Let I be the union of the sets of members of both houses, and let π be any permutation of I. A player i in the jth house is pivotal in π if he is the $[(n_j+1)/2]$th player of his house in π and a majority of players in the other house precede him. However, for every permutation

[25] This example is also due to Shapley and Shubik [1954].

π in which a player in the first house is pivotal, the reverse permutation makes a player in the second house pivotal. (Why?) Moreover, every permutation has some player as pivotal. Thus, some player in house number 1 is pivotal in exactly $\frac{1}{2}$ of all permutations. Since all players in house number 1 are treated equally, any one of these players is pivotal in exactly $1/(2n_1)$ of the permutations. Similarly, any player of house number 2 will be pivotal in exactly $1/(2n_2)$ of the permutations. Thus, each player of house number 1 has power $1/(2n_1)$ and each player of house number 2 has power $1/(2n_2)$. In the U.S. House and Senate, $n_1 = 435$ and $n_2 = 101$, including the Vice-President who votes in case of a tie. According to our calculation, each representative has power $1/870 \approx .0011$ and each senator (including the Vice-President) has power $1/202 \approx .005$. Thus, a senator has about five times as much power as a representative.

Next, let us add an executive (a governor, the President) who can veto the vote in the two houses, but let us assume that there is no possibility of overriding the veto. Now there are $n_1 + n_2 + 1$ players and a coalition is winning if and only if it contains the executive and a majority from each house. Assuming that n_1 and n_2 are large, Shapley and Shubik [1954] argue that the executive will be pivotal in approximately one-half of the permutations. (This argument is a bit complicated and we omit it.) The two houses divide the remaining power almost equally. Finally, if the possibility of overriding the veto with a two-thirds majority of both houses is added, a similar discussion implies that the executive has power approximately one-sixth, and the two houses divide the remaining power almost equally. The reader is referred to Shapley and Shubik's paper for details.

Similar calculations can be made for the relative power that various states wield in the electoral college. Mann and Shapley [1964a,b] calculated this using the distribution of electoral votes as of 1961. New York had 43 out of the total of 538 electoral votes, and had a power of .0841. This compared to a power of .0054 for states like Alaska, which had three electoral votes. According to the Shapley-Shubik power index, the power of New York exceeded its percentage of the vote, whereas that of Alaska lagged behind its percentage.

Similar results for the distribution of electoral votes as of 1972 were obtained by Boyce and Cross [unpublished observations, 1973]. In the 1972 situation, New York had a total of 41 electoral votes (the total was still 538) and a power of .0797, whereas Alaska still had three electoral votes and a power of .0054. For a more comprehensive discussion of power in electoral games, see Brams, Lucas, and Straffin [1983], Lucas [1983], Shapley [1981], and Straffin [1980].

2.15.5 Cost Allocation

Game-theoretic solutions such as the Shapley-Shubik power index and the more general Shapley value have long been used to allocate costs to different users in shared projects. Examples of such applications include allocating runway fees to different users of an airport, highway fees to different-size trucks, costs to different colleges sharing library facilities, and telephone calling charges among users. See Lucas [1981a], Okada, Hashimoto, and Young [1982], Shubik [1962], Straffin and

Heaney [1981], and Young [1986].

These ideas have found fascinating recent application in multicast transmissions, for example, of movies over the Internet. In unicast routing, each packet sent from a source is delivered to a single receiver. To send the same packet to multiple sites requires the source to send multiple copies of the packet and results in a large waste of bandwidth. In multicast routing, we use a "directed tree" connecting the source to all receivers, and at branch points a packet is duplicated as necessary. The bandwidth used by a multicast transmission is not directly attributable to a single receiver and so one has to find a way to distribute the cost among various receivers. Feigenbaum, Papadimitriou, and Shenker [2000], Herzog, Shenker, and Estrin [1997], and Jain and Vazirani [2001] applied the Shapley value to determining cost distribution in the multicasting application and studied the computational difficulty of implementing their methods.

2.15.6 Characteristic Functions

We have concentrated in this section on simple games, games that can be defined by giving each coalition S a *value* $v(S)$ equal to 0 or 1. The value is often interpreted as the best outcome a coalition can guarantee itself through cooperation. If the value function or *characteristic function* $v(S)$ can take on arbitrary real numbers as values, the game is called a game in *characteristic function form*. Such games have in recent years found a wide variety of applications, such as in water and air pollution, disarmament, and bargaining situations. For a summary of applications, see Brams, Schotter, and Schwödianer [1979] or Lucas [1981a]. For more information about the theory of games in characteristic function form, see Fudenberg and Tirole [1991], Myerson [1997], Owen [1995], or Roberts [1976].

For more on game theory in general, see, for example, Aumann and Hart [1998], Fudenberg and Tirole [1991], Jones [1980], Lucas [1981b], Myerson [1997], Owen [1995], or Stahl [1998].

EXERCISES FOR SECTION 2.15

1. For each of the following weighted majority games, describe all winning coalitions.
   (a) $[65; 50, 30, 20]$ (b) $[125; 160, 110, 10]$
   (c) The Board of Supervisors, Nassau County, NY, 1964: $[59; 31, 31, 21, 28, 2, 2]$
   (d) $[80; 44, 43, 42, 41, 5]$ (e) $[50; 35, 35, 35, 1]$
2. For the following weighted majority games, identify all *minimal* winning coalitions, that is, winning coalitions with the property that removal of any player results in a losing coalition.
   (a) $[14; 6, 6, 8, 12, 2]$ (b) $[60; 58, 7, 1, 1, 1, 1]$
   (c) $[20; 6, 6, 6, 6]$ (d) All games of Exercise 1

3. Calculate the Shapley-Shubik power index for each player in the following weighted majority games.
   (a) $[51; 49, 47, 4]$
   (b) $[201; 100, 100, 100, 100, 1]$
   (c) $[151; 100, 100, 100, 1]$
   (d) $[51; 26, 26, 26, 22]$
   (e) $[20; 8, 8, 4, 2]$ (*Hint:* Is player 4 ever pivotal?)
4. Calculate the Shapley-Shubik power index for the following games.
   (a) $[16; 9, 9, 7, 3, 1, 1]$. (This game arose in the Nassau County, New York, Board of Supervisors in 1958; see Banzhaf [1965].)
   (b) $[59; 31, 31, 21, 28, 2, 2]$. (This game arose in the Nassau County, New York, Board of Supervisors in 1964; again see Banzhaf [1965].)
5. Consider a conference committee consisting of three senators, x, y, and z, and three members of the House of Representatives, a, b, and c. A measure passes this committee if and only if it receives the support of at least two senators and at least two representatives.
   (a) Identify the winning coalitions of this simple game.
   (b) Show that this game is not a weighted majority game. That is, we cannot find votes $v(x), v(y), v(z), v(a), v(b)$, and $v(c)$ and a quota q such that a measure passes if and only if the sum of the votes in favor of it is at least q. (*Note:* A similar argument shows that, in general, a bicameral legislature cannot be thought of as a weighted majority game.)
6. Which of the following defines a weighted majority game in the sense that there is a weighted majority game with the same winning coalitions? Give a proof of your answer.
   (a) Three players, and a coalition wins if and only if player 1 is in it.
   (b) Four players, a, b, x, y; a coalition wins if and only if at least a or b and at least x or y is in it.
   (c) Four players and a coalition wins if and only if at least three players are in it.
7. Suppose that a country has 3 provinces. The number of representatives of each province in the state legislature is given as follows: Province A has 6, province B has 7, and province C has 2. If all representatives of a province vote alike, and a two-thirds majority of votes is needed to win, find the power of each province using the Shapley-Shubik power index.
8. Calculate the Shapley-Shubik power index for the conference committee (Exercise 5).
9. Prove that in a bicameral legislature, for every permutation in which a player in the first house is pivotal, the reverse permutation makes a player in the second house pivotal.
10. (Lucas [1983]) In the original Security Council, there were five permanent members and only six nonpermanent members. The winning coalitions consisted of all five permanent members plus at least two nonpermanent members.
    (a) Formulate this as a weighted majority game.
    (b) Calculate the Shapley-Shubik power index.

11. (Lucas [1983]) It has been suggested that Japan be added as a sixth permanent member of the Security Council. If this were the case, assume that there would still be 10 nonpermanent members and winning coalitions would consist of all six permanent members plus at least four nonpermanent members.

    (a) Formulate this as a weighted majority game.

    (b) Calculate the Shapley-Shubik power index.

12. Compute the Shapley-Shubik power index of a player with 1 vote in the game in which 6 players have 11 votes each, 12 players have 1 vote each, and 71 votes are needed to win.

13. If we do not require that every subset of a losing coalition is a losing coalition or that for all S, either S or $1 - S$ is losing, then how many different simple games are there on a set of n players?

14. In a simple game, if p_i is the Shapley-Shubik power index for player i and $\sum_{i \in S} p_i$ is greater than $\frac{1}{2}$, is S necessarily a winning coalition? Why?

15. Suppose that $v(S)$ gives 1 if coalition S is winning and 0 if S is losing. If p_i is the Shapley-Shubik power index for player i, show that

$$p_i = \sum \{\gamma(s)[v(S) - v(S - \{i\})] : S \text{ such that } i \in S\}, \tag{2.9}$$

where

$$s = |S| \text{ and } \gamma(s) = \frac{(s-1)!(n-s)!}{n!}.$$

16. Apply formula (2.9) in Exercise 15 to compute the Shapley-Shubik power index for each of the weighted majority games in Exercise 3.

17. It is usually assumed that if v is a characteristic function, then

$$v(\emptyset) = 0 \tag{2.10}$$

$$v(S \cup T) \geq v(S) + v(T) \text{ if } S \cap T = \emptyset. \tag{2.11}$$

Which of the following characteristic functions on $I = \{1, 2, \ldots, n\}$ have these two properties?

    (a) $n = 3$, $v(\emptyset) = 0$, $v(\{1\}) = v(\{2\}) = v(\{3\}) = -1$, $v(\{1, 2\}) = 3$, $v(\{1, 3\}) = 3$, $v(\{2, 3\}) = 4$, $v(\{1, 2, 3\}) = 2$.

    (b) $n = 3$, $v(\emptyset) = 0$, $v(\{1\}) = v(\{2\}) = 0$, $v(\{3\}) = -1$, $v(\{1, 2\}) = 1$, $v(\{1, 3\}) = 0$, $v(\{2, 3\}) = -1$, $v(\{1, 2, 3\}) = 0$.

    (c) Arbitrary n, $v(S) = -|S|$, for all S.

18. Show that the following characteristic function on $I = \{1, 2, 3, 4\}$ satisfies conditions (2.10) and (2.11) of Exercise 17.[26]

[26] This and the next exercise are unpublished exercises due to A. W. Tucker.

$$\begin{aligned} v(\emptyset) &= 0 \\ v(\{i\}) &= 0 && \text{(all } i) \\ v(\{i,j\}) &= \frac{i+j}{10} && \text{(all } i \neq j) \\ v(\{i,j,k\}) &= \frac{i+j+k}{10} && \text{(all distinct } i,j,k) \\ v(\{1,2,3,4\}) &= \frac{1+2+3+4}{10}. \end{aligned}$$

19. Generalizing Exercise 18, let $I = \{1, 2, \ldots, 2n\}$, and let

$$\begin{aligned} v(\emptyset) &= 0 \\ v(\{i\}) &= 0 && \text{(all } i) \\ v(S) &= \sum_{i \in S} c_i && \text{(for each } S \text{ with } |S| > 1, \end{aligned}$$

where the c_i are $2n$ positive constants with sum equal to 1. Verify that v satisfies conditions (2.10) and (2.11) of Exercise 17.

20. In the game called *deterrence*, each of the n players has the means to destroy the wealth of any other player. If w_i is the wealth of player i, then $v(S)$ is given by

$$v(S) = \begin{cases} -\sum_{i \in S} w_i & \text{(if } |S| < n) \\ 0 & \text{(if } |S| = n). \end{cases}$$

Show that $v(S)$ satisfies conditions (2.10) and (2.11) of Exercise 17.

21. In the game called *pure bargaining*, a private foundation has offered n states a total of d dollars for development of water pollution abatement facilities provided that the states can agree on the distribution of the money. In this game $v(S)$ is given by

$$v(S) = \begin{cases} 0 & \text{(if } |S| < n) \\ d & \text{(if } |S| = n). \end{cases}$$

Show that $v(S)$ satisfies conditions (2.10) and (2.11) of Exercise 17.

22. In the game called *two-buyer market*, player 1 owns an object worth a units to him. Player 2 thinks the object is worth b units and player 3 thinks it is worth c units. Assuming that $a < b < c$, the characteristic function of this game is given by

$$\begin{aligned} v(\{1\}) &= a & v(\{2\}) &= 0 & v(\{3\}) &= 0 & v(\{1,2\}) &= b \\ v(\{1,3\}) &= c & v(\{2,3\}) &= 0 & v(\{1,2,3\}) &= c. \end{aligned}$$

Show that $v(S)$ satisfies conditions (2.10) and (2.11) of Exercise 17.

23. (Shapley and Shubik [1969]) In the *garbage game*, each of n players has a bag of garbage that he or she *must* drop in someone else's yard. The utility or worth of b bags of garbage is $-b$. Then

$$v(S) = \begin{cases} 0 & \text{(if } s = 0) \\ -(n-s) & \text{(if } 0 < s < n) \\ -n & \text{(if } s = n), \end{cases}$$

where $s = |S|$. Show that $v(S)$ satisfies conditions (2.10) and (2.11) of Exercise 17.

24. Formula (2.9) of Exercise 15 can be used as the definition of the Shapley-Shubik power index for a game in characteristic function form. Using this formula:
    (a) Calculate the Shapley-Shubik power index for the game of Exercise 20.
    (b) Calculate the Shapley-Shubik power index for the game of Exercise 21.
    (c) Calculate the Shapley-Shubik power index for the game of Exercise 22 if $a = 3$, $b = 5$, and $c = 10$.
    (d) Calculate the Shapley-Shubik power index for the game of Exercise 23.
    (e) Calculate the Shapley-Shubik power index for the game of Exercise 19.

