CHAPTER 1

Basic Counting

Introduction

Before beginning, we must confront some matters of notation. Two words that we shall often use are *set* and *list*. Both words refer to collections of objects. There is no standard notation for lists. Some of those in use are

apple banana pear peach	*a list of four items ...*
apple, banana, pear, peach	*commas added for clarity ...*
and (apple, banana, pear, peach)	*parentheses added.*

The notation for sets is standard: the items are separated by commas and surround by curly brackets as in

$$\{\text{apple}, \text{banana}, \text{pear}, \text{peach}\}.$$

The curly bracket notation for sets is so well established that you can normally assume it means a set—but beware, Mathematica® uses curly brackets for lists.

What is the difference between a set and a list? Quite a bit, and nothing. "Set" means a collection of distinct objects in which the order doesn't matter. Thus

$$\{\text{apple}, \text{peach}, \text{pear}\} \quad \text{and} \quad \{\text{peach}, \text{apple}, \text{pear}\}$$

are the same sets, and the set {apple, peach, apple} is the same as the set {apple, peach}. In other words, repeated elements are treated as if they occurred only once. Thus two sets are the same if and only if each element that is in one set is in both. In a list, order is important and repeated objects are usually allowed. Thus

$$(\text{apple}, \text{peach}) \quad (\text{peach}, \text{apple}) \quad \text{and} \quad (\text{apple}, \text{peach}, \text{apple})$$

are three different lists. Two lists are the same if and only if they have exactly the same items in exactly the same positions. Thus, sets and lists are different.

On the other hand, people talk about things like "unordered lists," "sets with repetition," and so on. In fact, a set with repetition is so common that it has a name: *multiset*. Two multisets are the same if and only if each item that occurs exactly k times in one of them occurs exactly k times in both. In summary

- *list*: an ordered sequence (repeats allowed),
- *set*: a collection of distinct objects where order does not matter,
- *multiset*: a collection of objects (repeats allowed) where order does not matter.

Thus, an ordered set with repetition allowed is a list and an unordered list of distinct elements is a set. Whenever we refer to a list, we will indicate whether the elements must be distinct. Unless we say otherwise, a list is ordered. An ordered list is sometimes called a *string*, a *sequence* or a *word*. A list is also called a *sample* or a *selection*, especially in probability and statistics. Lists are sometimes called vectors and the elements components.

The terminology "k-list" is frequently used in place of the more cumbersome "k long list." Similarly, we use k-set and k-multiset. Vertical bars (also used for absolute value) are used to denote the number of elements in a set or in a list. For example, if S is an n-set, then $|S| = n$.

We want to know how many ways we can do various things with a set. Here are some examples, which we illustrate by using the set $S = \{x, y, z\}$.

1. How many ways can we *list*, without repetition, all the elements of S? This means, how many ways can we arrange the elements of S in an (ordered) list so that each element of S appears exactly once in each of the lists. For the illustration, there are six ways: xyz, xzy, yxz, yzx, zxy and zyx. (These are all called permutations of S. People often use Greek letters like π and σ to indicate a permutation of a set.)
2. How many ways can we construct a k-list of distinct elements from the set? When $k = |S|$, this is the previous question. If $k = 2$ in the illustration, there are six ways: xy, xz, yx, yz, zx and zy.
3. If the list in the previous question is allowed to contain repetitions, what is the answer? There are nine ways for the illustration: xx, xy, xz, yx, yy, yz, zx, zy and zz.
4. If, in Questions 2 and 3, the order in which the elements appear in the list doesn't matter, what are the answers? For the illustration, the answers are three and six, respectively.
5. How many ways can the set S be partitioned into a collection of k pairwise disjoint nonempty smaller sets? With $k = 2$, the illustration has three such: $\{\{x\}, \{y, z\}\}$, $\{\{x, y\}, \{z\}\}$ and $\{\{x, z\}, \{y\}\}$.

We'll learn how to answer these questions without going through the time-consuming process of constructing (listing) all the items in question as we did for our illustration. Our answer to the last question will be somewhat unsatisfactory. Other answers to it will be discussed in later chapters.

1.1 Lists with Repetitions Allowed

How many ways can we construct a k-list (repeats allowed) using an n-set? Look at our illustration in Question 3 above. The first entry in the list could be x, y or z. After any of these there were three choices (x, y or z) for the second entry. Thus there are $3 \times 3 = 9$ ways to construct such a list. The general pattern should be clear: There are n ways to choose each list entry. Thus

Theorem 1.1 *There are n^k ways to construct a k-list from an n-set.*

This calculation illustrates an important principle:

Theorem 1.2 Rule of Product *Suppose structures are to be constructed by making a sequence of k choices such that, (i) the ith choice can be made in c_i ways, a number independent of what choices were made previously, and (ii) each structure arises in exactly one way in this process. Then, the number of structures is $c_1 \times \cdots \times c_k$.*

"Structures" as used above can be thought of simply as elements of a set. We prefer the term structures because it emphasizes that the elements are built up in some way; in this case, by making a sequence of choices. In the previous calculation, the structures are lists of k things which are built up by adding one thing at a time. Each thing is chosen from a given set of n things and $c_1 = c_2 = \ldots = c_k = n$.

Definition 1.1 Cartesian Product *If $C_1, \ldots, C_k$ are sets, the* **Cartesian product** *of the sets is written $C_1 \times \cdots \times C_k$ and consists of all k-lists $(x_1, \ldots, x_k)$ with $x_i \in C_i$ for $1 \le i \le k$.*

A special case of the Rule of Product is the fact that the number of elements in $C_1 \times \cdots \times C_k$ is the product $|C_1| \cdots |C_k|$. Here C_i is the collection of ith choices and $c_i = |C_i|$. This is only a special case because the Rule of Product would allow the *collection* C_i to depend on the previous choices $x_1, \ldots, x_{i-1}$ as long as the *number* c_i of possible choices does not depend on $x_1, \ldots, x_{i-1}$. The last example in Appendix A gives a proof of this special case of the Rule of Product. In fact, that proof can be altered to give a proof of the general case of the Rule of Product. We will not do so.

Here is a property associated with Cartesian products that we will find useful in our later discussions.

Definition 1.2 Lexicographic Order *If $C_1, \ldots, C_k$ are ordered lists of distinct elements, we may think of them as sets and form the Cartesian product $P = C_1 \times \cdots \times C_k$. The* **lexicographic order** *on P is defined by saying that $a_1 \ldots a_k < b_1 \ldots b_k$ if and only if there is some $t \le k$ such that $a_i = b_i$ for $i < t$ and $a_t < b_t$.*

Often we say *lex order* instead of lexicographic order. If all the C_i's equal $(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)$, then lex order is simply numerical order of k digit integers with leading zeroes allowed. Suppose that all the C_i's equal (<space>, A, B, …, Z). If we throw out those elements of P that have a letter following a space, the result is dictionary order. Unlike these two simple examples, the C_i's usually vary with i.

Example 1.1 A simple count The North-South streets in Rectangle City are named using the numbers 1 through 12 and the East-West streets are named using the letters A through H. Thus, the most southwesterly intersection occurs where First and A streets meet. How many blocks are within the city?

We may think of the city of as consisting of rows of blocks. Each row contains the blocks encountered as we cross the city from East to West. The number of rows is the number of rows of blocks encountered as we cross the city from North to South. This is much like the rows and columns of a matrix. We can apply the Rule of Product: Choose a row and then choose a block in that row. What answer does this give? If you think it is $12 \times 8 = 96$, you're almost correct. Read on.

Each block can be labeled by the streets at its southwesterly corner. These labels have the form (x, y) where x is between 1 and 11 inclusive and y is between A and G. (If you don't see why 12 and H are missing, draw a picture and look at southwesterly corners.) By the Rule of Product there are $11 \times 7 = 77$ blocks. In this case the structures can be taken to be the descriptions of the blocks. Each description has two parts: the names of the north-south and East-West streets at the block's southwest corner. □

Example 1.2 Counting names We now return to the faraway galaxy that was mentioned in Example 2 (p. 1).

The possible positions for the two vowels are $(2, 4)$, $(2, 5)$ and $(3, 5)$. Each of these results in two isolated consonants and two adjacent consonants. Thus the answer is the product of the following factors:

- choose the vowel locations (3 ways);
- choose the vowels (2×2 ways);
- choose the isolated consonants (3×3 ways);
- choose the adjacent consonants (3×2 ways).

The answer is 648. This construction can be interpreted as a Cartesian product as follows. C_1 is the set of lists of possible positions for the vowels, C_2 is the set of lists of vowels in those positions, and C_3 and C_4 are sets of lists of consonants. Thus

$$C_1 = \{(2,4),(2,5),(3,5)\} \qquad C_2 = \{\mathrm{AA},\mathrm{AI},\mathrm{IA},\mathrm{II}\}$$
$$C_3 = \{\mathrm{LL},\mathrm{LS},\mathrm{LT},\mathrm{SL},\mathrm{SS},\mathrm{ST},\mathrm{TL},\mathrm{TS},\mathrm{TT}\} \qquad C_4 = \{\mathrm{LS},\mathrm{LT},\mathrm{SL},\mathrm{ST},\mathrm{TL},\mathrm{TS}\}.$$

For example, ((2,5), IA, SS, ST) in the Cartesian product corresponds to the word SISTAS. □

Here's another important principle, the proof of which is self evident:

Theorem 1.3 Rule of Sum *Suppose a set T of structures can be partitioned into sets $T_1, \ldots, T_j$ so that each structure in T appears in exactly one T_i, then*

$$|T| = |T_1| + \cdots + |T_j|.$$

Example 1.3 Counting names (revisited) We'll redo the previous example using this principle.

The possible vowel (V) and consonant (C) patterns for names are CCVCVC, CVCCVC and CVCVCC. Since these patterns are disjoint and cover all cases, we must compute the number of names of each type and add the results together. For the first pattern we have a product of six factors, one for each choice of a letter: $3 \times 2 \times 2 \times 3 \times 2 \times 3 = 216$. The other two patterns also give 216, for a total of 648 names.