2.16 GENERATING PERMUTATIONS AND COMBINATIONS[27]

In Examples 2.10 and 2.17 we discussed algorithms that would proceed by examining every possible permutation of a set. (Other times, we may be interested in every r-combination or every subset.) We did not comment there on the problem of determining in what order to examine the permutations, because we were making the point that such algorithms are not usually very efficient. However, there are occasions when such algorithms are useful. In connection with them, we need a procedure to generate all permutations of a set and in general, all members of a certain class of combinatorial objects. In this section we describe such procedures.

2.16.1 An Algorithm for Generating Permutations

A natural order in which to examine permutations is the *lexicographic order*. To describe this order, suppose that $\pi = \pi_1\pi_2\pi_3$ and $\sigma = \sigma_1\sigma_2\sigma_3$ are two permutations of the set $\{1,2,3\}$. We say that π *precedes* σ if $\pi_1 < \sigma_1$ or if $\pi_1 = \sigma_1$ and $\pi_2 < \sigma_2$. (Note that if $\pi_1 = \sigma_1$ and $\pi_2 = \sigma_2$, then $\pi_3 = \sigma_3$.) For instance, $\pi = 123$ precedes $\sigma = 231$ since $\pi_1 = 1 < 2 = \sigma_1$, and $\pi = 123$ precedes $\sigma = 132$ because $\pi_1 = 1 = \sigma_1$ and $\pi_2 = 2 < 3 = \sigma_2$. More generally, suppose that $\pi = \pi_1\pi_2\cdots\pi_n$ and $\sigma = \sigma_1\sigma_2\cdots\sigma_n$ are two permutations of the set $\{1,2,\ldots,n\}$. Then π *precedes* σ if $\pi_1 < \sigma_1$ or if $\pi_1 = \sigma_1$ and $\pi_2 < \sigma_2$ or if $\pi_1 = \sigma_1$ and $\pi_2 = \sigma_2$ and $\pi_3 < \sigma_3$ or $\ldots$ or if $\pi_1 = \sigma_1$ and $\pi_2 = \sigma_2$ and $\ldots$ and $\pi_k = \sigma_k$ and $\pi_{k+1} < \sigma_{k+1}$ or if $\ldots$. Thus, $\pi = 42135$ precedes $\sigma = 42153$ because $4 = 4, 2 = 2, 1 = 1$, and $3 < 5$. In this lexicographic order, we order as we do words in a dictionary, considering first the first "letter," then in case of ties the second "letter," and so on. The following lists all permutations of $\{1,2,3\}$ in lexicographic order:

$$123, 132, 213, 231, 312, 321. \tag{2.12}$$

Notice that in terms of permutations of a set of numbers $\{1,2,\ldots,n\}$, if $n \le 9$ the permutations can be thought of "numbers" themselves. For example, the permutation 3241 can be thought of as three-thousand-two-hundred-forty-one. In

[27]This section may be omitted.

these cases, the lexicographic order of the permutations will be equivalent to the increasing order of the "numbers." In (2.12), the permutations in lexicographic order increase from the number one-hundred-twenty-three to three-hundred-twenty-one.

We shall describe an algorithm for listing all permutations in lexicographic order. The key step is to determine, given a permutation $\pi = \pi_1\pi_2\cdots\pi_n$, what permutation comes next. The last permutation in the lexicographic order is $n(n-1)(n-2)\ldots 21$. This has no next permutation in the order. Any other permutation π has $\pi_i < \pi_{i+1}$ for some i. If $\pi_{n-1} < \pi_n$, the next permutation in the order is obtained by interchanging π_{n-1} and π_n. For instance, if $\pi = 43512$, then the next permutation is 43521. Now suppose that $\pi_{n-1} > \pi_n$. If $\pi_{n-2} < \pi_{n-1}$, we rearrange the last three entries of π to obtain the next permutation in the order. Specifically, we consider π_{n-1} and π_n and find the smallest of these which is larger than π_{n-2}. We put this in the $(n-2)$nd position. We then order the remaining two of the last three digits in increasing order. For instance, suppose that $\pi = 15243$. Then $\pi_{n-1} = 4 > 3 = \pi_n$ but $\pi_{n-2} = 2 < 4 = \pi_{n-1}$. Both π_{n-1} and π_n are larger than π_{n-2} and 3 is the smaller of π_{n-1} and π_n. Thus, we put 3 in the third position and put 2 and 4 in increasing order, obtaining the permutation 15324. If π is 15342, we switch 4 into the third position, not 2, since $2 < 3$, and obtain 15423.

In general, if $\pi \neq n, n-1, n-2, \ldots, 2, 1$, there must be a rightmost i so that $\pi_i < \pi_{i+1}$. Then the elements from π_i and on must be rearranged to find the next permutation in the order. This is accomplished by examining all π_j for $j > i$ and finding the smallest such π_j that is larger than π_i. Then π_i and π_j are interchanged. Having made the interchange, the numbers following π_j after the interchange are placed in increasing order. They are now in decreasing order, so simply reversing them will suffice. For instance, suppose that $\pi = 412653$. Then $\pi_i = 2$ and $\pi_j = 3$. Interchanging π_i and π_j gives us 413652. Then reversing gives us 413256, which is the next permutation in the lexicographic order.

The steps of the algorithm are summarized as follows.

Algorithm 2.1: Generating All Permutations of $1, 2, \ldots, n$

Input: n.
Output: A list of all $n!$ permutations of $\{1, 2, \ldots, n\}$, in lexicographic order.

Step 1. Set $\pi = 12\cdots n$ and output π.

Step 2. If $\pi_i > \pi_{i+1}$ for all i, stop. (The list is complete.)

Step 3. Find the largest i so that $\pi_i < \pi_{i+1}$.

Step 4. Find the smallest π_j so that $i < j$ and $\pi_i < \pi_j$.

Step 5. Interchange π_i and π_j.

Step 6. Reverse the numbers following π_j in the new order, let π denote the resulting permutation, output π, and return to step 2.

Note that Algorithm 2.1 can be modified so that as a permutation is generated, it is examined for one purpose or another. For details of a computer implementation of Algorithm 2.1, see, for example, Reingold, Nievergelt, and Deo [1977].

2.16.2 An Algorithm for Generating Subsets of Sets

In Section 2.6 we considered the problem of finding all possible pizzas given a particular set of toppings. This was tantamount to finding all subsets of a given set. In this section we describe an algorithm for doing so.

We start by supposing that S is a subset of the set $\{1, 2, \ldots, n\}$. An equivalent way to denote S is by a bit string B of length n, where a 1 in B's ith spot indicates that i is in S and a 0 in B's ith spot indicates that i is not. For instance, if $S = \{1, 3, 4, 6\} \subseteqq \{1, 2, 3, 4, 5, 6, 7\}$ then $B = 1011010$. Thus, the problem of generating all subsets of an n-set becomes the problem of generating all bit strings of length n.

An ordering similar to the lexicographic ordering that we introduced for permutations will be used for these bit strings. Suppose that $\alpha = \alpha_1\alpha_2\alpha_3$ and $\beta = \beta_1\beta_2\beta_3$ are two bit strings of length 3. We say that α *precedes* β if $\alpha_1 < \beta_1$ or if $\alpha_1 = \beta_1$ and $\alpha_2 < \beta_2$ or if $\alpha_1 = \beta_1$, $\alpha_2 = \beta_2$, and $\alpha_3 < \beta_3$. For instance, $\alpha = 001$ precedes $\beta = 010$ since $\alpha_1 = 0 = \beta_1$ and $\alpha_2 = 0 < 1 = \beta_2$. More generally (and more succinctly), suppose that $\alpha = \alpha_1\alpha_2\cdots\alpha_n$ and $\beta = \beta_1\beta_2\cdots\beta_n$ are two bit strings of length n. Then α precedes β if $\alpha_i < \beta_i$ for the smallest i in which α_i and β_i differ. Thus, $\alpha = 01010$ precedes $\beta = 01100$ because $\alpha_1 = 0 = \beta_1$, $\alpha_2 = 1 = \beta_2$, and $\alpha_3 = 0 < 1 = \beta_3$. In this "lexicographic" order, we again order as we do words in a dictionary; however, in this case we are dealing with a restricted "alphabet," 0 and 1. The following lists all bit strings of length 3 in lexicographic order:

$$000, 001, 010, 011, 100, 101, 110, 111. \tag{2.13}$$

Treating bit strings like numbers and using an increasing ordering is another way to think about a lexicographic ordering of subsets of a set. Notice that in (2.13), the length 3 bit strings go from zero to one-hundred-eleven and increase in between.

Next, we describe an algorithm for listing all bit strings in lexicographic order. Given a bit string, what bit string comes next? Since $11\cdots1$ has no 0's, it will not precede any other bit string by our definition of *precede* above and will thus be last in the order. Any other bit string β has $\beta_i = 0$ for some i. The next bit string after β in the order is obtained by starting at β_n and working backwards, changing all occurrences of 1's to 0's, and vice versa. By stopping the process the first time a 0 is changed to a 1, the next bit string in the order is obtained. For instance, suppose that $\beta = 1001011$. We change the places of β_5, β_6, and β_7 (which are in bold below) to obtain the next bit string, i.e.,

$$1001\mathbf{011} \to 1001\mathbf{100}.$$

Alternatively, if we think of the bit strings as "numbers," then the next bit string after β is the next-largest bit string. This can be found by adding 1 to β. It is not hard to see that adding 1 to β will have the same effect as what was described above.

The steps of this algorithm are summarized as follows.

Algorithm 2.2: Generating All Bit Strings of Length n

Input: n.
Output: A list of all 2^n bit strings of length n, in lexicographic order.

Step 1. Set $\beta = 00\cdots 0$ and output β.

Step 2. If $\beta_i = 1$ for all i, stop. (The list is complete.)

Step 3. Find the largest i so that $\beta_i = 0$.

Step 4. Change β_i to 1 and $\beta_{i+1}, \beta_{i+2}, \ldots, \beta_n$ to 0, let β denote the resulting bit string, output β, and return to step 2.

There are certainly other orderings for bit strings of length n (and permutations of $\{1, 2, \ldots, n\}$) than the ones of the lexicographic type. We describe another ordering of all of the bit strings of length n. This new order in which we will examine these bit strings is called the *binary-reflected Gray code order*.[28] (The reason for the term *binary-reflected* should become clear when we describe the ordering below. The use of the word *code* comes from its connection to coding theory; see Chapter 10.)

The binary-reflected Gray code order for bit strings of length n, denoted $G(n)$, can easily be defined recursively. That is, we will define the binary-reflected Gray code order for bit strings of length n in terms of the binary-reflected Gray code order for bit strings of length less than n. We will use the notation $G_i(n)$ to refer to the ith bit string in the ordering $G(n)$. Normally, a binary-reflected Gray code order begins with the all-0 bit string, and recall that the number of subsets of an n-element set is 2^n. Thus, $G(1)$ is the order that starts with the bit string $0 = G_1(1)$ and ends with the bit string $1 = G_2(1)$, i.e.,

$$G(1) = 0, 1.$$

To find $G(2)$, we list the elements of $G(1)$ and attach a 0 at the beginning of each element. Then list the elements of $G(1)$ in "reverse order" and attach a 1 at the beginning of each of these elements. Thus,

$$G(2) = 0G_1(1), 0G_2(1), 1G_2(1), 1G_1(1) = 00, 01, 11, 10. \tag{2.14}$$

This same procedure is used whether we are going from $G(1)$ to $G(2)$ or from $G(n)$ to $G(n+1)$.

$$G(n+1) = 0G_1(n), 0G_2(n), \ldots, 0G_{2^n}(n), 1G_{2^n}(n), 1G_{2^n-1}(n), \ldots, 1G_1(n). \tag{2.15}$$

Letting $G(n)^R$ be the reverse order of $G(n)$ and with a slight abuse of notation, $G(n+1)$ can be defined as

$$G(n+1) = 0G(n), 1G(n)^R. \tag{2.16}$$

[28] This order is based on work due to Gray [1953].

Thus,

$$G(3) = 0G(2), 1G(2)^R = 000, 001, 011, 010, 110, 111, 101, 100.$$

Notice that we doubled the number of elements in going from $G(1)$ to $G(2)$ and $G(2)$ to $G(3)$. This is not an anomaly. $|G(2)| = 2^2 = 2 \cdot 2^1 = 2|G(1)|$ and in general

$$|G(n+1)| = 2^{n+1} = 2 \cdot 2^n = 2|G(n)|.$$

$G(n+1)$ as defined in Equation (2.15) has $2 \cdot 2^n = 2^{n+1}$ elements and no duplicate elements (Why?). Therefore, the recursively defined binary-reflected Gray code order $G(i)$ is in fact an ordering of all of the bit strings of length i, $i = 1, 2, \ldots$. It is left to the reader (Exercise 15) to find an algorithm for listing all terms in $G(n)$ directly, as opposed to recursively. Again, the key step will be to produce, given a length n bit string $\beta = \beta_1\beta_2 \cdots \beta_n$, the length n bit string that comes next in $G(n)$.

Although the lexicographic order for this problem is probably more intuitive, the binary-reflected Gray code order is better in another sense. It is sometimes important that the change between successive elements in an ordering be kept to a minimum. In this regard, the binary-reflected Gray code order is certainly more efficient than the lexicographic order. Successive elements in the binary-reflected Gray code order differ in only one spot. This is obviously best possible. The lexicographic order for bit strings of length n will always have n changes for some pair of successive elements. (Why?) Notice that in (2.13), there are $n = 3$ spots which change when going from 011 to 100.

2.16.3 An Algorithm for Generating Combinations

The next combinatorial objects that we defined in this chapter after permutations and subsets were the r-combinations of an n-set. In terms of the preceding section's subsets of a set, these can be thought of as those subsets of $\{1, 2, \ldots, n\}$ of size exactly r or those bit strings of length n with exactly r 1's. Because of this association, the algorithm for their generation follows closely from Algorithm 2.2.