This approach has a wider range of applicability than the method we used in the previous example. We were only able to avoid the Rule of Sum in the first method because each pattern contained the same number of vowels, isolated consonants and adjacent consonants. Here's an example that requires the Rule of Sum. Suppose a name consists of only four letters, namely two vowels and two consonants, constructed so that the vowels are not adjacent and, if the consonants are adjacent, then they are different. There are four patterns: CVCV, VCVC, VCCV. By the Rule of Product, the first two are each associated with 36 names, but VCCV is associated with only 24 names because of the adjacent consonants. Hence, we cannot choose a pattern and then proceed to choose vowels and consonants. On the other hand, we can apply the Rule of Sum to get a total of 96 names. □

Example 1.4 Smorgasbord College committees Smorgasbord College has four departments which have 6, 35, 12 and 7 faculty members. The president wishes to form a faculty judicial committee to hear cases of student misbehavior. To avoid the possibility of ties, the committee will have three members. To avoid favoritism the committee members will be from different departments and the committee will change daily. If the committee only sits during the normal academic year (165 days), how many years can pass before a committee must be repeated?

If T is the set of all possible committees, the answer is $|T|/165$. Let T_i be the set of committees with no members from the ith department. By the Rule of Sum $|T| = |T_1| + |T_2| + |T_3| + |T_4|$. By the Rule of Product

$$|T_1| = 35 \times 12 \times 7 = 2940 \qquad |T_3| = 35 \times 6 \times 7 = 1470$$
$$|T_2| = 6 \times 12 \times 7 = 504 \qquad |T_4| = 35 \times 12 \times 6 = 2520.$$

Thus the number of years is $7434/165 = 45^+$. Due to faculty turnover, a committee need never repeat—if the president's policy lasts that long. □

Using the Rules of Sum and Product

Whenever we encounter a new technique, there are two questions that arise:

- *When* is it used?
- *How* is it used?

For the Rules of Sum and Product, the answers are intertwined:

> **Technique Rules for AND and OR** *Suppose you wish to count the number of structures in a set and that you can describe how to construct the structures in terms of subconstructions that are connected by "ands" and "ors." If this leads to the construction of each structure in a unique way, then the Rules of Sum and Product apply. To use them, replace "ands" by products and "ors" by sums. Whenever you write something like "Do A AND do B," it should mean "Do A AND THEN do B" because the Rule of Product requires that the choices be made sequentially. We will usually omit "then".*

Example 1.5 Applying the technique To see how this technique is applied, let's look back at Example 1.4. A committee consists of either

- One person from Dept. 1 AND one person from Dept. 2 AND one person from Dept. 3, OR
- One person from Dept. 1 AND one person from Dept. 2 AND one person from Dept. 4, OR
- One person from Dept. 1 AND one person from Dept. 3 AND one person from Dept. 4, OR
- One person from Dept. 2 AND one person from Dept. 3 AND one person from Dept. 4.

The number of ways to choose a person from a department equals the number of people in the department. □

Until you become comfortable using the Rules of Sum and Product, look for "and" and "or" in what you do. This is an example of the *divide and conquer* tactic: break the problem into parts and work on each piece separately. Here the first part is getting a phrasing with "ands" and "ors;" the second part is calculating each of the individual pieces; and the third part is applying the Rules of Sum and Product.

Example 1.6 Palindromes A *palindrome* is a list that reads the same from right to left as it does from left to right. For example, ignoring capitalization, punctuation and spaces, "Madam I'm Adam." becomes the palindrome madamimadam.

How many k-long palindromes can be formed from an n-set? The first $\lceil k/2 \rceil$ list elements are arbitrary and the remaining elements are determined.* Thus the answer is $n^{\lceil k/2 \rceil}$.

Imagine a necklace of beads with a clasp. How many k-bead necklaces can be formed if we are given n different colors of round beads. When the necklace is worn we can tell the end of the necklace because of the clasp, but we can't distinguish a left end versus a right end. We can think of this as k-long lists where we consider two lists the same if one can be obtained from the other by reversing the list. If a list is a palindrome, it contributes one to the count. If a list is not a palindrome, the list and its reversal together contribute one to the count.

Let p be the number of palindrome lists and q the number of non-palindrome lists. We want $p+q/2$. The number of lists is $p+q$, which equals n^k and the number of palindromes is $n^{\lceil k/2 \rceil}$. Thus

$$p+q = n^k \qquad \text{and} \qquad p = n^{\lceil k/2 \rceil}$$

* The notation $\lceil x \rceil$ means least integer not less than x (that is, round up). For example $\lceil \pi \rceil = 4$.

and so $q = n^k - n^{\lceil k/2 \rceil}$. Finally we obtain our answer:

$$p + q/2 \;=\; n^{\lceil k/2 \rceil} + \frac{n^k - n^{\lceil k/2 \rceil}}{2} \;=\; \frac{n^{\lceil k/2 \rceil} + n^k}{2}. \quad \square$$

Example 1.7 Listing instead of counting Suppose we want to write a program to actually list the things in a set T rather than just counting them. Instead of computing $|T|$, we have to execute a program that lists all items $t \in T$. What about the Rules of Sum and Product? The Rule of Sum becomes

```
For each t_1 ∈ T_1: list t_1.
For each t_2 ∈ T_2: list t_2.
...
For each t_j ∈ T_j: list t_j.
```

The Rule of Product becomes

```
For each first choice d_1:
   ...
      For each kth choice d_k:
         List the structure arising from the choices d_1, ..., d_k.
      End for
   ...
End for
```

This is actually more general than Theorem 1.2 since, in the code, the number of choices in each loop may depend on previous choices. See Chapter 3 for more discussion. $\square$

Exercises

In each of the exercises, indicate how you are using the Rules of Sum and Product. You can do this with the AND/OR technique.

1.1.1. How many different three digit positive integers are there? (No leading zeroes are allowed.) How many positive integers with at most three digits? What are the answers when "three" is replaced by "n?"

1.1.2. A small neighboring country of the one we revisited in Example 1.3 has the same alphabet and the same rules of formation, but names are only five letters long. How many names are possible?

1.1.3. Prove that the number of subsets of a set S, including the empty set and S itself, is $2^{|S|}$.
Hint. For each element of S you must make one of two choices: "x is/isn't in the subset."

1.1.4. A *composition* of a positive integer n is an ordered list of positive integers (called *parts*) that sum to n. The four compositions of 3 are 3; 2,1; 1,2 and 1,1,1.

(a) By considering ways to insert plus signs and commas in a list of n ones, obtain a formula for the number of compositions of n.
Hint. The four compositions above correspond to 1+1+1; 1+1,1; 1,1+1 and 1,1,1, respectively.

(b) Prove that the average number of parts in a composition of n is $(n+1)/2$.
Hint. Reverse the roles of "+" and "," and then look at the number of parts in the original and role-reversed compositions.

*1.1.5. In Example 1.3 we found that there were 648 possible names. Suppose that these are listed in the usual dictionary order. What is the last word in the first half of the dictionary (the 324th word)? the first word in the second half?

1.2 Lists with Repetitions Forbidden

What happens if we do not allow repeats in our list? Suppose we have n elements to choose from and wish to form a k-list with no repeats. How many lists are there?

We can choose the first entry in the list AND choose the second entry AND $\cdots$ AND choose the kth entry. There are $n-i+1$ ways to choose the ith entry since $i-1$ elements have been removed from the set to make the first part of the list. By the Rule of Product, the number of lists is $n(n-1)\cdots(n-k+1)$. Using the notation $n!$ for the product of the first n integers and writing $0!=1$, you should be able to see that this answer can be written as $n!/(n-k)!$, which is often designated by $(n)_k$ and called the *falling factorial*. We have proven

Theorem 1.4 *When repeats are not allowed, there are $n!/(n-k)! = (n)_k$ k-lists that can be constructed from an n-set.*

When $k=n$, a list without repeats is simply a *linear ordering* of the set. We frequently say "ordering" instead of "linear ordering." An ordering is sometimes called a "permutation" of S. Thus, we have proven that a set S can be (linearly) ordered in $|S|!$ ways.

Example 1.8 Lists without repeats How many lists without repeats can be formed from a 5-set? There are $5! = 120$ 5-lists without repeats, $5!/1! = 120$ 4-lists without repeats, $5!/2! = 60$ 3-lists, $5!/3! = 20$ 2-lists and $5!/4! = 5$ 1-lists. By the Rule of Sum, this gives a total of 325 lists, or 326 if we count the empty list. In Exercise 1.2.11 you are asked to obtain an estimate when "5-set" is replaced with "n-set".

Suppose we have a problem involving k-lists with repeats allowed and we want the formula when repeats are not allowed. Since allowing repeats leads to powers and forbidding repeats leads to falling factorials, we might try to replace powers with falling factorials. Doing this without thinking, can easily give the wrong answers. Look back at Example 1.6 where we needed to count palindromes and obtained the formula $p = n^{\lceil k/2 \rceil}$. Except for 1-long lists, a palindrome has repeated elements; for example, the first and last elements are equal. Thus we obtain $p=n$ when $k=1$ and $p=0$ when $k>1$ for palindromes without repeats. □

Lists can appear in many guises. In this next example, the people could be thought of as the positions in a list and the seats the things in the list. Sometimes it helps to find a reinterpretation like this for a problem. At other times it is easier to tackle the problem starting over again from scratch. These methods can lead to several approaches to a problem. That can make the difference between a solution and no solution or between a simple solution and a complicated one. You should practice using both methods, even on the same problem.

Example 1.9 Linear arrangements How many different ways can 100 people be arranged in the seats in a classroom that has exactly 100 seats?

Each seating is simply an ordering of the people. Thus the answer is 100!. Simply writing 100! probably gives you little idea of the size of the number of seatings. A useful approximation for factorials is given by Stirling's formula:

Theorem 1.5 Stirling's formula *$\sqrt{2\pi n}\,(n/e)^n$ approximates $n!$ with a relative error under $1/10n$.*

We say that $f(x)$ approximates $g(x)$ with a *relative error* at most $\delta(x)$ if $|f(x)/g(x) - 1| \le \delta(x)$. Thus, the theorem states that $\sqrt{2\pi n}\,(n/e)^n/n!$ differs from 1 by less than $1/10n$. When relative error is multiplied by 100, we obtain "percentage error." If we simply want to note that the relative error goes to 0 as $n \to \infty$, we can write[1]

$$n! \;\sim\; \sqrt{2\pi n}(n/e)^n \quad \text{or, equivalently,} \quad n! \;=\; \sqrt{2\pi n}(n/e)^n(1+o(1)).$$

This is weaker than Theorem 1.5 because $o(1)$ stands for something that can be replaced by *some* function $h(n)$ with $\lim_{n\to\infty} h(n) = 0$, but the theorem tells us more, namely the function $h(n)$ is so small that $|h(n)| < 1/10n$.

By Stirling's formula, we find that 100! is nearly 9.32×10^{157}, which is much larger than estimates of the number of atoms in the universe.

Now suppose we still have 100 seats but have only 95 people. We need to think a bit more carefully than before. One approach is to put the people in some order (e.g., alphabetical), select a list of 95 seats, and then pair up people and seats so that the first person gets the first seat, the second person the second seat, and so on. By the general formula for lists without repetition, the answer is $100!/(100-95)! = 100!/120$. We can also solve this problem by thinking of the people as positions in a list and the seats as entries. Do it. □

The next example is starred because it is above the level of this chapter; therefore you may want to just skim it or maybe even omit it. It illustrates some of the calculations that one often runs into in obtaining estimates for large values of n and obtains the useful formula (1.2).

***Example 1.10 Estimating $n!/(n-k)!$** This example requires familiarity with the notations $O(\,)$ and $o(\,)$, which are discussed in Appendix B.

Suppose we want to estimate the number of k-lists without repeats that can be formed from an n-set; that is, we want to estimate $n!/(n-k)!$. In this example, we're interested in obtaining the estimate when n is large and k is much smaller than n. Of course, we can use Stirling's formula, which gives us the estimate

$$\frac{n!}{(n-k)!} \;\sim\; \frac{\sqrt{2\pi n}(n/e)^n}{\sqrt{2\pi(n-k)}((n-k)/e)^{n-k}} \;=\; \frac{n^{n+1/2}e^{-k}}{(n-k)^{n-k+1/2}}.$$

This is still rather messy. How can we simplify it? We have

$$\frac{n^{n+1/2}}{(n-k)^{n-k+1/2}} \;=\; n^k\left(\frac{n}{n-k}\right)^{n-k+1/2} \;=\; n^k\left(1+\frac{k}{n-k}\right)^{n-k+1/2}.$$

We need a result from calculus:

$$\text{If } x \text{ is small, then } \ln(1+x) \;=\; x - x^2/2 + O(x^3) \text{ and so } 1+x \;=\; \exp(x - x^2/2 + O(x^3)). \qquad 1.1$$

If you know Taylor's Theorem, you should be able to prove it; otherwise, just accept the result. Since k is much smaller than n, $\frac{k}{n-k}$ is small. Let it be x. By (1.1),

$$\left(1+\frac{k}{n-k}\right)^{n-k+1/2} \;=\; \exp(A(n-k+1/2)) \qquad \text{where} \qquad A \;=\; \frac{k}{n-k} - \frac{\left(\frac{k}{n-k}\right)^2}{2} + O((k/(n-k))^3).$$

With some algebra and the ability to work with $O(\,)$, one can deduce that

$$\exp(A(n-k+1/2)) \;=\; \exp(k - k^2/2n + O(k^3/n^2)).$$

[1] The notation in the next equations is discussed in Appendix B. It simply means that $n!\,/\,\left(\sqrt{2\pi n}(n/e)^n\right) \to 0$ as $n \to \infty$.

These manipulations are beyond what we expect of you at this point, so we'll omit them—you'll have to figure out how to do them or just accept this result.

Putting all this together:

$$\frac{n!}{(n-k)!} \sim \frac{n^{n+1/2}e^{-k}}{(n-k)^{n-k+1/2}} = n^k e^{-k}\exp(k + k^2/2n + O(k^3/n^2)).$$

If $k^3 = o(n^2)$, then $O(k^3/n^2) = o(1)$ and so $\exp(O(k^3/n^2)) = e^{o(1)} \sim 1$. Thus we have

$$\frac{n!}{(n-k)!} \sim n^k e^{-k^2/2n} \qquad \text{provided} \qquad k = o(n^{2/3}). \qquad 1.2$$

For example, by Theorem 1.4, the number of 200-lists without repeats that can be formed from a 10,000-set is about $10^{800}/e^2$. □

Example 1.11 Words from a collection of letters How many "words" of length k can be formed from the letters in ERROR when no letter may be used more often than it appears in ERROR? (A "word" is any list of letters, pronounceable or not.) If you are familiar with the game of Scrabble®, you can imagine that you have 5 tiles, namely one E, one O, and three R's. We cannot use 5^k since unlimited repetition is not allowed. On the other hand, we cannot use $(5)_k$ since repetition is allowed. At present, all we can do is carefully list the possibilities. Here they are in alphabetical order.

$k=1:$ E, O, R

$k=2:$ EO, ER, OE, OR, RE, RO, RR

$k=3:$ EOR, ERO, ERR, OER, ORE, ORR, REO, RER, ROE, ROR, RRE, RRO, RRR

$k=4:$ EORR, EROR, ERRO, ERRR, OERR, ORER, ORRE, ORRR, REOR, RERO, RERR, ROER, RORE, RORR, RREO, RRER, RROE, RROR, RRRE, RRRO

$k=5:$ EORRR, ERORR, ERROR, ERRRO, OERRR, ORERR, ORRER, ORRRE, REORR, REROR, RERRO, ROERR, RORER, RORRE, RREOR, RRERO, RROER, RRORE, RRREO, RRROE

This is obviously a tedious process—try it with ERRONEOUSNESS. We will explore better methods in Examples 1.19, 3.3 (p. 69), and 11.6 (p. 319). □

Example 1.12 Circular arrangements How many ways can n people be seated on a Ferris wheel with exactly one person in each seat? Equivalently, we can think of this as seating the people at a circular table with n chairs. Two seatings are defined to be "the same" if one can be obtained from the other by rotating the Ferris wheel (or rotating the seats around the table).

If the people were seated in a straight line instead of in a circle, the answer would be $n!$. Can we convert the circular seating into a linear seating (i.e., an ordered list)? In other words, *can we convert the unsolved problem to a solved one*? Certainly—simply cut the circular arrangement between two people and unroll it. Thus, to arrange n people in a linear ordering,

first arrange them in a circle AND then cut the circle.

According to our AND/OR technique, we must prove that each linear arrangement arises in *exactly one way* with this process.

- Since a linear seating can be rolled up into a circular seating, it can also be obtained by unrolling that circular seating. Hence each linear seating arises *at least once*.
- Since the people at the circular table are all different, the place we cut the circle determines who the first person in the linear seating is, so each cutting of a circular seating gives a different linear seating. Obviously two different circular seatings cannot give the same linear seating. Hence each linear seating arises *at most once*.

1 1 1 1 1 1	111111	1 2 1 1 1 1	111112 111121 111211	112111 121111 211111
1 2 1 1 2 1	112112 121121 211211	1 1 1 2 2 1	112121 121211 212111	121112 211121 111212
1 2 2 1 1 2	121212 212121	1 1 1 2 2 2	112221 122211 222111	221112 211122 111222

Figure 1.1 Some circular arrangements with the corresponding linear arrangements.

Putting these two observations together, we see that each linear seating arises *exactly once*. By the Rule of Product,

$$\begin{aligned} n! &= (\text{number of circular arrangements}) \times \\ &\quad (\text{number of places to cut the circle}). \end{aligned}$$

Hence the number of circular arrangements is $n!/n = (n-1)!$.

Our argument was somewhat indirect. We can derive the result by a more direct argument. For convenience, let the people be called 1 through n. We can read off the people in the circular list starting with person 1. This gives a linear ordering of $\underline{n}$ that starts with 1. Conversely, each such linear ordering gives rise to a circular ordering. Thus the number of circular orderings equals the number of such linear orderings. Having listed person 1, there are $(n-1)!$ ways to list the remaining $n-1$ people. Thus the number of circular arrangements is $(n-1)!$.

If we are making circular necklaces using n distinct beads, then the arguments we have just given prove that there are $(n-1)!$ possible necklaces provided we are not allowed to flip necklaces over. What happens if the beads are not distinct?

The direct method fails if there are multiple copies of bead 1 because we don't know where to start reading. What about the indirect method? The different cuttings of the circular arrangement may not be distinct. Let's have a look at an example to see why. We'll take a circular arrangement with six "places" and put beads of type 1 and 2 around the circle, where we can use any number of each of the two types of beads. In Figure 1.1 are some distinct necklaces and, next to each, the distinct linear arrangements we get by unrolling. There are 2^6 different linear arrangements. Since some necklaces have less than six unrollings, $2^6/6$ is an underestimate of the number of necklaces.

We can describe what we're doing as follows: Call two lists (i.e., linear arrangements) "equivalent" if one can be gotten from the other by "circularly permuting" the elements; that is, by shifting everything down some fixed number of positions and putting what is shifted off the end at the beginning. The lists fall into sets of equivalent lists, each set corresponding to one circular seating. Figure 1.1 can be thought of as containing six such sets of equivalent lists. The number of necklaces is the number of sets of equivalent lists.