Suppose that we are interested in generating the 3-combinations of a 5-set. Since we are dealing with subsets, order does not matter. Thus,

$$\{1, 2, 5\}, \{1, 5, 2\}, \{2, 1, 5\}, \{2, 5, 1\}, \{5, 1, 2\}, \{5, 2, 1\}$$

are all considered the same. Therefore, as a matter of choice, our algorithm will generate the r-combinations with each subset's elements in increasing order. So, of the 6 identical subsets listed above, our algorithm will generate $\{1, 2, 5\}$. There are $\binom{5}{3} = 10$ 3-combinations of our 5-set which in increasing lexicographic order are

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

Our algorithm for generating r-combinations of an n-set works in the following way. Given an r-combination $\gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_r\}$, find the largest i such that $\{\gamma_1, \gamma_2, \ldots, \gamma_{i-1}, \gamma_i + 1, \gamma_{i+1}, \ldots, \gamma_r\}$ is an r-combination whose elements are still in increasing order. Then reset $\gamma_{i+1}, \gamma_{i+2}, \ldots, \gamma_r$ to their lowest possible values. For example, suppose that $\gamma = \{1, 3, 4, 7, 8\}$ is a 5-combination of the 8-set $\{1, 2, \ldots, 8\}$ whose elements are in increasing order. Then, $i = 3$ and $\gamma_i = 4$. Incrementing γ_i by one yields $\{1, 3, 5, 7, 8\}$. Then γ_4 and γ_5 can be reset to 6 and 7, respectively, giving us $\{1, 3, 5, 6, 7\}$, which is the next 5-combination in the increasing lexicographic order. The algorithm starts with $\{1, 2, \ldots, r\}$ and ends with $\{n - r + 1, n - r + 2, \ldots, n\}$ since no element in the latter can be incremented.

This algorithm, like our previous lexicographic algorithms, is not efficient in terms of minimizing changes between successive terms in the order. A binary-reflected Gray code order, similar to the one in the preceding section, is most efficient in this regard. Recall that $G(n + 1)$ is defined recursively by (2.16). Next, let $G(n, r)$, $0 \leq r \leq n$, be the binary-reflected Gray code order for bit strings of length n with exactly r 1's. $G(n, r)$, $0 < r < n$, can be defined recursively by

$$G(n, r) = 0G(n - 1, r), 1G(n - 1, r - 1)^R$$

and

$$G(n, 0) = 0^n = \underbrace{00 \cdots 0}_{n \text{ of them}} \quad \text{and} \quad G(n, n) = 1^n = \underbrace{11 \cdots 1}_{n \text{ of them}}.$$

In the rest of the cases, the first and last bit strings in $G(n, r)$ will be

$$0^{n-r}1^r \text{ and } 10^{n-r}1^{r-1},$$

respectively. For example,

$$\begin{array}{ll} G(1,0) = 0 & G(2,0) = 00 \\ G(1,1) = 1 & G(2,1) = 01, 10 \\ & G(2,2) = 11 \end{array}$$

$$\begin{array}{ll} G(3,0) = 000 & G(4,0) = 0000 \\ G(3,1) = 001, 010, 100 & G(4,1) = 0001, 0010, 0100, 1000 \\ G(3,2) = 011, 110, 101 & G(4,2) = 0011, 0110, 0101, 1100, 1010, 1001 \\ G(3,3) = 111 & G(4,3) = 0111, 1101, 1110, 1011 \\ & G(4,4) = 1111. \end{array}$$

Since each bit string must contain exactly r 1's, the best that could be hoped for between successive terms of an order is that at most two bits differ. The binary-reflected Gray code order for bit strings of length n with exactly r 1's does in fact attain this minimum.

See Reingold, Nievergelt, and Deo [1977] for more algorithms for generating permutations, subsets, and combinations, in addition to compositions and partitions. An early but comprehensive paper on generating permutations and combinations is Lehmer [1964].

EXERCISES FOR SECTION 2.16

1. For each of the following pairs of permutations, determine which comes first in the lexicographic order.

(a) 3412 and 2143 (b) 3124 and 3214
(c) 234651 and 235164 (d) 76813254 and 76813524

2. For each of the following pairs of bit strings, determine which comes first in the lexicographic order.

(a) 1001 and 1101 (b) 0101 and 0110
(c) 11011 and 11101 (d) 110110101 and 110101100

3. For each of the following pairs of combinations, determine which comes first in the increasing lexicographic order.

(a) $\{1,2,3,4\}$ and $\{1,2,3,5\}$ (b) $\{1,2,6,7,9,16\}$ and $\{1,2,6,7,10,13\}$
(c) $\{2,4,5,6\}$ and $\{1,3,5,6\}$ (d) $\{8,10,11,12,13\}$ and $\{8,9,10,11,13\}$

4. For each of the following permutations, find the permutation immediately following it in the lexicographic order.

(a) 123456 (b) 3457621 (c) 152463
(d) 82617543 (e) 4567321 (f) 54328716

5. For each of the following bit strings, find the bit string immediately following it in the binary-reflected Gray code order.

(a) 011010 (b) 0101001 (c) 0000
(d) 11111 (e) 1110111 (f) 011111

6. For each of the following r-combinations of an 8-set, find the combination immediately following it in the increasing lexicographic order.

(a) $\{1,3,5,6,8\}$ (b) $\{2,3,4,7\}$
(c) $\{1,3,4,5,6,7\}$ (d) $\{4,5,6,7,8\}$

7. List all permutations of $\{1,2,3,4\}$ in lexicographic order.

8. List all permutations of $\{u,v,x,y\}$ in lexicographic order.

9. List all bit stings of length 4 in lexicographic order.

10. In JAVA (Example 2.8), consider all variables of length 3. If a letter precedes an underscore, an underscore precedes a dollar sign, and all of these precede a decimal digit, find the first and last 3-character variables in lexicographic order.

11. List all 2-combinations of $\{1,2,3,4,5\}$ in increasing lexicographic order.

12. Find a bit string of length n which changes in all n of its spots when producing the next one in lexicographic order.

13. Find $G(4)$.

14. Using the recursive definitions in Equations (2.14) and (2.15), prove that $G(n)$ has no duplicate terms in the order.

15. Find an algorithm for producing all bit strings of length n in binary-reflected Gray code order.

16. Using Algorithm 2.2, prove that the lexicographic order for generating all subsets of a given n-set will always have n changes for some pair of successive terms.

17. Develop an algorithm for generating all permutations of the set $\{1,2,\ldots,n\}$ such that i does not appear in the ith spot of any permutation for all i.

18. Explain the reason for the use of the term "binary-reflected" from the binary-reflected Gray code order. [*Hint:* Refer to the procedure for finding $G(n)$ given by Equation (2.15).]
19. Prove that any two successive bit strings in a binary-reflected Gray code order differ in exactly one position.
20. Find $G(5, r)$, $0 \leq r \leq 5$.
21. Let $f_n(\pi)$ be i if π is the ith permutation in the lexicographic order of all permutations of the set $1, 2, \ldots, n$. Compute:
    (a) $f_2(21)$ (b) $f_3(231)$
    (c) $f_5(15243)$ (d) $f_6(654321)$
22. Suppose that $f_n(\pi)$ is defined as in Exercise 21 and that permutation π' is obtained from permutation $\pi = \pi_1\pi_2\cdots\pi_n$ by deleting π_1 and reducing by 1 all elements π_j such that $\pi_j > \pi_1$. Show that $f_n(\pi) = (\pi_1 - 1)(n-1)! + f_{n-1}(\pi')$.
23. Find an algorithm for generating all of the r-permutations of an n-set.
24. Using Algorithm 2.1 and the note following the algorithm, find another algorithm for generating all r-combinations of an n-set.
25. Recall the idea of a complexity function from Section 2.4. Calculate:
    (a) The complexity function for Algorithm 2.1
    (b) The complexity function for Algorithm 2.2
    (c) The complexity function for the algorithm which produces all r-combinations of an n-set in increasing lexicographic order

2.17 INVERSION DISTANCE BETWEEN PERMUTATIONS AND THE STUDY OF MUTATIONS[29]

Mutations are a key process by which species evolve. These mutations can occur in the sequences representing DNA. Sometimes the mutations can involve *inversions* (where a segment of DNA is reversed), *transpositions* (where two adjacent segments of DNA exchange places), *translocations* (where the ends of two chromosomes are exchanged), and sometimes they are more complicated. Much recent work has focused on algorithmic analysis of these mutations. For example, see Ferretti, Nadeau, and Sankoff [1996], Hannenhalli and Pevzner [1996], and Kaplan, Shamir, and Tarjan [1997]. We concentrate on inversions here.

Inversions seem to be the dominant form of mutation in some species. For example, inversions play a special role in the evolution of fruit flies (Dobzhansky [1970]), pea plants (Palmer, Osorio, and Thompson [1988]), and certain bacteria (O'Brien [1993]). At a large level, we can study inversions of genes, where genes correspond to subsequences of DNA. Genes are arranged on chromosomes and it is a reasonable starting assumption that genes on a given chromosome are distinguishable, so they

[29] This section closely follows Gusfield [1997]. It may be omitted without loss of continuity.

can be thought of as labeled $1, 2, \ldots, n$ on a given chromosome and as a permutation of $\{1, 2, \ldots, n\}$ after a series of inversions. An inversion then reverses a subsequence of the permutation. For instance, starting with the identity permutation, 123456, we can invert subsequence 2345 to get 154326, then subsequence 15 to get 514326, then subsequence 432 to get 512346. In the study of evolution, we know the current species or DNA sequence or sequence of genes on a chromosome, and try to reconstruct how we got to it. A natural question that occurs, then, is how to find the way to get from one permutation, e.g., 123456, to another, e.g., 512346, in the smallest number of steps, or in our case, the smallest number of inversions. This is called the *inversion distance* of permutation 512346 from permutation 123456, or simply the inversion distance of 512346. We have seen how to get 512346 from 123456 in three inversions. In fact, the inversion distance is at most 2: Invert subsequence 12345 to get 543216 and invert subsequence 4321 to get 512346. Thus, the inversion distance of 512346 equals 2 since it can't be 1. (Why?) We shall describe a heuristic algorithm that looks for the smallest number of inversions of the identity permutation $12 \cdots n$ required to obtain a permutation π of $\{1, 2, \ldots, n\}$, i.e., computes the inversion distance of π.

If π represents a permutation, then π_i represents the number in the ith position in π. In other words, $\pi = \pi_1 \pi_2 \cdots \pi_n$. For instance, $\pi_3 = 2$ in the permutation 512346. A *breakpoint* in π occurs between two numbers π_i and π_{i+1} if

$$|\pi_i - \pi_{i+1}| \neq 1.$$

Additionally, π has a breakpoint at its front if $\pi_1 \neq 1$ and at its end if $\pi_n \neq n$. So, $\pi = 143265$ has breakpoints between 1 and 4, between 2 and 6, and at its end.

Let $\Phi(\pi)$ equal the total number of breakpoints in π.

Theorem 2.10 The inversion distance of any permutation π is at least $\left\lceil \dfrac{\Phi(\pi)}{2} \right\rceil$.

Proof. If $\pi = \pi_1 \pi_2 \cdots \pi_n$ and the subsequence $\pi_i \pi_{i+1} \cdots \pi_{i+j}$ is inverted, then at most two new breakpoints can be created by this inversion. These two new breakpoints could occur between π_{i-1} and π_{i+j} and/or between π_i and π_{i+j+1} in the new permutation

$$\pi^* = \pi_1 \pi_2 \cdots \pi_{i-1} \pi_{i+j} \pi_{i+j-1} \cdots \pi_{i+1} \pi_i \pi_{i+j+1} \pi_{i+j+2} \cdots \pi_n.$$

Since the identity permutation has no breakpoints, at least $\lceil \Phi(\pi)/2 \rceil$ inversions are needed to transform the identity permutation into π (or vice versa). Q.E.D.

It was noted above that permutation 143265 has three breakpoints. Therefore, by Theorem 2.10, its inversion distance is at least $\lceil 3/2 \rceil = 2$.

The subsequence in a permutation between a breakpoint and (a) the front of the permutation, (b) another breakpoint, or (c) the end of the permutation, with no other breakpoint between them, is called a *strip*. If the numbers in the strip are increasing (decreasing), the strip is called *increasing* (*decreasing*). Strips consisting of a single number are defined to be decreasing. For example, the permutation

541236 has 3 breakpoints and 3 strips. The two decreasing strips are 54 and 6, while the increasing strip is 123.

The problem of finding the inversion distance of a permutation has been shown to be hard.[30] Sometimes, methods that come close to the inversion distance—or at least within a fixed factor of it—can come in handy. The next two lemmas can be used to give an algorithm for transforming any permutation into the identity permutation using a number of inversions that is at most four times the inversion distance.

Lemma 2.1 If permutation π contains a decreasing strip, then there is an inversion that decreases the number of breakpoints.

Proof.[31] Consider the decreasing strip with the smallest number π_i contained in any decreasing strip. By definition, π_i is at the right end of this strip. If $\pi_i = 1$, then $\pi_1 \neq 1$, in which case there must be breakpoints before π_1 and after π_i. Inverting $\pi_1\pi_2\cdots\pi_i$ reduces the number of breakpoints by at least one since 1 moves into the first spot of the inverted permutation.

Suppose that $\pi_i \neq 1$. Consider π_{i+1}, if it exists. It cannot be $\pi_i - 1$ since otherwise it would be in a decreasing strip. It also cannot be $\pi_i + 1$, for otherwise π_i would not be in a decreasing strip. So there must be a breakpoint between π_i and π_{i+1} or after π_i (i.e., at the end of the entire permutation) if π_{i+1} doesn't exist. By similar reasoning, there must be a breakpoint immediately to the right of $\pi_i - 1$.

If $\pi_i - 1$ is located to the right of π_i then invert $\pi_{i+1}\pi_{i+2}\cdots\pi_i - 1$. And if $\pi_i - 1$ is located to the left of π_i then invert starting with the term immediately to the right of $\pi_i - 1$ through π_i. In either case, we are inverting a subsequence with breakpoints at both its ends and reducing the number of breakpoints by at least one since $\pi_i - 1$ and π_i are now consecutive elements in the inverted permutation. Q.E.D.

For example, 54 in the permutation 541236 is the decreasing strip with the smallest number, 4. Locate the number 3 and, as must be the case, it is in an increasing sequence with a breakpoint immediately to its right. Invert 123 to get 543216. This new permutation has 2 breakpoints whereas the original had 3.

Lemma 2.2 If permutation π is not the identity and contains no decreasing strips, then there is an inversion that does not increase the number of breakpoints but creates a decreasing strip.

Proof.[32] Since there are no decreasing strips, every strip must be increasing. If $\pi_1 \neq 1$ or $\pi_n \neq n$, then inverting the increasing strip leading from π_1 or leading to π_n, respectively, will satisfy the lemma. Otherwise, find the first and second breakpoints after $\pi_1 = 1$. These exist since π is not the identity permutation. The subsequence between these two breakpoints satisfies the lemma. Q.E.D.

[30] Inversion distance calculation is NP-hard (Caprara [1997]), using the language of Section 2.18.

[31] The proof may be omitted.

[32] The proof may be omitted.

Consider the *inversion algorithm* that works as follows:

Step 1. If there is a decreasing strip, use the inversion of Lemma 2.1. Repeat until there is no decreasing strip.

Step 2. Use the inversion of Lemma 2.2 and return to Step 1.

The number of breakpoints is decreased at least once every two inversions. By Theorem 2.10, we have the following theorem.

Theorem 2.11 (Kececioglu and Sankoff [1994]) The inversion algorithm transforms any permutation into the identity permutation using a number of inversions that is at most four times the inversion distance.

The bound of using at most four times the number of inversions as the optimal is not the best known. Kececioglu and Sankoff [1995] were able to reduce the error bound in half by proving the following lemma.

Lemma 2.3 Let π be a permutation with a decreasing strip. If every inversion that reduces the number of breakpoints of π leaves a permutation with no decreasing strips, then π has an inversion that reduces the number of breakpoints by two.

With this new lemma, we know that, essentially, there always exist two successive inversions that reduce the number of breakpoints by 2. This means that we can reach the identity permutation with at most $\Phi(\pi)$ inversions, which is at most twice the inversion distance (by Theorem 2.10). Bafna and Pevzner [1996] have lowered the Kececioglu and Sankoff [1995] bound to 1.75 by considering the effects of an inversion on future inversions.

Inversion is only one type of mutation but an important one, especially in organisms of one chromosome (Sessions [1990]). Other inversion variants are addressed in the exercises. Transpositions and translocations are also interesting ways to modify a permutation. We say a few words about the former. A transposition $\pi_i\pi_{i+1}$ of the permutation $\pi = \pi_1\pi_2\cdots\pi_n$ results in the permutation $\pi_1\pi_2\cdots\pi_{i-1}\pi_{i+1}\pi_i\pi_{i+2}\cdots\pi_n$. Changing the identity permutation into another permutation (or vice versa) by transpositions of this form is a well-studied problem in combinatorics. The number of transpositions needed to do this is readily established. According to a well-known formula (see, e.g., Jerrum [1985]), the number of such transpositions required to switch an identity permutation into the permutation π of $\{1, 2, \ldots, n\}$ is given by

$$J(\pi) = |\{(i,\ j) : 1 \le i < j \le n \;\&\; \pi_i > \pi_j\}|. \tag{2.17}$$

The proof of this is left to the reader (Exercise 8). To illustrate the result, we note that the number of transpositions needed to change the permutation $\pi = 143652$ into the identity is 6 since $J(\pi) = |\{(2,3),(2,6),(3,6),(4,5),(4,6),(5,6)\}|$. More general work, motivated by considerations of mutations, allows transpositions of entire segments of a permutation. For example, we could transform 15823647 into 13658247 by transposing segments 582 and 36. For references, see the papers by Bafna and Pevzner [1998], Christie [1996], and Heath and Vergara [1998]. Transpositions have also arisen in the work of Mahadev and Roberts [2003] in applications to channel assignments in communications and physical mapping of DNA.

EXERCISES FOR SECTION 2.17

1. Which of the following permutations have inversion distance 1?

   (a) 4567123 (b) 12354678 (c) 54321

2. Which of the following permutations have inversion distance 2?

   (a) 4567123 (b) 23456781 (c) 54321

3. Give an example of a permutation with 3 breakpoints but inversion distance 3. (Note that this is another example of the fact that the bound in Theorem 2.10 is not the best possible.)

4. Prove that a strip is always increasing or decreasing.

5. Consider the following greedy algorithm for inversion distance. First find and apply the inversion that brings 1 into position π_1. Next find and apply the inversion that brings 2 into position π_2. And so on.

   (a) Apply this greedy algorithm to 512346.

   (b) Prove that this algorithm ends in at most $\Phi(\pi)$ inversions.

   (c) Find a permutation of $1, 2, \ldots, n$ that requires $n-1$ inversions using this greedy algorithm.

   (d) What is the inversion distance of your permutation in part (c)?

6. (Kececioglu and Sankoff [1995]) In the *signed inversion problem*, each number in a permutation has a sign (+ or −) that changes every time the number is involved in an inverted subsequence. For example, starting with the permutation $+1\ -5\ +4\ +3\ -2\ -6$, we can invert 543 to get $+1\ -3\ -4\ +5\ -2\ -6$. The signed inversion distance problem is to use the minimum number of inversions to transform a signed permutation into the identity permutation whose numbers have positive sign.

   (a) Give a series of inversions that transform $-3\ -4\ -6\ +7\ +8\ +5\ -2\ -1$ into the all-positive identity permutation.

   (b) In a signed permutation π, an *adjacency* is defined as a pair of consecutive numbers of the form $+i\ +(i+1)$ or $-(i+1)\ -i$. A *breakpoint* is defined as occurring between any two consecutive numbers that do not form an adjacency. Also, there is a breakpoint at the front of π unless the first number is $+1$, and there is a breakpoint at the end of π unless the last number is $+n$.

      i. How many breakpoints does $-3\ -4\ -6\ +7\ +8\ +5\ -2\ -1$ have?

      ii. Describe a bound for signed inversion distance in terms of breakpoints analogous to Theorem 2.10.

7. Find the minimum number of transpositions that transform the following permutations into the identity and identify which transpositions achieve the minimum in each case.

   (a) 54321 (b) 15423 (c) 625143

8. Prove (2.17) (Jerrum's formula).

9. Recall that *mutations due to transpositions* occur when two adjacent *segments* of DNA exchange places. Find segment transpositions that transform the following permutations into the identity permutation. What is the minimum number of such transpositions?

   (a) 5123746 (b) 987123456

10. Suppose that an organism has a circular chromosome. In this case, the permutation $\pi_1, \pi_2, \ldots, \pi_n$ is equivalent to $\pi_i, \pi_{i+1}, \ldots, \pi_n, \pi_1, \pi_2, \ldots, \pi_{i-1}$. We call such a permutation *connected*. Find a series of inversions that transform the following connected permutations into the identity permutation. What is the minimum number of inversions?

    (a) 5123746 (b) 987123456

2.18 GOOD ALGORITHMS[33]

2.18.1 Asymptotic Analysis

We have already observed in Section 2.4 that some algorithms for solving combinatorial problems are not very good. In this section we try to make precise what we mean by a good algorithm. As we pointed out in Section 2.4, the cost of running a particular computer program on a particular machine will vary with the skill of the programmer and the characteristics of the machine. Thus, in the field of computer science, the emphasis is on analyzing algorithms for solving problems rather than on analyzing particular computer programs, and that will be our emphasis here.

In analyzing how good an algorithm is, we try to estimate a complexity function $f(n)$, to use the terminology of Section 2.4. If n is relatively small, then $f(n)$ is usually relatively small, too. Most any algorithm will suffice for a small problem. We shall be mainly interested in comparing complexity functions $f(n)$ for n relatively large.

The crucial concept in the analysis of algorithms is the following. Suppose that F is an algorithm with complexity function $f(n)$ and that $g(n)$ is any function of n. We write that F or f is $O(g)$, and say that F or f is "big oh of g" if there is an integer r and a positive constant k so that for all $n \geq r$, $f(n) \leq k \cdot g(n)$. [If f is $O(g)$, we sometimes say that g *asymptotically dominates* f.] If f is $O(g)$, then for problems of input size at least r, an algorithm with complexity function f will never be more than k times as costly as an algorithm with complexity function g. To give some examples, $100n$ is $O(n^2)$ because for $n \geq 100$, $100n \leq n^2$. Also, $n + 1/n$ is $O(n)$, because for $n \geq 1$, $n + 1/n \leq 2n$. An algorithm that is $O(n)$ is called *linear*, an algorithm that is $O(n^2)$ is called *quadratic*, and an algorithm that is $O(g)$ for g a polynomial is called *polynomial*. Other important classes of algorithms in computer science are algorithms that are $O(\log n)$, $O(n \log n)$, $O(c^n)$ for $c > 1$, and $O(n!)$. We discuss these below or in the exercises.

[33] This section should be omitted in elementary treatments.

An algorithm whose complexity function is $c^n, c > 1$, is called *exponential.* Note that every exponential algorithm is $O(c^n)$, but not every $O(c^n)$ algorithm is exponential. For example, an algorithm whose complexity function is n is $O(c^n)$ for any $c > 1$. This is because $n \leq c^n$ for n sufficiently large.

A generally accepted principle is that an algorithm is *good* if it is polynomial. This idea is originally due to Edmonds [1965]. See Garey and Johnson [1979], Lawler [1976], or Reingold, Nievergelt, and Deo [1977] for a discussion of good algorithms. We shall try to give a very quick justification here.[34]

Since we are interested in $f(n)$ and $g(n)$ only for n relatively large, we introduce the constant r in defining the concept "f is $O(g)$." But where does the constant k come from? Consider algorithms F and G whose complexity functions are, respectively, $f(n) = 20n$ and $g(n) = 40n$. Now clearly algorithm F is preferable, because $f(n) \leq g(n)$ for all n. However, if we could just improve a particular computer program for implementing algorithm G so that it would run in $\frac{1}{2}$ the time, or if we could implement G on a faster machine so that it would run in $\frac{1}{2}$ the time, then $f(n)$ and $g(n)$ would be the same. Since the constant $\frac{1}{2}$ is independent of n, it is not farfetched to think of improvements by this constant factor to be a function of the implementation rather than of the algorithm. In this sense, since $f(n)/g(n)$ equals a constant, that is, since $f(n) = kg(n)$, the functions $f(n)$ and $g(n)$ are considered the same for all practical purposes.

Now, to say that f is $O(g)$ means that $f(n) \leq kg(n)$ (for n relatively large). Since $kg(n)$ and $g(n)$ are considered the same for all practical purposes, $f(n) \leq kg(n)$ says that $f(n) \leq g(n)$ for all practical purposes. Thus, to say that f is $O(g)$ says that an algorithm of complexity g is no more efficient than an algorithm of complexity f.

Before justifying the criterion of polynomial boundedness, we summarize some basic results in the following theorem.

Theorem 2.12

(a) If c is a positive constant, then f is $O(cf)$ and cf is $O(f)$.

(b) n is $O(n^2)$, n^2 is $O(n^3), \ldots, n^{p-1}$ is $O(n^p), \ldots$. However, n^p is not $O(n^{p-1})$.

(c) If $f(n) = a_q n^q + a_{q-1} n^{q-1} + \cdots + a_0$ is a polynomial of degree q, with $a_q > 0$, and if $a_i \geq 0$, all i, then f is $O(n^q)$.

(d) If $c > 1$ and $p \geq 0$, then n^p is $O(c^n)$. Moreover, c^n is not $O(n^p)$.

Part (a) of Theorem 2.12 shows that just as we have assumed, algorithms of complexity f and cf are considered equally efficient. Part (b) asserts that an $O(n^p)$ algorithm is more efficient the smaller the value of p. Part (c) asserts that the degree of the polynomial tells the relative complexity of a polynomial algorithm. Part (d) asserts that polynomial algorithms are always more efficient than exponential algorithms. This is why polynomial algorithms are treated as *good*, whereas

[34] The reader who only wants to understand the definition may skip the rest of this subsection.

Table 2.11: Growths of Different Complexity Functions

	Complexity function $f(n)$			
Input size n	n	n^2	$10n^2$	2^n
5	5	25	250	32
10	10	10^2	10^3	$1{,}024 \approx 1.02 \times 10^2$
20	20	400	4,000	$1{,}048{,}576 \approx 1.05 \times 10^6$
30	30	900	9,000	$\approx 1.07 \times 10^9$
50	50	2,500	25,000	$\approx 1.13 \times 10^{15}$
$100 = 10^2$	10^2	10^4	10^5	$\approx 1.27 \times 10^{30}$
$1{,}000 = 10^3$	10^3	10^6	10^7	$> 10^{300}$
$10{,}000 = 10^4$	10^4	10^8	10^9	$> 10^{3000}$

exponential ones are not. The results of Theorem 2.12 are vividly demonstrated in Table 2.11, which shows how rapidly different complexity functions grow. Notice how much faster the exponential complexity function 2^n grows in comparison to the other complexity functions.

Proof of Theorem 2.12.

(a) Clearly, cf is $O(f)$. Take $k = c$. Next, f is $O(cf)$ because $f(n) \leq (1/c)cf(n)$ for all n.

(b) Since $n^p \geq n^{p-1}$ for $n \geq 1$, n^{p-1} is $O(n^p)$. Now, n^p is not $O(n^{p-1})$. For $n^p \leq cn^{p-1}$ only for $n \leq c$.

(c) Note that since $a_i \geq 0$, $a_i n^i \leq a_i n^q$, for all i and all $n \geq 1$. Hence, it follows that $f(n) \leq (a_0 + a_1 + \cdots + a_q)n^q$, for all $n \geq 1$.

(d) This is a standard result from calculus or advanced calculus. It can be derived by noting that $n^p/c^n \to 0$ as $n \to \infty$. This result is obtained by applying l'Hôspital's rule (from calculus) p times. Since $n^p/c^n \to 0$, $n^p/c^n \leq k$ for n sufficiently large. A similar analysis shows that $c^n/n^p \to \infty$ as $n \to \infty$, so c^n could not be $\leq kn^p$ for $n \geq r$ and constant k. Q.E.D.