Although we will not study tools for dealing with problems having equivalences until Chapter 4, there is one important class of problems with equivalences that we can deal with now. Suppose we allow the list entries to be rearranged in any fashion; in other words, we want to count unordered lists. We'll take up this subject in the next section. □

Our first derivation of the formula, $n!/n$, for seating n people at a circular table illustrates an important but obvious principle:

No matter how you count a set, the number is always the same.

For circular arrangements, we counted the set of linear arrangements in two ways. Another obvious principle is

If there is a one-to-one correspondence between two sets, then they are the same size.

This can be used to show that two counting problems have the same answer. In the next example we consider a famous example of this—the Catalan numbers, which arise in a variety of counting problems.

Example 1.13 Catalan numbers Suppose we have an election between two candidates and the ballots are counted one-by-one. Further suppose that the first candidate is never behind (she's always ahead or tied), but that the final count ends in a tie with each candidate getting n votes. How many ways can this happen? The answer is called the Catalan number C_n. We are looking at ordered lists of that contain n ones and n twos such that, for all k, the number of twos in the first k elements is at most $k/2$. The lists for $n \leq 3$ are

$$12 \qquad 1122 \quad 1212 \qquad 111222 \quad 112122 \quad 112212 \quad 121122 \quad 121212$$

and so $C_1 = 1$, $C_2 = 2$, $C_3 = 5$. In general $C_n = \frac{(2n)!}{n!\,(n+1)!}$. We won't derive the formula for C_n now, but we want to look at other problems that have the same answer. (If you look up Catalan numbers in the index, you can find a derivation of the formula in the text as well as other problems that have the same answer.)

In computer science we have the notion of a *stack*. This is an ordered list with two operations:

- PUSH: Add an item to the end of the list.
- POP: Remove an item from the end of the list.

It is illegal to attempt to "POP" an empty stack. How many ways can we start out with an empty stack, PUSH and POP in some order and end up with an empty stack at the end? There must be the same number of PUSHs and POPs. Suppose there are n of each. You should be able to convince yourself that this is the same as the election problem and so the answer is C_n.

Suppose we have n things we want to multiply together. In general, $ab \neq ba$ so order matters; however, we can group them in any way we want. (This is true if the things being multiplied are matrices.) For example, here are the ways we could group four things for multiplication.

$$a(b(cd)) \quad a((bc)d) \quad (ab)(cd) \quad (a(bc))d \quad ((ab)c)d.$$

We can do this with a stack using one of two operations:

- STORE: PUSH the next thing onto the stack
- MULT: POP two things off the stack and PUSH their product onto the stack.

For example, to do $a((bc)d)$ we would do

STORE, STORE, STORE, MULT, STORE, MULT, MULT.

There must be n STOREs to get all n items onto the stack. There must be $n-1$ MULTs. The number of STOREs in the first k things must exceed the number of MULTs. (Can you see why the last two statements are true?) Forgetting the first STORE, this is just the original voting problem with $n-1$ votes each. Thus the answer is C_{n-1}.

A regular n-gon can be cut up into triangles all of whose vertices are vertices of the n-gon. To do this, one must draw $n-3$ nonintersecting diagonals. We call this a "triangulation of the n-gon." Here are the five triangulations of the pentagon.

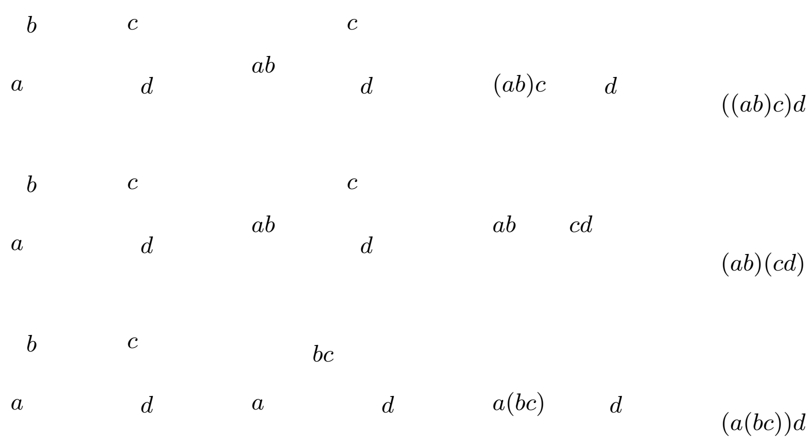


Figure 1.2 The reduction of three of the five triangulations of the pentagon to multiplications of *abcd*.

We want to know how many triangulations there are for a regular n-gon. This is trickier than the previous correspondences. First, we need to know a little about what the triangulations look like.

It turns out that, for $n > 3$, every triangulation has $n-3$ diagonals, $n-2$ triangles and exactly two triangles that contain two edges of the original n-gon. Actually, any of these three claims can be used to prove the other two. To see this, suppose there are D diagonals and T triangles. Then the triangles have a total of $3T$ edges. These edges come from the original n-gon and from *both sides* of the diagonals. Thus $3T = n + 2D$. It is clear that every triangle contains either one or two edges of the n-gon. Call the number of these triangles T_1 and T_2, respectively. Then $T_1 + T_2 = T$ and $T_1 + 2T_2 = n$. In summary

$$3T = n + 2D \qquad T_1 + T_2 = n \qquad T_1 + 2T_2 = n.$$

We have three equations in the four unknowns D, T, T_1 and T_2. If any of these is known (e.g., $D = n-3$), we can solve the equations for the other three. Which value should we determine so that the others can be found?

We'll prove that $D = n-3$. This is even true for 3-gons (triangles) since no diagonals are needed. We'll use induction for $n > 3$. Suppose we are given a triangulation of an n-gon. Cut it along any diagonal to split it into two polygons. Let the number of sides of the two polygons be k_1 and k_2. Since cutting along the diagonal has given us two new sides, $k_1 + k_2 = n + 2$. Notice that the k_1-gon and k_2-gon are triangulated. By induction, the k_1-gon has $k_1 - 3$ diagonals and the k_2-gon has $k_2 - 3$. Thus, counting the diagonal we cut along, the number of diagonals in the original n-gon triangulation is

$$(k_1 - 3) + (k_2 - 3) + 1 \;=\; (k_1 + k_2) - 5 \;=\; (n+2) - 5 \;=\; n - 3,$$

and the induction is complete.

We'll now describe a method for associating a multiplication of $n-1$ things with a triangulation of an n-gon. Draw the n-gon with one side at the bottom. We'll call this side the "base". Label all the sides except the base. (See the left side of Figure 1.2.) There are two triangles that have two sides belonging to the n-gon. Thus there must be a triangle with two labeled sides. Remove the labeled sides and place the product of their labels on the third side. Repeat this process until we are left with a labeled base. Figure 1.2 contains examples.

To complete the process we need to know that this gives us a one-to-one correspondence between the triangulations and the multiplications. Simply write the multiplication on the base and reverse the steps. In other words, read Figure 1.2 from right to left instead of from left to right. We leave it to you to convince yourself that every multiplication leads to a unique triangulation and vice versa. Thus there are C_{n-2} triangulations of a regular n-gon.

We have looked at only a few of the dozens of combinatorial interpretations of the Catalan numbers. □

Exercises

In each of the exercises, indicate how you are using the Rules of Sum and Product. It is instructive to first do these exercises using only the techniques introduced so far and then, after reading the next section, to return to these exercises and look for other ways of doing them. More generally, looking back at earlier sections to get a new viewpoint is often helpful. We do this in the text to some extent, but you should do it on your own, too.

1.2.1. Find to two decimal places the answer to the birthday question asked in Example 1 (p. 1).
Hint. Assigning birthdays to 30 people is the same as forming an ordered list of 30 dates.

1.2.2. Use (1.2) to estimate the solution to the birthday problem in Example 1 (p. 1).

1.2.3. How many ways are there to form an ordered list of two distinct letters from the set of letters in the word COMBINATORICS? three distinct letters? four distinct letters?

1.2.4. Repeat the previous problem when the letters need not be distinct but cannot be used more often than they appear in COMBINATORICS.

1.2.5. We are interested in forming 3 letter words ("3-words") using the letters in LITTLEST. For the purposes of the problem, a "word" is any ordered list of letters.

(a) How many words can be made with no repeated letters?

(b) How many words can be made with unlimited repetition allowed?

(c) How many words can be made if repeats are allowed but no letter can be used more often than it appears in LITTLEST?

1.2.6. Redo the previous exercise for k-words. The last part should be starred. It can be done if you treat each value of $k \leq 8$ separately and carefully break it down into cases with OR. Even so, you should study the next section before you attempt it.

1.2.7. Each of the following belongs to one of the four types of things described in Example 1.13. In each case, list the other three things that correspond to it using the correspondences in the example.

(a) 1122112122

(b) $(a(bc))(((de)f)g)$

(c)

1.2.8. Suppose we have an election as in Example 1.13, but now the first candidate is *always* ahead except for the 0–0 and n–n ties at the start and finish. How many ways can this happen?

1.2.9. By 2001 spelling has deteriorated considerably. The dictionary defines the spelling of "relief" to be any combination (with repetition allowed) of the letters R, L, F, I and E subject to certain constraints listed below How many spellings are possible? The most popular spelling is the one that, in dictionary order, is five before the spelling RELIEF. What is it?

(i) The number of letters must not exceed 6.

(ii) The word must contain at least one L.

(iii) The word must begin with an R and end with an F.

(iv) There is just one R and one F.

1.2.10. By the year 2010, further deterioration in spelling has relaxed the last condition listed above so that we can have any number of initial R's and any number of terminal F's, provided there is at least one of each. How many spellings are possible? Which spelling is five before RELIEF in dictionary order?

1.2.11. Prove that the number of ordered lists without repeats that can be constructed from an n-set is very nearly $n!e$. The lists can be of any length.
Hint. Recall that from Taylor's Theorem in calculus $e^x = 1 + x + x^2/2! + x^3/3! + \cdots$.

1.2.12. In this exercise, we look at ways of seating n people at a long table that has n seats. In (c)–(e), n is even.
Hint. If you fix a corner of the table and read out the seating arrangement counterclockwise starting at that corner, you have an ordered list. If you draw pictures, you should be able to see how many ordered lists give an equivalent seating arrangement; for example, by reversing right and left in (b).

(a) Suppose that everyone is to be seated on one side of the table. How many ways can it be done?

(b) Suppose we don't care if left and right are interchanged; that is, seating $A, B, C, \ldots$ from left to right will be considered the same as doing it from right to left. (This is reasonable if all we care about is who a person's neighbors are.) How many ways can this be done?

(c) How many ways can it be done if n is even and half the people are seated on each side of the table? Assume that we can tell the two sides of the table apart; for example, one side faces a wall and the other side faces into the room. Also assume seating left to right is different from seating right to left.

(d) Suppose we seat people on both sides as in (c) and all we care about is who a person's neighbors are on each side, as in (b).

(e) Suppose we are dealing with a seating as in (d), but now we also care about who is sitting opposite a person as well as who a persons neighbors on each side are.

*1.2.13. This exercise contains several related questions. In each case we would like a formula that answers the question "How many ways can p people run for k offices?" under the given constraints. Unless the constraints say otherwise, a person may run for no offices. At present, we have the tools to do only two parts of this exercise. The challenge in this exercise is to avoid finding wrong "solutions" to the parts that we are unable to do, as well as doing the two parts we can do now. One way you can check your "solution" is to actually list all the possible ways p people can run for k offices for each of the parts for some small values of p and k. We will return to this exercise later as we develop tools for doing other parts of it.

(a) Each person must be a candidate for at most one office.

(b) Each person must be a candidate for exactly one office and each office must have at least one candidate.

(c) Each person must be a candidate for at most one office and each office must have at least one candidate.

(d) Each person can be a candidate for any number of offices (including none) and each office must have at least one candidate.

(e) Each person must be a candidate for at least one office and each office must have at least one candidate.

*1.2.14. In Example 1.12: How many are there of length 3 made from A's and B's? Length 5? Can you prove a general result for all primes? What about allowing more than two kinds of letters?

1.3 Sets

People use $C(n,k)$ to stand for the number of different k-subsets that can be formed from an n-set. The notation $\binom{n}{k}$ is also frequently used. These are called *binomial coefficients* and are read "n choose k." Think about how you might count k-subsets, that is, unordered k-lists.

* * * Stop and think about this! * * *

You may have concluded that this seems a bit trickier to do than counting ordered lists. Can we rephrase the problem in a way that lets us solve it, or convert it to an ordered list problem?

- An unordered k-list of distinct elements from a set S is simply a k-subset of S. This doesn't seem to be of any help at present; however, we will generally think in terms of subsets rather than unordered lists since the subset view is used more often in the literature.
- If the original set consisted of something ordered, like the integers, we could introduce a "natural" ordering to an unordered list, namely the one in which the elements are in increasing order (or, if you prefer, decreasing order). Again this doesn't seem to help, but provides a possibly useful interpretation.
- We can adjust the previous idea a bit. Let's consider all possible orderings of our lists. This is a way of constructing all ordered lists with distinct elements in two steps: First construct an unordered list with no repeats, then order it. An unordered k-list with no repeats is simply a k-set. We can order it by forming a k-list without repeats from it. By Theorem 1.4 (p. 11), we know that this can be done in $k!$ ways. By the Rule of Product, there are $C(n,k)k!$ ordered k-lists with no repeats. By Theorem 1.4 again, this number is $n(n-1)\cdots(n-k+1) = n!/(n-k)!$. Dividing by $k!$, we have

Theorem 1.6 Binomial coefficient formula *The value of the binomial coefficients is*

$$\binom{n}{k} = C(n,k) = \frac{n(n-1)\cdots(n-k+1)}{k!} = \frac{n!}{k!\,(n-k)!}.$$

Example 1.14 A generating function for binomial coefficients We'll now approach the problem of evaluating $C(n,k)$ in another way. In other words, we'll "forget" the formula we just derived and start over with a new approach.

You may ask "Why waste time using another approach when we've already gotten what we want?" We gave a partial answer to this earlier. Here is a more complete response.

- By looking at a problem from different viewpoints, we may come to understand it better and so be more comfortable working similar problems in the future.
- By looking at a problem from different viewpoints, we may discover that things we previously thought were unrelated have interesting connections. These connections might open up easier ways to solve some types of problems and may make it possible for us to solve problems we couldn't do before.
- A different point of view may lead us to a whole new approach to problems, putting powerful new tools at our disposal.

In the approach we are about to take, we'll begin to see a powerful tool for solving counting problems. It's called "generating functions" and it lets us put calculus and related subjects to work in combinatorics. In later chapters, we'll devote more time to generating functions. Now, we'll just get a brief glimpse of them.

Suppose that $S = \{x_1, \ldots, x_n\}$ where x_1, x_2, … and x_n are variables as in high school algebra. Let $P(S) = (1+x_1)\cdots(1+x_n)$. The first three values of $P(S)$ are

$$\begin{aligned} n=1: &\quad 1+x_1 \\ n=2: &\quad 1+x_1+x_2+x_1x_2 \\ n=3: &\quad 1+x_1+x_2+x_3+x_1x_2+x_1x_3+x_2x_3+x_1x_2x_3. \end{aligned}$$

From this you should be able to convince yourself that $P(S)$ consists of a sum of terms where each term represents one of the subsets of S as a product of its elements. Can we reach some understanding of why this is so? Yes, but we'll only explore it briefly now. The understanding relates to the Rules of Sum and Product. Interpret plus as OR, times as AND and 1 as "nothing." Then $(1+x_1)(1+x_2)(1+x_3)$ can be read as

- include the factor 1 in the term OR include the factor x_1 AND
- include the factor 1 in the term OR include the factor x_2 AND
- include the factor 1 in the term OR include the factor x_3.

This is simply a description of how to form an arbitrary subset of $\{x_1, x_2, x_3\}$. On the other hand we can form an arbitrary subset by the rule

- Include nothing in the subset OR
- include x_1 in the subset OR
- include x_2 in the subset OR
- include x_3 in the subset OR
- include x_1 AND x_2 in the subset OR
- include x_1 AND x_3 in the subset OR
- include x_2 AND x_3 in the subset OR
- include x_1 AND x_2 AND x_3 in the subset.

If we drop the subscripts on the x_i's, then a product representing a k-subset becomes x^k. We get one such term for each subset and so it follows that the coefficient of x^k in the polynomial $f(x) = (1+x)^n$ is $C(n,k)$; that is,

$$(1+x)^n \;=\; \sum_{k=0}^{n} C(n,k)x^k. \qquad 1.3$$

Can this help us evaluate $C(n,k)$? Calculus comes to the rescue! Remember Taylor's Theorem? It tells us that the coefficient of x^k in $f(x)$ is $f^{(k)}(0)/k!$. Let $f(x) = (1+x)^n$. You should be able to prove by induction on k that

$$f^{(k)}(x) \;=\; n(n-1)\cdots(n-k+1)\,(1+x)^{n-k}.$$

Thus $c(n,k)$, the coefficient of x^k in $(1+x)^n$, is

$$C(n,k) \;=\; \frac{f^{(k)}(0)}{k!} \;=\; \frac{n(n-1)\cdots(n-k+1)}{k!}.$$

We conclude this example with a useful formula that follows from (1.3). Since $(x+y)^n = x^n(1+(y/x))^n$, it follows that the coefficient of $x^n(y/x)^k$ in $(x+y)^n$ is $C(n,k)$. This gives us the

Theorem 1.7 Binomial Theorem

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

The expressions we've been studying are called *generating functions.* □

Example 1.15 Card hands: Full house Card hands provide a source of some simple sounding but tricky set counting problems. A standard deck of cards contains 52 cards, each of which is marked with two labels. The first label, called the "suit," belongs to the set

$$\{\clubsuit, \heartsuit, \diamondsuit, \spadesuit\}.$$

The second label, called the "value" belongs to the set

$$\{2, 3, 4, 5, 6, 7, 8, 9, 10, \mathrm{J}, \mathrm{Q}, \mathrm{K}, \mathrm{A}\}.$$

Each pair of labels occurs exactly once in the deck. A hand is a subset of a deck. Two cards are a pair if they have the same values.

How many 5 card hands consist of a pair and a triple? (In poker, such a hand is called a full house.)

To calculate this we describe how to construct such a hand:

- Choose the value for the pair AND
- Choose the value for the triple different from the pair AND
- Choose the 2 suits for the pair AND
- Choose the 3 suits for the triple.

This produces each full house exactly once, so the number is the product of the answers for the four steps, namely

$$13 \times 12 \times C(4,2) \times C(4,3) \ = \ 3{,}744.$$

What is the probability of being dealt a full house? There are $\binom{52}{5}$ distinct hands of cards so we could simply divide the previous answer by this number. This approach looks at the *result* of the deal rather than the actual deal. Why do we say that? When a hand of cards is dealt, the *order* in which you receive the cards matters. Thus:

- If we look at the resulting hand, then the order of the cards doesn't matter. That's the way we just got the answer.
- If we look at the dealing process, then the order of the cards matters. We'll do the problem that way next.

Each of the $52 \times 51 \times 50 \times 49 \times 48$ ways of dealing five cards from 52 as equally likely. Then we should divide this into the number of ways of being dealt a full house. Since all the cards in a hand of five cards are different, they can be ordered in 5! ways. Hence the probability of being dealt a full house is $\frac{3{,}774 \times 5!}{52\times 51\times 50\times 49\times 48}$, which gives the same answer as before, $3{,}744/\binom{52}{5}$.

Let's phrase these in terms of probability spaces. We'll use the uniform distribution on both spaces.

- Resulting hand: The space contains all $\binom{52}{5}$ 5-card subsets of the 52-card deck.
- Dealing process: The space contains all $52 \times 51 \times 50 \times 49 \times 48$ 5-card lists without repeats that can be made from a 52-card deck.