The proof of part (d) alludes to the fact that limits can be used to prove whether or not f is "big oh" of g. It is important to note that $f(n)$ and $g(n)$ should not be considered general functions of n but as nonnegative functions of n. This is the case since they measure the cost or complexity of an algorithm.

Theorem 2.13 If $g(n) > 0$ and $\displaystyle\lim_{n\to\infty} \frac{f(n)}{g(n)}$ exists, then f is $O(g)$.

Proof. Suppose that $\lim_{n\to\infty} \frac{f(n)}{g(n)} = L$. Then for every $\epsilon > 0$, there exists $N > 0$, such that $\left|\frac{f(n)}{g(n)} - L\right| < \epsilon$ whenever $n > N$. Note that

$$\left|\frac{f(n)}{g(n)} - L\right| < \epsilon$$
$$\Downarrow$$
$$\frac{f(n)}{g(n)} - L < \epsilon$$
$$\Downarrow$$
$$f(n) < \epsilon g(n) + Lg(n)$$
$$\Downarrow$$
$$f(n) < (\epsilon + L)g(n).$$

Using the definition of "big oh," the proof is completed by letting $r = N$ and $k = \epsilon + L$. Note that the proof uses the fact that $g(n) > 0$. Q.E.D.

Using Theorem 2.13 and l'Hôspital's rule from calculus, we see that $f(n) = 7\log n$ is $O(n^5)$ since

$$\lim_{n\to\infty} \frac{7\log n}{n^5} = \lim_{n\to\infty} \frac{7\left(\frac{1}{n}\right)}{5n^4} = \lim_{n\to\infty} \frac{7}{5n^5} = 0.$$

If the limit does not exist in Theorem 2.13, no conclusion can be drawn. Consider $f(n) = \sin n + 1$ and $g(n) = \cos n + 3$. Then $\lim_{n\to\infty} \frac{\sin n + 1}{\cos n + 3}$ doesn't exist. However, it is easy to see that

$$\sin n + 1 \leq k(\cos n + 3)$$

for $k = 1$ and all $n \geq 1$. Thus, f is $O(g)$. On the other hand, as we saw in the proof of Theorem 2.12(d), $\lim_{n\to\infty} \frac{c^n}{n^p} \to \infty$ as $n \to \infty$, which implies that c^n could not be $\leq kn^p$ for $n \geq r$ and constant k.

Before closing this subsection, we should note again that our results depend on the crucial "equivalence" between algorithms of complexities f and cf, and on the idea that the size n of the input is relatively large. In practice, an algorithm of complexity $100n$ is definitely worse than an algorithm of complexity n. Moreover, it is also definitely worse, for small values of n, than an algorithm of complexity 2^n. Thus, an $O(n)$ algorithm, in practice, can be worse than an $O(2^n)$ algorithm. The results of this section, and the emphasis on polynomial algorithms, must be interpreted with care.

2.18.2 NP-Complete Problems

In studying algorithms, it is convenient to distinguish between deterministic procedures and nondeterministic ones. An algorithm may be thought of as passing from state to state(s). A *deterministic algorithm* may move to only one new state at a time, while a *nondeterministic algorithm* may move to several new states at once. That is, a nondeterministic algorithm may explore several possibilities simultaneously. In this book we concentrate exclusively on deterministic algorithms, and indeed, when we use the term *algorithm*, we shall mean deterministic. The class of problems for which there is a deterministic algorithm whose complexity is polynomial is called P. The class of problems for which there is a nondeterministic algorithm whose complexity is polynomial is called NP. Clearly, every problem in P is also in NP. To this date, no one has discovered a problem in NP that can be shown not to be in P. However, there are many problems known to be in NP that may or may not be in P. Many of these problems are extremely common and seemingly difficult problems, for which it would be very important to find a deterministic polynomial algorithm. Cook [1971] discovered the remarkable fact that there were some problems L, known as NP-*hard* problems, with the following property: If L can be solved by a deterministic polynomial algorithm, then so can every problem in NP. The traveling salesman problem discussed in Example 2.10 is such an NP-hard problem. Indeed, it is an NP-*complete* problem, an NP-hard problem that belongs to the class NP. Karp [1972] showed that there were a great many NP-complete problems. Now many people doubt that every problem for which there is a nondeterministic polynomial algorithm also will have a deterministic polynomial algorithm. Hence, they doubt whether it will ever be possible to find deterministic polynomial algorithms for such NP-hard (NP-complete) problems as the traveling salesman problem. Thus, NP-hard (NP-complete) problems are hard in a very real sense. See Garey and Johnson [1979] for a comprehensive discussion of NP-completeness. See also Reingold, Nievergelt, and Deo [1977].

Since real-world problems have to be solved, we cannot simply stop seeking a solution when we find that a problem is NP-complete or NP-hard. We make compromises, for instance by dealing with special cases of the problem that might not be NP-hard. For example, we could consider the traveling salesman problem only when the two cheapest links are available when leaving any city; or when, upon leaving a city, only the five closest cities are considered. Alternatively, we seek good algorithms that approximate the solution to the problem with which we are dealing. An increasingly important activity in present-day combinatorics is to find good algorithms that come close to the (optimal) solution to a problem.

EXERCISES FOR SECTION 2.18

1. In each of the following cases, determine if f is $O(g)$ and justify your answer *from the definition.*

   (a) $f = 2^n$, $g = 5^n$
   (b) $f = 6n + 2/n$, $g = n^2$
   (c) $f = 10n$, $g = n^2$
   (d) $f = 4^{2n}$, $g = 4^{5n} - 50$
   (e) $f = \frac{1}{5}n^4$, $g = n^2 + 7$
   (f) $f = \cos n$, $g = 4$

2. Use limits to determine if f is $O(g)$.

(a) $f = 3^n$, $g = 4^n$ (b) $f = n + 2/n$, $g = n^3$
(c) $f = 10n$, $g = n^2$ (d) $f = 3^{2n}$, $g = 3^5 n - 100$
(e) $f = n!$, $g = e^n$ (*Hint:* Recall Stirling's approximation from Section 2.3.)

3. Prove that if f is $O(h)$ and g is $O(h)$, then $f + g$ is $O(h)$.
4. Prove that if f is $O(g)$ and g is $O(h)$, then f is $O(h)$.
5. Prove that if f is $O(g)$ and g is $O(f)$, then for all h, h is $O(f)$ if and only if h is $O(g)$.
6. (a) Show that $\log_2 n$ is $O(n)$. (b) Show that n is not $O(\log_2 n)$.
7. Suppose that $c > 1$.

   (a) Show that c^n is $O(n!)$. (b) Show that $n!$ is not $O(c^n)$.
8. (a) Is it true that n is $O(n \log_2 n)$? Why?

   (b) Is it true that $n \log_2 n$ is $O(n)$? Why?
9. For each of the following functions f and g, determine if f is $O(g)$ and justify your answer. You may use the definition, limits, a theorem from the text, or the result of a previous exercise.

   (a) $f = 5n$, $g = n^3$ (b) $f = n^3$, $g = 5n$
   (c) $f = 6n^2 + 2n + 1$, $g = n^2$ (d) $f = n^3$, $g = 2^n$
   (e) $f = n^3$, $g = n^5$ (f) $f = 7n^7$, $g = 3^n$
   (g) $f = 10 \log_2 n$, $g = n$ (h) $f = n^2 + 2^n$, $g = 4^n$
   (i) $f = \log_2 n + 25$, $g = n^3$ (j) $f = n^2$, $g = 2^n + n$
   (k) $f = 7^n$, $g = 4n^{50} + 25n^{10} + 100n$ (l) $f = 3n^4 + 2n^2 + 1$, $g = 2^n$
   (m) $f = n + n \log_2 n$, $g = 2^n$ (n) $f = n^3 + 3^n$, $g = n!$
10. Explain what functions are $O(1)$, that is, $O(g)$ for $g \equiv 1$.
11. Let $f(n) = \sum_{i=1}^{n} i$. Show that f is $O(n^2)$.
12. Which of the following functions are "big oh" of the others: $4n \log_2 n + \log_2 n$, $n^2 + 3/n^2$, $\log_2 \log_2 n$.
13. Repeat Exercise 12 for $(\log_2 n)^2$, 2^n, and $4n^2 \log_2 n + 5n$.

2.19 PIGEONHOLE PRINCIPLE AND ITS GENERALIZATIONS

2.19.1 The Simplest Version of the Pigeonhole Principle

In combinatorics, one of the most widely used tools for proving that a certain kind of arrangement or pattern *exists* is the *pigeonhole principle*. Stated informally, this principle says the following: If there are "many" pigeons and "few" pigeonholes, then there must be two or more pigeons occupying the same pigeonhole. This principle is also called the *Dirichlet drawer principle*,[35] the *shoebox principle*, and by

[35] Although the origin of the pigeonhole principle is not clear, it was widely used by the nineteenth-century mathematician Peter Dirichlet.

other names. It says that if there are many objects (shoes) and few drawers (shoeboxes), then some drawer (shoebox) must have two or more objects (shoes). We present several variants of this basic combinatorial principle and several applications of it. Note that the pigeonhole principle simply states that there must *exist* two or more pigeons occupying the same pigeonhole. It does not help us to identify such pigeons.

Let us start by stating the pigeonhole principle more precisely.

Theorem 2.14 (Pigeonhole Principle) If $k+1$ pigeons are placed into k pigeonholes, then at least one pigeonhole will contain two or more pigeons.

To illustrate Theorem 2.14, we note that if there are 13 people in a room, at least two of them are sure to have a birthday in the same month. Similarly, if there are 677 people chosen from the telephone book, then there will be at least two whose first and last names begin with the same letter. The next two examples are somewhat deeper.

Example 2.33 Scheduling Meetings of Legislative Committees (Example 1.4 Revisited) Consider the meeting schedule problem of Example 1.4. A *clique* consists of a set of committees each pair of which have a member in common. The *clique number* corresponding to the set of committees is the size of the largest clique. Given the data of Table 1.5, the largest clique has size 3. The cliques of size 3 correspond to the triangles in the graph of Figure 1.1. Since all committees in a clique must receive different meeting times, the pigeonhole principle says that the number of meeting times required is at least as large as the size of the largest clique. To see why, let the vertices of a clique be the pigeons and the meeting times be pigeonholes. (In the language of Chapter 3, this conclusion says that the chromatic number of a graph is always at least as big as the clique number.) ■

Example 2.34 Manufacturing Personal Computers A manufacturer of personal computers (PCs) makes at least one PC every day over a period of 30 days, doesn't start a new PC on a day when it is impossible to finish it, and averages no more than $1\frac{1}{2}$ PCs per day. Then there must be a period of consecutive days during which *exactly* 14 PCs are made. To see why, let a_i be the number of PCs made through the end of the ith day. Since at least one PC is made each day, and at most 45 PCs in 30 days, we have

$$\begin{aligned} a_1 &< a_2 < \cdots < a_{30}, \\ a_1 &\geq 1, \\ a_{30} &\leq 45. \end{aligned}$$

Also,

$$a_1 + 14 < a_2 + 14 < \cdots < a_{30} + 14 \leq 45 + 14 = 59.$$

Now consider the following numbers:

$$a_1, a_2, \ldots, a_{30}, a_1 + 14, a_2 + 14, \ldots, a_{30} + 14.$$

These are 60 numbers, each between 1 and 59. By the pigeonhole principle, two of these numbers are equal. Since $a_1, a_2, \ldots, a_{30}$ are all different and $a_1 + 14, a_2 + 14, \ldots, a_{30} + 14$ are all different, there exist i and j $(i \neq j)$ so that $a_i = a_j + 14$. Thus, between days i and j, the manufacturer makes exactly 14 PCs. ■

2.19.2 Generalizations and Applications of the Pigeonhole Principle

To begin with, let us state some stronger versions of the pigeonhole principle. In particular, if $2k + 1$ pigeons are placed into k pigeonholes, at least one pigeonhole will contain more than two pigeons. This result follows since if every pigeonhole contains at most two pigeons, then since there are k pigeonholes, there would be at most $2k$ pigeons in all. By the same line of reasoning, if $3k + 1$ pigeons are placed into k pigeonholes, then at least one pigeonhole will contain more than three pigeons.

Speaking generally, we have the following theorem.

Theorem 2.15 If m pigeons are placed into k pigeonholes, then at least one pigeonhole will contain more than

$$\left\lfloor \frac{m-1}{k} \right\rfloor$$

pigeons.

Proof. If the largest number of pigeons in a pigeonhole is at most $\lfloor (m-1)/k \rfloor$, the total number of pigeons is at most

$$k \left\lfloor \frac{m-1}{k} \right\rfloor \leq m - 1 < m.$$

Q.E.D.

To illustrate Theorem 2.15, note that if there are 40 people in a room, a group of more than three will have a common birth month, for $\lfloor 39/12 \rfloor = 3$. To give another application of this result, suppose we know that a computer's memory has a capacity of 8000 bits in eight storage locations. Then we know that we have room in at least one location for at least 1000 bits. For $m = 8000$ and $k = 8$, so $\lfloor (m-1)/k \rfloor = 999$. Similarly, if a factory has 40 electrical outlets with a total 9600-volt capacity, we know there must be at least one outlet with a capacity of 240 or more volts.

These last two examples illustrate the following corollary of Theorem 2.15, whose formal proof is left to the reader (Exercise 21).

Corollary 2.15.1 The average value of a set of numbers is between the smallest and the largest of the numbers.

An alternative way to state this corollary is:

Corollary 2.15.2 Given a set of numbers, there is always a number in the set whose value is at least as large (at least as small) as the average value of the numbers in the set.

Example 2.35 Scheduling Meetings of Legislative Committees (Example 2.33 Revisited) To continue with Example 2.33, let us say that a set of committees is an *independent set* if no pair of them have a member in common. An assignment of meeting times partitions the n committees into k disjoint groups each of which is an independent set. The average size of such an independent set is n/k. Thus, there is at least one independent set of size at least n/k. In the data of Table 1.5, $n = 6$, $k = 3$, and $n/k = 2$. If there are 8 committees and 3 meeting times, we know that at least one meeting time will have at least 8/3, i.e., at least 3, committees. ■

Example 2.36 Web Servers A company has 15 web servers and 10 Internet ports. We never expect more than 10 web servers to need a port at any one time. Every 5 minutes, some subset of the web servers requests ports. We wish to connect each web server to some of the ports, in such a way that we use as few connections as possible, but we are always sure that a web server will have a port to access. (A port can be used by at most one web server at a time.) How many connections are needed? To answer this question, note that if there are fewer than 60 connections, the average port will have fewer than six connections, so by Corollary 2.15.2, there will be some port that will be connected to five or fewer web servers. If the remaining 10 web servers were used at one time, there would be only 9 ports left for them. Thus, at least 60 connections are required. It is left to the reader (Exercise 19) to show that there is an arrangement with 60 connections that has the desired properties. ■

Another application of Theorem 2.15 is a result about increasing and decreasing subsequences of a sequence of numbers. Consider the sequence of numbers $x_1, x_2, \ldots, x_p$. A *subsequence* is any sequence $x_{i_1}, x_{i_2}, \ldots, x_{i_q}$ such that $1 \leq i_1 < i_2 < \cdots < i_q \leq p$. For instance, if $x_1 = 9, x_2 = 6, x_3 = 14, x_4 = 8$, and $x_5 = 17$, then we have the sequence 9, 6, 14, 8, 17; the subsequence x_2, x_4, x_5 is the sequence 6, 8, 17; the subsequence x_1, x_3, x_4, x_5 is the sequence 9, 14, 8, 17; and so on. A subsequence is *increasing* if its entries go successively up in value, and *decreasing* if its entries go successively down in value. In our example, a longest increasing subsequence is 9, 14, 17, and a longest decreasing subsequence is 14, 8.

To give another example, consider the sequence

$$12, 5, 4, 3, 8, 7, 6, 11, 10, 9.$$

A longest increasing subsequence is 5, 8, 11 and a longest decreasing subsequence is 12, 11, 10, 9. These two examples illustrate the following theorem, whose proof depends on Theorem 2.15.

Theorem 2.16 (Erdős and Szekeres [1935]) Given a sequence of n^2+1 *distinct* integers, either there is an increasing subsequence of $n+1$ terms or a decreasing subsequence of $n + 1$ terms.

Note that n^2+1 is required for this theorem; that is, the conclusion can fail for a sequence of fewer than n^2+1 integers. For example, consider the sequence

$$3, 2, 1, 6, 5, 4, 9, 8, 7.$$

This is a sequence of 9 integers arranged so that the longest increasing subsequences and the longest decreasing subsequences are 3 terms long.

Proof of Theorem 2.16. Let the sequence be

$$x_1, x_2, \ldots, x_{n^2+1}.$$

Let t_i be the number of terms in the longest increasing subsequence beginning at x_i. If any t_i is at least $n+1$, the theorem is proved. Thus, assume that each t_i is between 1 and n. We therefore have n^2+1 pigeons (the n^2+1 t_i's) to be placed into n pigeonholes (the numbers $1, 2, \ldots, n$). By Theorem 2.15, there is a pigeonhole containing at least

$$\left\lfloor \frac{(n^2+1)-1}{n} \right\rfloor + 1 = n+1$$

pigeons. That is, there are at least $n+1$ t_i's that are equal. We shall show that the x_i's associated with these t_i's form a decreasing subsequence. For suppose that $t_i = t_j$, with $i < j$. We shall show that $x_i > x_j$. If $x_i \leq x_j$, then $x_i < x_j$ because of the hypothesis that the n^2+1 integers are all distinct. Then x_i followed by the longest increasing subsequence beginning at x_j forms an increasing subsequence of length $t_j + 1$. Thus, $t_i \geq t_j + 1$, which is a contradiction. Q.E.D.

To illustrate this proof, let us consider the following sequence of 10 distinct integers:

$$10, 3, 2, 1, 6, 5, 4, 9, 8, 7.$$

Here $n = 3$ since $10 = 3^2 + 1$, and we have

i	x_i	t_i	Sample subsequence
1	10	1	10
2	3	3	3, 6, 7
3	2	3	2, 6, 7
4	1	3	1, 6, 7
5	6	2	6, 7

i	x_i	t_i	Sample subsequence
6	5	2	5, 7
7	4	2	4, 7
8	9	1	9
9	8	1	8
10	7	1	7

Hence, there are four 1's among the t_i's, and the corresponding x_i's, namely x_1, x_8, x_9, x_{10}, form a decreasing subsequence, 10, 9, 8, 7.

We close this section by stating one more generalization of the pigeonhole principle, whose proof we leave to the reader (Exercise 22).

Theorem 2.17 Suppose that $p_1, p_2, \ldots, p_k$ are positive integers. If

$$p_1 + p_2 + \cdots + p_k - k + 1$$

pigeons are put into k pigeonholes, then either the first pigeonhole contains at least p_1 pigeons, or the second pigeonhole contains at least p_2 pigeons, or $\ldots$, or the kth pigeonhole contains at least p_k pigeons.

2.19.3 Ramsey Numbers

One simple application of the version of the pigeonhole principle stated in Theorem 2.15 is the following.

Theorem 2.18 Assume that among 6 persons, each pair of persons are either friends or enemies. Then either there are 3 persons who are mutual friends or 3 persons who are mutual enemies.

Proof. Let a be any person. By the pigeonhole principle, of the remaining 5 people, either 3 or more are friends of a or 3 or more are enemies of a. (Take $m = 5$ and $k = 2$ in Theorem 2.15.) Suppose first that b, c, and d are friends of a. If any 2 of these persons are friends, these 2 and a form a group of 3 mutual friends. If none of b, c, and d are friends, then b, c, and d form a group of 3 mutual enemies. The argument is similar if we suppose that b, c, and d are enemies of a. Q.E.D.

Theorem 2.18 is the simplest result in the combinatorial subject known as Ramsey theory, dating back to the original article by Ramsey [1930]. It can be restated as follows:

Theorem 2.19 Suppose that S is any set of 6 elements. If we divide the 2-element subsets of S into two classes, X and Y, then either

1. there is a 3-element subset of S all of whose 2-element subsets are in X,

or

2. there is a 3-element subset of S all of whose 2-element subsets are in Y.

Generalizing these conclusions, suppose that p and q are integers with $p,\ q\ \geq 2$. We say that a positive integer N has the (p, q) *Ramsey property* if the following holds: Given any set S of N elements, if we divide the 2-element subsets of S into two classes X and Y, then either

1. there is a p-element subset of S all of whose 2-element subsets are in X,

or

2. there is a q-element subset of S all of whose 2-element subsets are in Y.

For instance, by Theorem 2.19, the number 6 has the $(3, 3)$ Ramsey property. However, the number 3 does not have the $(3, 3)$ Ramsey property. For consider the set $S = \{a, b, c\}$ and the division of 2-element subsets of S into $X = \{\{a, b\}, \{b, c\}\}$ and $Y = \{\{a, c\}\}$. Then clearly there is no 3-element subset of S all of whose 2-element subsets are in X or 3-element subset of S all of whose 2-element subsets are in Y. Similarly, the numbers 4 and 5 do not have the $(3, 3)$ Ramsey property.

Note that if the number N has the (p, q) Ramsey property and $M > N$, the number M has the (p, q) Ramsey property. (Why?) The main theorem of Ramsey theory states that the Ramsey property is well defined.

Table 2.12: The Known Ramsey Numbers $R(p, q)^a$

p	q	$R(p,q)$	Reference(s)
2	n	n	
3	3	6	Greenwood and Gleason [1955]
3	4	9	Greenwood and Gleason [1955]
3	5	14	Greenwood and Gleason [1955]
3	6	18	Kéry [1964]
3	7	23	Kalbfleisch [1966], Graver and Yackel [1968]
3	8	28	Grinstead and Roberts [1982], McKay and Min [1992]
3	9	36	Kalbfleisch [1966], Grinstead and Roberts [1982]
4	4	18	Greenwood and Gleason [1955]
4	5	25	Kalbfleisch [1965], McKay and Radziszowski [1995]

[a]Note that $R(p, q) = R(q, p)$.

Theorem 2.20 (Ramsey's Theorem[36]) If p and q are integers with $p, q \geq 2$, there is a positive integer N which has the (p, q) Ramsey property.

For a proof, see Graham, Rothschild, and Spencer [1990].

One of the key problems in the subject known as Ramsey theory is the identification of the *Ramsey number*, $R(p, q)$, which is the smallest number that has the (p, q) Ramsey property. Note that by Theorem 2.19, $R(3, 3) \leq 6$. In fact, $R(3, 3) = 6$. The problem of computing a Ramsey number, $R(p, q)$, is an example of an optimization problem. So, in trying to compute Ramsey numbers, we are working on the third basic type of combinatorics problem, the optimization problem. Computation of Ramsey numbers is in general a difficult problem. Very few Ramsey numbers are known explicitly. Indeed, the only known Ramsey numbers $R(p, q)$ are given in Table 2.12. (Some of these entries are verified, at least in part, in Section 3.8.) For a comprehensive survey article on Ramsey numbers, the reader is referred to Radziszowski [2002]. See also Graham [1981], Graham, Rothschild, and Spencer [1990], and Chung and Grinstead [1983].

Ramsey's Theorem (Theorem 2.20) has various generalizations. Some are discussed in Section 3.8. For others, see Graham, Rothschild, and Spencer [1990].

Ramsey theory has some intriguing applications to topics such as confusion graphs for noisy channels, design of packet-switched networks, information retrieval, and decisionmaking. A decisionmaking application will be discussed in Section 4.3.3. An overview of some applications of Ramsey theory can be found in Roberts [1984].

[36]This theorem is essentially contained in the original paper by Ramsey [1930], which was mainly concerned with applications to formal logic. The basic results were rediscovered and popularized by Erdős and Szekeres [1935]. See Graham, Rothschild, and Spencer [1990] for an account.

EXERCISES FOR SECTION 2.19

1. How many people must be chosen to be sure that at least two have:
   (a) The same first initial?
   (b) A birthday on the same day of the year?
   (c) The same last four digits in their social security numbers?
   (d) The same first three digits in their telephone numbers?
2. Repeat Exercise 1 if we ask for at least three people to have the desired property.
3. If five different pairs of socks are put unsorted into a drawer, how many individual socks must be chosen before we can be sure of finding a pair?
4. (a) How many three-digit bit strings must we choose to be sure that two of them agree on at least one digit?
   (b) How many n-digit bit strings must we choose?
5. Final exam times are assigned to 301 courses so that two courses with a common student get different exam times and 20 exam times suffice. What can you say about the largest number of courses that can be scheduled at any one time?
6. If a rental car company has 95 cars with a total of 465 seats, can we be sure that there is a car with at least 5 seats?
7. If a school has 400 courses with an average of 40 students per course, what conclusion can you draw about the largest course?
8. If a telephone switching network of 20 switching stations averages 65,000 connections for each station, what can you say about the number of connections in the smallest station?
9. There are 3 slices of olive pizza, 5 slices of plain pizza, 7 slices of pepperoni pizza, and 8 slices of anchovy pizza remaining at a pizza party.
   (a) How many slices need to be requested to assure that 3 of at least one kind of pizza are received?
   (b) How many slices need to be requested to assure that 5 slices of anchovy are received?
10. A building inspector has 77 days to make his rounds. He wants to make at least one inspection a day, and has 132 inspections to make. Is there a period of consecutive days in which he makes exactly 21 inspections? Why?
11. A researcher wants to run at least one trial a day over a period of 50 days, but no more than 75 trials in all.
    (a) Show that during those 50 days, there is a period of consecutive days during which the researcher runs exactly 24 trials.
    (b) Is the conclusion still true for 30 trials?
12. Give an example of a committee scheduling problem where the size of the largest clique is smaller than the number of meeting times required.
13. Find the longest increasing and longest decreasing subsequences of each of the following sequences and check that your conclusions verify the Erdős-Szekeres Theorem.

(a) 6,5,7,4,1 (b) 6,5,7,4,1,10,9,11,14,3 (c) 4,12,3,7,14,13,15,16,10,8

14. Give an example of a sequence of 16 distinct integers that has neither an increasing nor a decreasing subsequence of 5 terms.
15. An employee's time clock shows that she worked 81 hours over a period of 10 days. Show that on some pair of consecutive days, the employee worked at least 17 hours.
16. A modem connection is used for 300 hours over a period of 15 days. Show that on some period of 3 consecutive days, the modem was used at least 60 hours.
17. There are 25 workers in a corporation sharing 12 cutting machines. Every hour, some group of the workers needs a cutting machine. We never expect more than 12 workers to require a machine at any given time. We assign to each machine a list of the workers cleared to use it, and make sure that each worker is on at least one machine's list. If the number of names on each of the lists is added up, the total is 95. Show that it is possible that at some hour some worker might not be able to find a machine to use.
18. Consider the following sequence:

$$9, 8, 4, 3, 2, 7, 6, 5, 10, 1.$$

Find the numbers t_i as defined in the proof of Theorem 2.16, and use these t_i's to find a decreasing subsequence of at least four terms.

19. In Example 2.36, show that there is an arrangement with 60 connections that has the properties desired.
20. Suppose that there are 10 people at a party whose (integer) ages range from 0 to 100.

    (a) Show that there are two distinct, but not necessarily disjoint, subsets of people that have exactly the same total age.

    (b) Using the two subsets from part (a), show that there exist two *disjoint* subsets that have exactly the same total age.

21. Prove Corollary 2.15.1 from Theorem 2.15.
22. Prove Theorem 2.17.
23. Show that if $n+1$ numbers are selected from the set $1, 2, 3, \ldots, 2n$, one of these will divide a second one of them.
24. Prove that in a group of at least 2 people, there are always 2 people who have the same number of acquaintances in the group.
25. An interviewer wants to assign each job applicant interviewed a rating of P (pass) or F (fail). She finds that no matter how she assigns the ratings, at least 3 people receive the same rating. What is the least number of applicants she could have interviewed?
26. Repeat Exercise 25 if she always finds 4 people who receive the same rating.
27. Given a sequence of p integers $a_1, a_2, \ldots, a_p$, show that there exist consecutive terms in the sequence whose sum is divisible by p. That is, show that there are i and j, with $1 \le i \le j \le p$, such that $a_i + a_{i+1} + \cdots + a_j$ is divisible by p.