Which approach should you use? That's up to you. However, whichever approach you choose, there you may have problems if there is more than one copy of a card. For example, one might add two jokers to a deck or one might combine two identical decks as in canasta. In this case, it's probably easiest and safest to pretend that the cards have been marked so you can tell them apart; for example, call them joker-1 and joker-2. □

Example 1.16 Card hands: Two pairs We'll continue with our poker hands. How many 5 card hands consist of two pairs? A description of a hand always means that there is nothing better in the hand, so "two pairs" means we don't have a full house or four of a kind.

One thing we might try is to go back to the preceding example's description of how to construct a full house and two simple changes: (a) replace "triple" by "second pair" and (b) add a choice for the card that belongs to no pair. This is wrong! Each hand is constructed *twice*, depending on which pair is the "second pair." Try it! What happened? Before choosing the cards for a pair and a triple, we can distinguish the pair from the triple because one contains two cards and the other contains three. We can't distinguish the two pairs, though, until the values are specified. This is an example of a situation where we can easily make mistakes if we forget that "AND" means "AND then." Here's a correct description, with "then" put in for emphasis.

- Choose the values for the two pairs AND then
- Choose the 2 suits for the pair with the larger value AND then
- Choose the 2 suits for the pair with the smaller value AND then
- Choose the remaining card from the 4×11 cards that have different values than the pairs.

The value is

$$\binom{13}{2} \times \binom{4}{2} \times \binom{4}{2} \times 44 \;=\; 123{,}552.$$

You may find what we've just been through disquieting: How can you decide between distinguishable and indistinguishable? The answer is simple: Draw a picture of the cards and fill in the information after each step. Let's do this for the full house and the two pair problems. To begin with, we have five blank cards. For the full house, we divide the cards up into a pair and a triple:

.

We can tell the two groups apart, so it makes sense to talk about assigning a value to the pair, say 9, and to the triple, say 7, to obtain

.

We can't tell the two nines apart, so all we can do is choose a subset of two suits to assign to them; likewise for the triple. We might choose $\{\heartsuit, \clubsuit\}$ and $\{\heartsuit, \spadesuit, \clubsuit\}$ and obtain

.

Now look at the case of two pairs. We have

.

Since we can't distinguish the two pairs, all we can do is choose a set of two values, say $\{7, 9\}$ and put them on the cards:

.

Now we can distinguish between the pairs. For the pair of sevens we might choose the set $\{\heartsuit, \clubsuit\}$ of suits, and for the nines, $\{\heartsuit, \spadesuit\}$. As a result, we have

. □

Example 1.17 Smorgasbord College programs Smorgasbord College allows students to study in three principal areas: (a) Swiss naval history, (b) elementary theory and (c) computer science. The number of upper division courses offered in these fields are 2, 92, and 15 respectively. To graduate a student must choose a major and take 6 upper division courses in it, and also choose a minor and take 2 upper division courses in it. Swiss naval history cannot be a major because only 2 upper division courses are offered in it.

How many programs are possible?

The possible major-minor pairs are b-a, b-c, c-a, and c-b. By the Rule of Sum we can simply add up the number of programs in each combination. Those programs can be found by the Rule of Product. The number of major programs in (b) is $C(92, 6)$ and in (c) is $C(15, 6)$. For minor programs: (a) is $C(2, 2) = 1$, (b) is $C(92, 2) = 4186$ and (c) is $C(15, 2) = 105$. Since the possible programs are constructed by

$$\begin{aligned} &\Big(\text{major (b) AND } \big(\text{minor (a) OR minor (c)}\big)\Big) \\ \text{OR } &\Big(\text{major (c) AND } \big(\text{minor (a) OR minor (b)}\big)\Big), \end{aligned}$$

the number of possible programs is

$$\binom{92}{6}(1 + 105) + \binom{15}{6}(1 + 4186) \;=\; 75{,}606{,}201{,}671,$$

a rather large number. □

Example 1.18 Multinomial coefficients Suppose we are given k boxes labeled 1 through k and a set S and are told to distribute the elements of S among the boxes so that the ith box contains exactly m_i elements. How many ways can this be done?

Let $n = |S|$. Unless $m_1 + \cdots + m_k = n$, the answer is zero because we don't have the right number of objects. Therefore, we assume from now on that

$$m_1 + \cdots + m_k = n.$$

Here's a way to describe filling the boxes.

- Fill the first box (There are $C(n, m_1)$ ways.) AND
- Fill the second box (There are $C(n - m_1, m_2)$ ways.) AND

  . . . . . .

- Fill the kth box. (There are $C(n - (m_1 + \ldots + m_{k-1}), m_k) = C(m_k, m_k) = 1$ ways.)

Now apply the Rule of Product, use the formula $C(p,q) = p!/q!(p-q)!$ everywhere, and cancel common factors in numerator and denominator to obtain $n!/m_1!m_2!\cdots m_k!$. This is called a *multinomial coefficient* and is written

$$\binom{n}{m_1, m_2, \ldots, m_k} = \frac{n!}{m_1!m_2!\cdots m_k!}, \tag{1.4}$$

where $n = m_1 + m_2 + \ldots + m_k$. In multinomial notation, the binomial coefficient $\binom{n}{k}$ would be written $\binom{n}{k,(n-k)}$. You can think of the first box as the k things that are chosen and the second box as the $n-k$ things that are not chosen.

Before you read on, try to think of an ordered list interpretation for the multinomial coefficient.

* * * Stop and think about this! * * *

Think of the objects being distributed as positions in a word and the boxes as letters. If the object "position 3" is placed in the box "D," then the letter D is the third letter in the word. The multinomial coefficient is then the number of words that can be made so that letter i appears exactly m_i times. A word can be thought of as an ordered list of its letters. □

Example 1.19 Words from a collection of letters Using the idea at the end of the previous example, we can more easily count the words that can be made from ERROR, a problem discussed in Example 1.11 (p. 13). Suppose we want to make words of length k. Let m_1 be the number of E's, m_2 the number of O's and m_3 the number of R's. By considering all possible cases for the number of each letter, you should be able to see that the answer is the sum of $\binom{k}{m_1,m_2,m_3}$ over all m_1, m_2, m_3 such that

$$m_1 + m_2 + m_3 = k, \quad 0 \le m_1 \le 1, \quad 0 \le m_2 \le 1, \quad 0 \le m_3 \le 3.$$

Thus we obtain

$$\begin{aligned}
k=1: &\quad \binom{1}{0,0,1} + \binom{1}{0,1,0} + \binom{1}{1,0,0} = 3\\
k=2: &\quad \binom{2}{0,0,2} + \binom{2}{0,1,1} + \binom{2}{1,0,1} + \binom{2}{1,1,0} = 7\\
k=3: &\quad \binom{3}{0,0,3} + \binom{3}{0,1,2} + \binom{3}{1,0,2} + \binom{3}{1,1,1} = 13\\
k=4: &\quad \binom{4}{0,1,3} + \binom{4}{1,0,3} + \binom{4}{1,1,2} = 20\\
k=5: &\quad \binom{5}{1,1,3} = 20.
\end{aligned}$$

This is better than in Example 1.11. Instead of having to list words, we have to list triples of numbers and each triple generally corresponds to more than one word. Here's the lists for the preceding computations

$$\begin{aligned}
k=1: &\quad 0,0,1 \quad 0,1,0 \quad 1,0,0\\
k=2: &\quad 0,0,2 \quad 0,1,1 \quad 1,0,1 \quad 1,1,0\\
k=3: &\quad 0,0,3 \quad 0,1,2 \quad 1,0,2 \quad 1,1,1\\
k=4: &\quad 0,1,3 \quad 1,0,3 \quad 1,1,2\\
k=5: &\quad 1,1,3
\end{aligned}$$

In Example 3.3 (p. 69), we will see how to do this more systematically and efficiently. □

Example 1.20 Card hands and multinomial coefficients We'll redo Examples 1.15 and 1.16, and then discuss the general situation using multinomial coefficients.

To form a full house, we must choose a face value for the triple, choose a face value for the pair, and leave eleven face values unused. This can be done in $\binom{13}{1,1,11}$ ways. We then choose the suits for the triple in $\binom{4}{3}$ ways and the suits for the pair in $\binom{4}{2}$ ways.

To form two pair, we must choose two face values for the pairs, choose a face value for the single card, and leave ten face values unused. This can be done in $\binom{13}{2,1,10}$ ways. We then choose suits for each of the face values in turn, so we must multiply by $\binom{4}{2}\binom{4}{2}\binom{4}{1}$.

Imagine an eleven card hand containing two triples, a pair and three single cards. You should be able to see that the number of ways to do this is

$$\binom{13}{2,1,3,7}\binom{4}{3}\binom{4}{3}\binom{4}{2}\binom{4}{1}\binom{4}{1}\binom{4}{1}.$$

Let's do the general case. Suppose our hand must contain c_1 singles, c_2 pairs, c_3 triples and c_4 four-of-a-kinds. The number of such hands is

$$\binom{13}{c_1,c_2,c_3,c_4,k}\binom{4}{1}^{c_1}\binom{4}{2}^{c_2}\binom{4}{3}^{c_3}\binom{4}{4}^{c_4},$$

where $k = 13 - c_1 - c_2 - c_3 - c_4$ is the number of face values not in the hand. □

Example 1.21 Choosing Teams Given 22 people, how many ways can we divide them into 4 teams of 5 players each plus 2 referees? If the teams and referees were labeled, the answer would be $\binom{22}{5,5,5,5,1,1}$. Given 4 different teams and two referees, there are 4! ways to label the teams as Team 1, 2, 3, and 4, and there are 2 ways to label the referees, so the answer is

$$\binom{22}{5,5,5,5,1,1}\frac{1}{4!\times 2} = \frac{(22)!}{2(5!)^4\,4!}.$$

Suppose now we must divide up the teams into pairs that compete against each other, and we assign a referee to each pair. If they were called Match #1 and Match #2, we could fill out Match #1 by choosing 2 of the 4 teams and 1 of the referees. Those left are Match #2. This gives us $\binom{4}{2}\times 2 = 12$. Thus we have $\frac{(22)!}{2(5!)^4\,4!}\times 12$. Of course, there isn't really a Match #1 and Match #2, but there are two ways to assign match labels and so we must divide the answer we just got by 2. □

***Example 1.22 Incomparable sets** Let A be an n-set. By a *Sperner family* on A we mean a family of subsets of A such that no subset in the family is contained in any other subset in the family. For example, let $A = \{1,2,3,4,5\}$. Then

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

is a Sperner family but

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

is not.