28. Show that given a sequence of $R(n+1, n+1)$ distinct integers, either there is an increasing subsequence of $n+1$ terms or a decreasing subsequence of $n+1$ terms.

29. Show by exhibiting a division X and Y that:

    (a) The number 4 does not have the $(3,3)$ Ramsey property

    (b) The number 5 does not have the $(3,4)$ Ramsey property

    (c) The number 6 does not have the $(4,4)$ Ramsey property

30. Find the following Ramsey numbers.

    (a) $R(2,2)$ (b) $R(2,8)$ (c) $R(7,2)$

31. Show that if the number N has the (p,q) Ramsey property and $M > N$, the number M has the (p,q) Ramsey property.

32. Consider a group of 10 people, each pair of which are either friends or enemies.

    (a) Show that if some person in the group has at least 4 friends, there are 3 people who are mutual friends or 4 people who are mutual enemies.

    (b) Similarly, if some person in the group has at least 6 enemies, show that either there are 3 people who are mutual friends or 4 people who are mutual enemies.

    (c) Show that by parts (a) and (b), a group of 10 people, each pair of which are either friends or enemies, has either 3 people who are mutual friends or 4 people who are mutual enemies.

    (d) Does part (c) tell you anything about a Ramsey number?

33. Suppose that p, q, and r are integers with $p \geq r$, $q \geq r$, and $r \geq 1$. A positive integer N has the $(p,q;r)$ *Ramsey property* if the following holds: Given any set S of N elements, if we divide the r-element subsets of S into two classes X and Y, then either:

    1. There is a p-element subset of S all of whose r-element subsets are in X,

    or

    2. There is a q-element subset of S all of whose r-element subsets are in Y.

    The *Ramsey number* $R(p,q;r)$ is defined to be the smallest integer N with the $(p,q;r)$ Ramsey property.[37] For a proof that $R(p,q;r)$ is well defined, i.e., that there is always such an N, see Graham, Rothschild, and Spencer [1990].

    (a) Show that $R(p,q;1) = p+q-1$.

    (b) Show that $R(p,r;r) = p$ and $R(r,q;r) = q$.

[37] The (p,q) Ramsey property is the same as the $(p,q;2)$ Ramsey property.

ADDITIONAL EXERCISES FOR CHAPTER 2

1. There are 1000 applicants for admission to a college that plans to admit 300. How many possible ways are there for the college to choose the 300 applicants admitted?
2. An *octapeptide* is a chain of 8 amino acids, each of which is one of 20 naturally occurring amino acids. How many octapeptides are there?
3. In an RNA chain of 15 bases, there are 3 A's, 6 U's, 5 G's, and 1 C. If the chain begins with GU and ends with ACU, how many such chains are there?
4. How many functions are there each of which assigns a number 0 or a number 1 to each $m \times n$ matrix of 0's and 1's?
5. How many switching functions of 5 variables either assign 1 to all bit strings that start with a 1 or assign 0 to all bit strings that start with a 1?
6. In scheduling appliance repairs, eight homes are assigned the morning and nine the afternoon. In how many different orders can we schedule repairs?
7. If campus telephone extensions have 4 digits with no repetitions, how many different extensions are possible?
8. In an RNA chain of 20 bases, there are 4 A's, 5 U's, 6 G's, and 5 C's. If the chain begins either AC or UG, how many such chains are there?
9. How many distinguishable permutations are there of the letters in the word *reconnaissance*?
10. A chain of 20 amino acids has 5 histidines, 6 arginines, 4 glycines, 1 aspargine, 3 lysines, and 1 glutamic acid. How many such chains are there?
11. A system of 10 components works if at least 4 of the first 5 components work and at least 4 of the second 5 components work. In how many ways can the system work?
12. Of 15 paint jobs to be done in a day, 5 of them are short, 4 are long, and 6 are of intermediate length. If the 15 jobs are all distinguishable, in how many different orders can they be run so that:

    (a) All the short jobs are run at the beginning?

    (b) All the jobs of the same length are run consecutively?

13. A person wishes to visit 6 cities, each exactly twice, and never visiting the same city twice in a row. In how many ways can this be done?
14. A family with 9 children has 2 children with black hair, 3 with brown hair, 1 with red hair, and 3 with blond hair. How many different birth orders can give rise to such a family?
15. An ice cream parlor offers 29 different flavors. How many different triple cones are possible if each scoop on the cone has to be a different flavor?
16. A man has 6 different suits. In how many ways can he choose a jacket and a pair of pants that do not match?
17. Suppose that of 11 houses on a block, 6 have termites.

    (a) In how many ways can the presence or absence of termites occur so that the houses with termites are all next to each other?

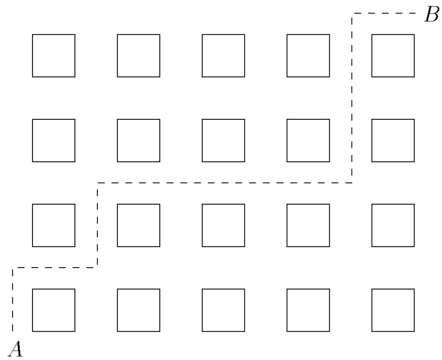

Figure 2.7: A grid of city streets. The dashed line gives a typical route.

(b) In how many ways can the presence or absence of termites occur so that none of the houses with termites are next to each other?

18. How many ways are there to distribute 8 gifts to 5 children?

19. How many ways are there to run an experiment in which each of 5 subjects is given some pills to take if there are 20 pills and each subject must take at least 1?

20. How many ways are there to prepare 20 recipes in a 7-day period if there is a list of 75 recipes to choose from, if the order in which the recipes are made does not matter, and if duplicate recipes are acceptable?

21. If a person owns 5 mutual funds holding a total of 56 stocks, what can you say about the number of stocks in the largest fund?

22. A web site was "hit" 300 times over a period of 15 days. Show that over some period of 3 consecutive days, it was "hit" at least 60 times.

23. How many steps does it take to change the permutation 84316275 into the identity permutation if:

    (a) We only allow interchanges between the ith and $(i+1)$st element?

    (b) We only allow interchanges that replace a subsequence (like 3162) by its reverse (2613)?

24. (Pólya) Consider a city with a grid of streets as shown in Figure 2.7. In an experiment, each subject starts at corner A and is told to proceed to corner B, which is five blocks east and four blocks north of A. He or she is given a route that takes exactly nine blocks to walk. The experimenter wants to use 100 subjects and give each one a different route to walk. Is it possible to perform this experiment? Why?

25. (Liu [1968]) We print one 5-digit number on a slip of paper. We include numbers beginning with 0's, for example, 00158. Since the digits 0,1, and 8 look the same upside down, and since 6 and 9 are interchanged when a slip of paper is turned upside down, 5-digit numbers such as 61891 and 16819 can share the same slip of paper. If we want to include all possible 5-digit numbers but allow this kind of sharing, how many different slips of paper do we need?

REFERENCES FOR CHAPTER 2

Applegate, D., Bixby, R., Chvátal, V., and Cook, W., "On the Solution of Traveling Salesman Problems," *Doc. Math., Extra Vol. III*, Proceedings of the ICM (1998), 645–656.

Aumann, R. J., and Hart, S. (eds.), *Handbook of Game Theory with Economic Applications*, Elsevier Science, New York, 1998.

Bafna, V., and Pevzner, P. A., "Genome Rearrangements and Sorting by Reversal," *SIAM J. Comput., 25* (1996), 272–289.

Bafna, V., and Pevzner, P. A., "Sorting by Transpositions," *SIAM J. Discrete Math., 11* (1998), 224–240.

Ball, M. O., Colbourn, C. J., and Provan, J. S., "Network Reliability," in M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser (eds.), *Handbooks in Operations Research and Management Science, 7*, North-Holland Publishing Co., Amsterdam, 1995, 673–762.

Banzhaf, J. F., III, "Weighted Voting Doesn't Work: A Mathematical Analysis," *Rutgers Law Rev., 19* (1965), 317–343.

Barlow, R. E., and Proschan, F., *Statistical Theory of Reliability and Life Testing: Probability Models*, Holt, Rinehart and Winston, New York, 1975.

Berge, C., *Principles of Combinatorics*, Academic Press, New York, 1971.

Bland, R. G., and Shallcross, D. F., "Large Traveling Salesman Problems Arising from Experiments in X-Ray Crystallography: A Preliminary Report on Computation," *Oper. Res. Letters, 8* (1989), 125–128.

Boros, E., Gurvich, V., Hammer, P. L., Ibaraki, T., and Kogan, A., "Decomposability of Partially Defined Boolean Functions," *Discrete Appl. Math., 62* (1995), 51–75.

Boros, E., Ibaraki, T., and Makino, K., "Error-free and Best-Fit Extensions of Partially Defined Boolean Functions," *Inform. Comput., 140* (1998), 254–283.

Boros, E., Hammer, P. L., Ibaraki, T., Kogan, A., Mayoraz, E., and Muchnik, I., "An Implementation of Logical Analysis of Data," *IEEE Trans. Knowl. Data Eng., 12* (2000), 292–306.

Boyce, D. M., and Cross, M. J., "An Algorithm for the Shapley-Shubik Voting Power Index for Weighted Voting," unpublished Bell Telephone Laboratories manuscript, 1973.

Brams, S. J., *Voting Procedures*, North-Holland, Amsterdam, 1994.

Brams, S. J., and Fishburn, P. C., *Approval Voting*, Birkhauser, Boston, 1983.

Brams, S. J., Lucas, W. F., and Straffin, P. D. (eds.), *Political and Related Models*, Vol. 2 of *Modules in Applied Mathematics*, Springer-Verlag, New York, 1983.

Brams, S. J., Schotter, A., and Schwödianer, G. (eds.), *Applied Game Theory*, IHS-Studies No. 1, Physica-Verlag, Würzburg, 1979.

Buck, R. C., *Advanced Calculus*, 2nd ed., McGraw-Hill, New York, 1965.

Caprara, A., "Sorting by Reversals Is Difficult," *Proc. of RECOMB 97: The First International Conf. on Comp. Biology*, ACM Press, New York, 1997, 75–83.

Christie, D. A., "Sorting Permutations by Block-Interchanges," *Inform. Process. Lett., 60* (1996), 165–169.

Chung, F. R. K., and Grinstead, C. M., "A Survey of Bounds for Classical Ramsey Numbers," *J. Graph Theory, 7* (1983), 25–37.

Clote, P., and Backofen, R., *Computational Molecular Biology: An Introduction*, Wiley, New York, 2000.

Cohen, D. I. A., *Basic Techniques of Combinatorial Theory*, Wiley, New York, 1978.

Coleman, J. S., "Control of Collectivities and the Power of a Collectivity to Act," in B. Lieberman (ed.), *Social Choice*, Gordon and Breach, New York, 1971, 269–298.

Congress of the United States, Office of Technology Assessment Staff, *Mapping Our Genes; Genome Projects - How Big, How Fast?*, Johns Hopkins University Press, Baltimore, MD, 1988.

Cook, S. A., "The Complexity of Theorem-Proving Procedures," *Proceedings of the Third ACM Symposium on Theory of Computing*, Association for Computing Machinery, New York, 1971, 151–158.

Crama, Y., Hammer, P. L., and Ibaraki, T., "Cause-Effect Relationships and Partially Defined Boolean Functions," *Annals of Oper. Res., 16* (1988), 299–325.

Cummings, M. R., *Human Heredity: Principles and Issues*, West Publishing Company, St. Paul, MN, 1997.

Deegan, J., and Packel, E. W., "A New Index of Power for Simple n-Person Games," *International Journal of Game Theory, 7* (1978), 113–123.

Dembski, W. A., *The Design Inference: Eliminating Chance through Small Probabilities*, Cambridge University Press, New York, 1998.

Deo, N., *Graph Theory with Applications to Engineering and Computer Science*, Prentice Hall, Englewood Cliffs, NJ, 1974.

Dobzhansky, T., *Genetics of the Evolutionary Process*, Columbia University Press, New York, 1970.

Edmonds, J., "Paths, Trees, and Flowers," *Canadian J. Mathematics, 17* (1965), 449–467.

Ekin, O., Hammer, P. L., and Kogan, A., "Convexity and Logical Analysis of Data," *Theoret. Comput. Sci., 244* (2000), 95–116.

Elsayed, E. A., "Algorithms for Optimal Material Handling in Automatic Warehousing Systems," *Int. J. Prod. Res., 19* (1981), 525–535.

Elsayed, E. A., and Stern, R. G., "Computerized Algorithms for Order Processing in Automated Warehousing Systems," *Int. J. Prod. Res., 21* (1983), 579–586.

Erdős, P., and Szekeres, G., "A Combinatorial Problem in Geometry," *Composito Math., 2* (1935), 464–470.

Farach-Colton, M., Roberts, F. S., Vingron, M., and Waterman, M. S. (eds.), *Mathematical Support for Molecular Biology*, Vol. 47, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, Providence, RI, 1999.

Farquharson, R., *Theory of Voting*, Yale University Press, New Haven, CT, 1969.

Feigenbaum, J., Papadimitriou, C. H., and Shenker, S., "Sharing the Cost of Multicast Transmissions," to appear in *J. Comput. System Sci.* [See also *Proceedings 32nd ACM Symposium on Theory of Computing*, (2000), 218–227.]

Feller, W., *An Introduction to Probability Theory and Its Applications*, 3rd ed., Wiley, New York, 1968.

Ferretti, V., Nadeau, J., and Sankoff, D., "Original Synteny," *Proc. 7th Symp. Combinatorial Pattern Matching, Springer LNCS 1075* (1996), 149–167.

Fine, T. L., *Theories of Probability*, Academic Press, New York, 1973.

Fudenberg, D., and Tirole, J., *Game Theory*, MIT Press, Cambridge, MA, 1991.