What is the largest number of subsets that we can have in a Sperner family of an n-set?

Clearly the family of all k-subsets of A is a Sperner family. Thus we can construct Sperner families of size at least $\binom{n}{k}$. What value of k will make this as large as possible? One way to find the value of k is to look at the ratio of $\binom{n}{k}$ to $\binom{n}{k-1}$. When this ratio exceeds 1, the sequence of binomial coefficients is increasing and when it is less than 1 the sequence is decreasing. Since

$$\frac{\binom{n}{k}}{\binom{n}{k-1}} = \frac{n!\,(n-k+1)!\,(k-1)!}{(n-k)!\,k!\,n!} = \frac{n-k+1}{k} = \frac{n+1}{k} - 1,$$

we see that the sequence is increasing when $(n+1)/k > 2$ and is decreasing when $(n+1)/k < 2$. It follows that

$$\binom{n}{k} \text{ is a maximum at } \begin{cases} k = n/2, \text{ when } n \text{ is even;} \\ k = (n-1)/2 \text{ and } k = (n+1)/2, \text{ when } n \text{ is odd.} \end{cases}$$

$\lfloor x \rfloor$, the floor function, denotes the largest integer not exceeding x. With this, we can write our conclusions in the form: There is a Sperner family of size $\binom{n}{\lfloor n/2 \rfloor}$, which can be obtained by taking all $\lfloor n/2 \rfloor$-subsets of A.

Sperner proved that this result is best possible: there are no larger Sperner families. We now present an adaptation of Lubell's proof of this result.

Call a k-set B an "initial part" of a list L if the first k elements of L are the elements of B. Let $\mathcal{S}$ be a Sperner family on the n-set A. Consider an n-list L of the n elements of A. We claim that at most one set in $\mathcal{S}$ can be an initial part of L, for if there were two such sets, one would correspond to a longer initial part than the other and so contain the other as a subset.

On the other hand, a k-set B is the initial part of exactly $k!\,(n-k)!$ n-lists. Why is this? The first k elements of the list must be some arrangement of the elements of B, AND the remaining $n-k$ elements of the list must be some arrangement of the remaining $n-k$ elements of S. Furthermore, any list satisfying these conditions has B as an initial part. By the Rule of Product and Theorem 1.4 (p. 11), there are $k!\,(n-k)!$ permutations which have B as an initial part. Adding this up over all B in $\mathcal{S}$, we obtain the number of rearrangements of A that have sets in $\mathcal{S}$ as initial parts. (This uses the result from previous paragraph that each list has at most one element of $\mathcal{S}$ as an initial part.) Since there are $n!$ lists, we have proved

$$\sum_{B \in \mathcal{S}} |B|!\,(n-|B|)! \;\le\; n!.$$

Dividing by $n!$ we obtain

$$\sum_{B \in \mathcal{S}} \frac{1}{\binom{n}{|B|}} \;\le\; 1. \qquad 1.5$$

This inequality is the key to the proof. By our earlier work on the size of binomial coefficients, we know that each term in the sum in (1.5) is at least as big as $1/\binom{n}{\lfloor n/2 \rfloor}$. Consequently, the sum in (1.5) can have at most $\binom{n}{\lfloor n/2 \rfloor}$ terms. In other words the size of the Sperner family is at most $\binom{n}{\lfloor n/2 \rfloor}$—but we have already constructed Sperner families this big! This completes the proof. □

***Example 1.23 When are two subsets disjoint?** Alice chooses a k-subset at random from and n-set. Bob chooses an l-subset at random from the same n-set. Find an exact expression and a simple estimate the probability that the two subsets are disjoint. For the estimate, you may assume that $k = o(n^{2/3})$ and $l = o(n^{2/3})$.

Call the probability $P(n,k,l)$. By the Rule of Product, there are $\binom{n}{k}\binom{n}{l}$ ways to choose two subsets of the given sizes. There are $\binom{n}{k}\binom{n-k}{l}$ ways to choose two disjoint subsets of the given sizes. Since things are done at random, all choices are equally likely and so

$$P(n,k,l) \;=\; \frac{\binom{n}{k}\binom{n-k}{l}}{\binom{n}{k}\binom{n}{l}} \;=\; \frac{(n-k)!/(n-k-l)!}{n!/(n-l)!} \;=\; \frac{(n-k)!}{n!} \times \frac{(n-l)!}{((n-l)-k)!}.$$

This is the exact answer written in various forms.

The exact answer does not give us a good idea of how the probability behaves when the numbers are large. To get a simple estimate, we use (1.2):

$$\frac{n!}{(n-k)!} \;\sim\; n^k e^{-k^2/2n} \qquad \text{and} \qquad \frac{(n-l)!}{((n-l)-k)!} \;\sim\; (n-l)^k e^{-k^2/2(n-l)}.$$

Thus

$$P(n,k,l) \;\sim\; \frac{(n-l)^k e^{-k^2/2(n-l)}}{n^k e^{-k^2/2n}} \;=\; \left(1-\frac{l}{n}\right)^k \exp\left(-\frac{k^2 l}{2n(n-l)}\right).$$

We need to look at the two factors on the right. Inside the exponential we have $\frac{-k^2 l}{2n(n-l)}$. Since l is small compared to n, this is nearly $\frac{-k^2 l}{2n^2}$. Since $k = o(n^{2/3})$ and $l = o(n^{2/3})$, we have $k^2 l = o((n^{2/3})^2 n^{2/3})$. Combining the exponents on the right, $k^2 l = o(n^2)$. Thus $\frac{k^l}{2n(n-l)} \to 0 = o(1)$. Since the exponential of a number close to zero is close to one, (1.5) becomes

$$\begin{aligned} P(n,k,l) \;&\sim\; \left(1-\frac{l}{n}\right)^k \;=\; \exp\left(k \ln\left(1-\frac{l}{n}\right)\right) \\ &\sim\; \exp\big(k\big((-l/n) - (-l/n)^2/2 + O(l^3/n^3)\big)\big) \qquad \text{by (1.1)} \\ &=\; \exp\big(-kl/n - (kl^2/2n^2) + O(kl^3/n^3)\big). \end{aligned}$$

You should be able show that $kl^2/2n^2 = o(1)$ and $kl^3/n^3 = o(n^{-1/3})$. Thus we have

$$P(n,k,l) \;\sim\; e^{-kl/n} \quad \text{provided } k = o(n^{2/3}) \text{ and } l = o(n^{2/3}).$$

Our constraints on the growth of k and l was necessary so that we could obtain our result, but looking at the result we can see some justification for the constraints: When kl is much larger than n, the probability will be very close to 0 and so may be uninteresting. If k and l are about the same size, the low probability occurs when they grow faster than $n^{1/2}$. $\square$

*Error Correcting Codes

We want to represent information by n-strings of zeroes and ones. For example, the information may be a letter of the alphabet. ASCII provides a way of doing this: an 8-string is used to represent the upper and lower case alphabet, the digits, the punctuation marks and some special "characters."

The ASCII representation of characters is quite sensitive to errors: if even a single entry in the 8-string is changed, we end up with a completely different character. This may be unacceptable. For example, suppose the characters are being transmitted over a data link which may have a small amount of static, the effect of which is to sometimes change a zero to a one or vice versa. A Soviet space probe was lost in 1989 because of a single character error in a lengthy control signal.

What can we do about the problem of errors in transmission?

One solution is to transmit the ASCII representation of each character some number $k > 1$ times. If $k = 2$ and the two transmitted values agree, we very likely have the correct value. If they disagree, we must ask the sender to try again. If $k = 3$ and the three transmitted values disagree, instead of asking for a retransmission, we can try to guess the answer by using a majority vote. For example, suppose we transmit three copies of 01010101, which we receive as 01010001, 01110100 and 11010101. A majority vote on each digit gives us the correct answer. This is known as an *error correcting code*. Of course, this method can fail. If we had received 01010001, 01110100 and 11010100 we would get the eighth digit wrong. We can increase our chances of getting the correct answer by increasing k, the number of repetitions.

There are better error correcting codes—they allow us to send shorter strings and still be at least as likely to be able to correct errors.

The basic idea is that we want to represent each of our characters by an n-string of zeroes and ones in such a way that if $a_1 a_2 \ldots a_n$ represents one character and $b_1 b_2 \ldots b_n$ represents another, then we often have $a_i \neq b_i$. Why is this good? It will help our discussion if we have some notation. Let A be the set of characters we are interested in and let f be a function that assigns to each $a \in A$ the n-string that will be used to represent a; i.e., $f(a) \in \{0,1\}^n$.

For $s, t \in \{0,1\}^n$, let $d(s,t)$ be the number of positions in which s differs from t. For example, if $a = 001001$ and $b = 000101$, then $d(a,b) = 2$. Finally, let $d(f)$ be the minimum of $d(f(x), f(y))$ over all $x \neq y$ in A. Whenever r and t differ in a position, either r and s differ in that position or s and t differ in that position. Thus

$$d(r,t) \;\leq\; d(r,s) + d(s,t). \tag{1.6}$$

We cannot replace the inequality with an equality, because r and t may agree in a position but both may differ from s in that position.