Gamow, G., "Possible Mathematical Relation between Deoxyribonucleic Acid and Proteins," *K. Dan. Vidensk. Selsk. Biol. Medd., 22* (1954), 1–13. (a)

Gamow, G., "Possible Relations between Deoxyribonucleic Acid and Protein Structures," *Nature, 173* (1954), 318. (b)

GAREY, M. R., and JOHNSON, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, 1979.

GOLOMB, S. W., "Efficient Coding for the Deoxyribonucleic Channel," in *Mathematical Problems in the Biological Sciences*, Proceedings of Symposia in Applied Mathematics, Vol. 14, American Mathematical Society, Providence, RI, 1962, 87–100.

GRAHAM, R. L., *Rudiments of Ramsey Theory*, CBMS Regional Conference Series in Mathematics, No. 45, American Mathematical Society, Providence, RI, 1981.

GRAHAM, R. L., ROTHSCHILD, B. L., and SPENCER, J. H., *Ramsey Theory*, 2nd ed., Wiley, New York, 1990.

GRAVER, J. E., and YACKEL, J., "Some Graph Theoretic Results Associated with Ramsey's Theorem," *J. Comb. Theory, 4* (1968), 125–175.

GRAY, F., "Pulse Code Communication," U.S. Patent 2,632,058, March 17, 1953.

GREENWOOD, R. E., and GLEASON, A. M., "Combinatorial Relations and Chromatic Graphs," *Canadian J. Math., 7* (1955), 1–7.

GRIFFITHS, A. J. F., MILLER, J. H., SUZUKI, D. T., LEWONTIN, R. C., and GELBART, W. M., *An Introduction to Genetic Analysis*, 6th ed., Freeman, New York, 1996.

GRINSTEAD, C. M., and ROBERTS, S. M., "On the Ramsey Numbers $R(3,8)$ and $R(3,9)$," *J. Comb. Theory, Series B, 33* (1982), 27–51.

GUSFIELD, D., *Algorithms on Strings, Trees and Sequences; Computer Science and Computational Biology*, Cambridge University Press, New York, 1997.

HANNENHALLI, S., and PEVZNER, P. A., "To Cut or Not to Cut: Applications of Comparative Physical Maps in Molecular Evolution," *Proc. 7th ACM-SIAM Symp. Discrete Algs.* (1996), 304–313.

HARDY, G. H., and WRIGHT, E. M., *An Introduction to the Theory of Numbers*, 5th ed., Oxford University Press, New York, 1980.

HARRISON, M. A., *Introduction to Switching and Automata Theory*, McGraw-Hill, New York, 1965.

HARVARD COMPUTATION LABORATORY STAFF, *Synthesis of Electronic Computing and Control Circuits*, Harvard University Press, Cambridge, MA, 1951.

HEATH, L. S., and VERGARA, J. P. C., "Sorting by Bounded Block-Moves," *Discrete Appl. Math., 88* (1998), 181–206.

HERZOG, S., SHENKER, S., and ESTRIN, D., "Sharing the 'Cost' of Multicast Trees: An Axiomatic Analysis," *IEEE/ACM Trans. Networking, 5* (1997), 847–860.

HILL, F. J., and PETERSON, G. R., *Switching Theory and Logical Design*, Wiley, New York, 1968.

HOLLEY, R. W., EVERETT, G. A., MADISON, J. T., MARQUISEE, M., and ZAMIR, A., "Structure of a Ribonucleic Acid," *Science, 147* (1965), 1462–1465.

HOPCROFT, J. E., "Recent Developments in Random Algorithms," paper presented at SIAM National Meeting, Troy, NY, June 1981.

HUTCHINSON, G., "Evaluation of Polymer Sequence Fragment Data Using Graph Theory," *Bull. Math. Biophys., 31* (1969), 541–562.

HWANG, F. K., MONMA, C., and ROBERTS, F. S. (eds.), *Reliability of Computer and Communication Networks*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 5, American Mathematical Society and Association for Computing Machinery, Providence, RI, 1991.

JAIN, K., and VAZIRANI, V. V., "Applications of Approximation Algorithms to Cooperative Games," *Proceedings 33nd ACM Symposium on Theory of Computing*, (2001), 364–372.

JERRUM, M. R., "The Complexity of Finding Minimum-Length Generator Sequences," *Theoret. Comput. Sci., 36* (1985), 265–289.

Johnston, R. J., "On the Measurement of Power: Some Reactions to Laver," *Environment and Planning A, 10* (1978), 907–914.

Jones, A. J., *Game Theory: Mathematical Models of Conflict*, Wiley, New York, 1980.

Kalbfleisch, J. G., "Construction of Special Edge-Chromatic Graphs," *Canadian Math. Bull., 8* (1965), 575–584.

Kalbfleisch, J. G., "Chromatic Graphs and Ramsey's Theorem," Ph.D. thesis, University of Waterloo, Waterloo, Ontario, Canada, January 1966.

Kaplan, H., Shamir, R., and Tarjan, R. E., "Faster and Simpler Algorithms for Sorting Signed Permutations by Reversal," *Proc. 8th ACM-SIAM Symp. Discrete Algs.* (1997).

Karp, R. M., "Reducibility among Combinatorial Problems," in R. E. Miller and J. W. Thatcher (eds.), *Complexity of Computer Computations*, Plenum Press, New York, 1972, 85–103.

Karp, R. M., and Luby, M. G., "Monte-Carlo Algorithms for Enumeration and Reliability Problems," *24th Annual ACM Symposium on Theory of Computing* (1983), 56–64.

Kececioglu, J. D., and Sankoff, D., "Efficient Bounds for Oriented Chromosome Inversion Distance," *Proc. 5th Symp. Combinatorial Pattern Matching, Springer LNCS 807* (1994), 307–325.

Kececioglu, J. D., and Sankoff, D., "Exact and Approximation Algorithms for Sorting by Reversal," *Algorithmica, 13* (1995), 180–210.

Kelly, J. S., *Social Choice Theory: An Introduction*, Springer-Verlag, New York, 1987.

Kéry, G., "On a Theorem of Ramsey," *Matematikai Lapok, 15* (1964), 204–224.

Kohavi, Z., *Switching and Finite Automata Theory*, McGraw-Hill, New York, 1970.

Lander, E. S., and Waterman, M. S. (eds.), *Calculating the Secrets of Life; Contributions of the Mathematical Sciences to Molecular Biology*, National Academy Press, Washington, DC, 1995.

Lawler, E. L., *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

Lehmer, D. H., "The Machine Tools of Combinatorics," in E. F. Beckenbach (ed.), *Applied Combinatorial Mathematics*, Wiley, New York, 1964, 5–31.

Lewin, B., *Genes VII*, Oxford University Press, New York, 2000.

Liu, C. L., *Introduction to Combinatorial Mathematics*, McGraw-Hill, New York, 1968.

Liu, C. L., *Elements of Discrete Mathematics*, McGraw-Hill, New York, 1977.

Lucas, W. F., "Applications of Cooperative Games to Equitable Allocation," in W. F. Lucas (ed.), *Game Theory and Its Applications*, Proceedings of Symposia in Applied Mathematics, Vol. 24, American Mathematical Society, Providence, RI, 1981, 19–36. (a)

Lucas, W. F. (ed.), *Game Theory and Its Applications*, Proceedings of Symposia in Applied Mathematics, Vol. 24, American Mathematical Society, Providence, RI, 1981. (b)

Lucas, W. F., "Measuring Power in Weighted Voting Systems," in S. J. Brams, W. F. Lucas, and P. D. Straffin (eds.), *Political and Related Models*, Vol. 2 of *Modules in Applied Mathematics*, Springer-Verlag, New York, 1983, 183–238.

Mahadev, N. V. R., and Roberts, F. S., "Consensus List Colorings of Graphs and Physical Mapping of DNA," in M. Janowitz, F.-J. LaPointe, F. R. McMorris, B. Mirkin, and F. S. Roberts (eds.), *Bioconsensus*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 61, American Mathematical Society, Providence, RI, 2003, 83–95.

Mann, I., and Shapley, L. S., "Values of Large Games IV: Evaluating the Electoral College by Monte Carlo Techniques," RAND Corporation Memorandum RM-2651, September 1960; reproduced in M. Shubik (ed.), *Game Theory and Related Approaches to Social Behavior*, Wiley, New York, 1964. (a)

Mann, I., and Shapley, L. S., "Values of Large Games VI: Evaluating the Electoral College Exactly," RAND Corporation Memorandum RM-3158-PR, May 1962; reproduced in part in M. Shubik (ed.), *Game Theory and Related Approaches to Social Behavior*, Wiley, New York, 1964. (b)

McKay, B. D., and Min, Z. K., "The Value of the Ramsey Number $R(3,8)$," *J. Graph Theory, 16* (1992), 99–105.

McKay, B. D., and Radziszowski, S. P., "$R(4,5) = 25$," *J. Graph Theory, 19* (1995), 309–322.

Mosimann, J. E., *Elementary Probability for the Biological Sciences*, Appleton-Century-Crofts, New York, 1968.

Mosimann, J. E., Shapiro, M. B., Merril, C. R., Bradley, D. F., and Vinton, J. E., "Reconstruction of Protein and Nucleic Acid Sequences IV: The Algebra of Free Monoids and the Fragmentation Stratagem," *Bull. Math. Biophys., 28* (1966), 235–260.

Muroga, S., *Logic Design and Switching Theory*, Krieger Publishing Company, Melbourne, 1990.

Myerson, R. B., *Game Theory*, Harvard University Press, Cambridge, MA, 1997.

Niven, I., *An Introduction to the Theory of Numbers*, Wiley, New York, 1991.

O'Brien, S. J. (ed.), *Genetic Maps: Locus Maps of Complex Genomes*, 6th ed., Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, 1993.

Okada, N., Hashimoto, T., and Young, P., "Cost Allocation in Water Resources Development," *Water Resources Research, 18* (1982), 361–373.

Owen, G., *Game Theory*, 3rd ed., Academic Press, San Diego, CA, 1995.

Palmer, J. D., Osorio, B., and Thompson, W. F., "Evolutionary Significance of Inversions in Legume Chloroplast DNAs," *Current Genetics, 14* (1988), 65–74.

Parzen, E., *Modern Probability Theory and Its Applications*, Wiley, New York, 1992.

Pattavina, A., *Switching Theory*, Wiley, New York, 1998.

Pevzner, P. A., *Computational Molecular Biology: An Algorithmic Approach*, MIT Press, Cambridge, MA, 2000.

Prather, R. E., *Discrete Mathematical Structures for Computer Science*, Houghton Mifflin, Boston, 1976.

Radziszowski, S. P., "Small Ramsey Numbers," *Electron. J. Combinat., DS1* (2002), 35 pp.

Ramsey, F. P., "On a Problem of Formal Logic," *Proc. Lond. Math. Soc., 30* (1930), 264–286.

Reingold, E. M., Nievergelt, J., and Deo, N., *Combinatorial Algorithms: Theory and Practice*, Prentice Hall, Englewood Cliffs, NJ, 1977.

Riordan, J., *An Introduction to Combinatorial Analysis*, Princeton University Press, Princeton, NJ, 1980.

Roberts, F. S., *Discrete Mathematical Models, with Applications to Social, Biological, and Environmental Problems*, Prentice Hall, Englewood Cliffs, NJ, 1976.

Roberts, F. S., *Measurement Theory, with Applications to Decisionmaking, Utility, and the Social Sciences*, Addison-Wesley, Reading, MA, 1979. (Digitally printed version, Cambridge University Press, Cambridge, UK, 2009.)

Roberts, F. S., "Applications of Ramsey Theory," *Discrete Appl. Math., 9* (1984), 251–261.

Ross, S. M., *A First Course in Probability*, 5th ed., Prentice Hall, Upper Saddle River, 1997.

Sessions, S. K., "Chromosomes: Molecular Cytogenetics," in D. M. Hillis and C. Moritz (eds.), *Molecular Systematics*, Sinauer, Sunderland, MA, 1990, 156–204.

Setubal, J. C., and Meidanis, J., *Introduction to Computational Molecular Biology*, PWS Publishers, Boston, 1997.

Shapley, L. S., "A Value for n-Person Games," in H. W. Kuhn and A. W. Tucker (eds.), *Contributions to the Theory of Games*, Vol. 2, Annals of Mathematics Studies No. 28, Princeton University Press, Princeton, NJ, 1953, 307–317.

Shapley, L. S., "Measurement of Power in Political Systems," in W. F. Lucas (ed.), *Game Theory and Its Applications*, Proceedings of Symposia in Applied Mathematics, Vol. 24, American Mathematical Society, Providence, RI, 1981, 69–81.

Shapley, L. S., and Shubik, M., "A Method for Evaluating the Distribution of Power in a Committee System," *Amer. Polit. Sci. Rev., 48* (1954), 787–792.

Shapley, L. S., and Shubik, M., "On the Core of an Economic System with Externalities," *Amer. Econ. Rev., 59* (1969), 678–684.

Shubik, M., "Incentives, Decentralized Control, the Assignment of Joint Costs and Internal Pricing," *Management Sci., 8* (1962), 325–343.

Stahl, S., *A Gentle Introduction to Game Theory*, American Mathematical Society, Providence, RI, 1998.

Stanat, D. F., and McAllister, D. F., *Discrete Mathematics in Computer Science*, Prentice Hall, Englewood Cliffs, NJ, 1977.

Stone, H. S., *Discrete Mathematical Structures and Their Applications*, Science Research Associates, Chicago, 1973.

Straffin, P. D., Jr., *Topics in the Theory of Voting*, Birkhäuser-Boston, Cambridge, MA, 1980.

Straffin, P. D., and Heaney, J. P., "Game Theory and the Tennessee Valley Authority," *Internat. J. Game Theory, 10* (1981), 35–43.

Tversky, A., and Gilovich, T., "The Cold Facts about the 'Hot Hands' in Basketball," *Chance, 2* (1989), 16–21.

Waterman, M. S., *Introduction to Computational Biology; Maps, Sequences and Genomes*, CRC Press, Boca Raton, FL, 1995.

Young, H. P., *Cost Allocation: Methods, Principles, Applications*, Elsevier Science, New York, 1986.

Zimmer, C., "And One for the Road," *Discover*, January 1993, 91–92.