Suppose that $d(f) = 2$, that we transmit $f(x)$ and that a single zero-one bit is changed by static so that we receive s. We claim that we can tell an error has been made. If we can't tell, it must be because $f(y) = s$ for some $y \in A$. This is impossible because it would imply that $d(x,y) = 1$ and so $d(f) \leq 1$.

We can do more. Suppose that $d(f) = 3$, that we transmit $f(x)$ and that a single zero-one bit is changed by static so that we receive s. We claim that x is the only $y \in A$ such that $d(f(y), s) < 2$. In other words, x is the only character whose "encoding" is less than two errors away from s. Why is this? By (1.6) and the definition of $d(f)$, if $y \in A$ and $y \neq x$, then

$$3 \;=\; d(f) \;\leq\; d(f(x), f(y)) \;\leq\; d(f(x), s) + d(s, f(y)) \;=\; 1 + d(s, f(y)).$$

Thus $d(s, f(y)) \geq 2$.

More generally, if $d(f) \geq 2k+1$ and $s \in \{0,1\}^n$, there is at most one $x \in A$ with $d(f(x), s) \leq k$. Thus, if we assume that at most k errors have been made, we can recover the value of x. Given that s is received, one wants to find an $x \in A$ so that $d(f(x), s)$ is a minimum. This is called "decoding." Decoding efficiently is a difficult problem that we will not study.

Suppose we want $d(f) \geq 2k+1$, how large must n be? First we study lower bounds on n and then we study upper bounds.

Example 1.24 A lower bound on codeword length Here's the idea for finding a lower bound. Let $N(x)$ be the set of all n-strings s such that $d(f(x), s) \leq k$, where d and f are as defined in the preceding paragraphs. Later we will prove that $|N(x)|$ does not depend on x. Let $N = |N(x)|$. Suppose that $x \neq y \in A$. We will prove that $N(x) \cap N(y) = \emptyset$, the empty set. The number of n-strings must therefore be at least $N\,|A|$; however, there are 2^n n-strings and so $N\,|A| \leq 2^n$.

We now prove that $N(x) \cap N(y) = \emptyset$ using proof by contradiction. Suppose $s \in N(x) \cap N(y)$. Then $d(f(x), s) \leq k$ and $d(s, f(y)) \leq k$. By (1.6), $d(f(x), f(y)) \leq 2k$, contradicting $d(f) \geq 2k+1$.

We now compute $|N(x)|$ by noting that $s \in N(x)$ if and only if it differs from $f(x)$ in exactly j positions for some $j \leq k$. There are $\binom{n}{j}$ ways to select the j positions that must be changed. Thus

$$|N(x)| \;=\; \sum_{j=0}^{k} \binom{n}{j}.$$

Incidentally, this proves that $|N(x)|$ does not depend on x.

Dividing our inequality $N\,|A| \leq 2^n$ by N and substituting our formula for $N = |N(x)|$, we obtain

$$|A| \leq \frac{2^n}{\sum_{j=0}^{k} \binom{n}{j}}. \tag{1.7}$$

The smallest n for which (1.7) is true is a lower bound on how long the strings must be. Here are the lower bounds that are obtained for $k = 1$ and 2.

$\lvert A\rvert$	2	3	4	5	10	20
$k = 1$	3	4	5	5	7	9
$k = 2$	5	7	7	8	9	11

For example, if we have 20 characters and want to be able to correct strings that contain at most 2 errors, then the string length will have to be at least 11 (and possibly larger since this is only a lower bound).

The bound we obtained is called the "sphere packing bound" because $N(x)$ is thought of as a type of sphere with center x and radius k. □

We've shown that, if there is a code for A that corrects up to k errors, then the length n of the codewords must be so large that (1.7) holds. Now we want a result in the other direction; that is, we want an inequality such that, if n satisfies it, then there must be a code for A that corrects up to k errors. In other words, we want to find an upper bound on how large n must be. There are at least two ways to obtain such a result. One is to actually construct a code. Another is to show that among all possible codes for A having words of length n, at least one must be able to correct k and fewer errors. We'll take the second approach and use a probabilistic argument.

Example 1.25 An upper bound on codeword length We begin by constructing a probability space $(S, \Pr)$. Let S be all possible subsets of size $|A|$ of $\{0,1\}^n$. In other words, S consists of all $\binom{2^n}{|A|}$ possible sets of codewords. To make a subset into a code, we simply associate an element of the alphabet A with each element of the subset. Let Pr be the uniform distribution on S. Thus the elementary events are subsets which are potential codes; that is the $|A|$-subsets of $\{0,1\}^n$. A subset $C \in S$ will be *good* if every pair of its n-strings are at least distance $2k+1$ apart. (We've use C to remind us that the subset is a potential code.) Then assigning letters to n-strings in C will give us a code for A that corrects k and fewer errors. We want to find an upper bound on n. This will be an inequality on n such that S will contain at least one good subset if n satisfies the inequality.

Here is a method that is often used to obtain such inequalities. Let the random variable X be the number of pairs of bad n-strings in an randomly chosen C. Thus C is good whenever $X(C) = 0$. Since X must be a nonnegative integer, the expectation of X is

$$\mathbf{E}(X) \;=\; \sum_{k=0}^{\infty} k \Pr(X=k) \;\geq\; \Pr(X>0).$$

If we can prove that $\mathbf{E}(X) < 1$, we will have $\Pr(X>0) \;<\; 1$ and so $\Pr(X=0) \;>\; 0$. Since $X(C) = 0$ means C is good, there must be a good C. We'll evaluate $\mathbf{E}(X)$ in a minute, but first, what is the general method? Here it is.

- Introduce a probability space $(S, \Pr)$.
- Introduce a random variable X such that
  - the values of X are nonnegative integers,
  - if $X(s) = 0$, then $s \in S$ has the property we want.
- Find conditions such that $\mathbf{E}(X) < 1$.

We'll now do the last step, showing that $\mathbf{E}(X) < 1$.

Since our probability is uniform and since each $C \in S$ contains $\binom{|A|}{2}$ pairs of n-strings,

the expected value of X,
which is
the number of pairs of n-strings in a random C which are too close
equals
$\binom{|A|}{2}$ times the probability that two random n-strings are too close
which equals
$\binom{|A|}{2}$ times the probability that a new random n-strings is too close to a given one.

As in the preceding example, we want the number of n-strings within distance $2k$ of a given string, not counting the given string. You should be able to show that this equals $\sum_{i=1}^{2k} \binom{n}{i}$. Since there are

$2^n - 1$ strings other than the given one, we have

$$\mathbf{E}(X) \;=\; \binom{|A|}{2} \frac{\sum_{i=1}^{2k} \binom{n}{i}}{2^n - 1}.$$

We wanted this to be less than one. In other words, given $|A|$ and k, there will be a k-error correcting code with $|A|$ codewords if n is so large that

$$1 \;>\; \binom{|A|}{2} \frac{1}{2^n - 1} \sum_{i=1}^{2k} \binom{n}{i}. \qquad 1.8$$

Here is a table of the smallest values of n that satisfy the inequality for some values of $|A|$ and two values of k.

$\lvert A\rvert$	2	3	4	5	10	20
$k = 1$	3	7	8	9	12	15
$k = 2$	5	11	13	14	18	21

As you can see, these upper bounds are quite a bit larger than the lower bounds in the preceding example. □

One approach to creating a code would be to choose n so that the right side of (1.8) is not too close to 1. For example, say it equals 0.7. Since this number is $\mathbf{E}(X)$ which we saw was an upper bound on the probability that a random code is bad, there is a 30% probability that a randomly chosen element of S will be good. After a few random tries, we should be able to find a code. This seems to be an easy way to construct error correcting codes. It is—but they're no good! Why is this? With a random set of codewords, there is no problem encoding our message for transmission; however, if $|A|$ is large, it will be quite difficult to decode. To make decoding easy, one needs to construct a code that has some nice structure that one can use. This need has led to a considerable amount of research and to texts on the subject.

Exercises

1.3.1. Suppose that k and $n - k$ both get large as n gets large. Use Stirling's formula to show that

$$\binom{n}{k} \sim \frac{1}{\sqrt{2n\pi\lambda(1-\lambda)}} \left(\frac{1}{\lambda^\lambda (1-\lambda)^{1-\lambda}}\right)^n \quad \text{where } \lambda = k/n.$$

1.3.2. Suppose we have an election between two candidates and the ballots are counted one-by-one. At the end, the candidates are tied with n votes each. If the order of the votes is random, what is the probability that one of the candidates was never behind in the counting?
Hint. See Example 1.13.

1.3.3. How many 6 card hands contain 3 pairs?

1.3.4. How many ways can a 5 card hand containing 2 pairs be dealt? In other words, the *order* in which a person gets her cards matters.

1.3.5. How many 5 card hands contain a straight? A straight is 5 consecutive cards from the sequence A,2,3,4,5,6,7,8,9,10,J,Q,K,A without regard to suit.

1.3.6. How many compositions of n are there that have exactly k parts? The composition 1,2,2 of 5 has 3 parts.
Hint. See Exercise 1.1.4.

1.3.7. How many rearrangements of the letters in EXERCISES are there? How many arrangements of eight letters can be formed using the letters in EXERCISES? (No letter may be used more frequently than it appears in EXERCISES.)

1.3.8. In some card games only the values of the cards matter and their suits are irrelevant. Thus there are effectively only 13 distinct cards. How many different ways can a deck of cards be arranged in this case? The answer is a multinomial coefficient.

1.3.9. Return to choosing teams (Example 1.21). Suppose half the people are women and half are men, that each team must be as nearly evenly split as possible, and that there is one referee of each sex. How many ways can this be done?

1.3.10. There is an empire in the far away galaxy we've been visiting. They use the same alphabet (A,I,L,S,T) but their names consist of seven letters. Each name begins and ends with a consonant, contains no adjacent vowels and never contains three adjacent consonants. As before, if two consonants are adjacent, they cannot be the same.

(a) List the first 4 names in dictionary order.

(b) List the last 4 names in dictionary order.

(c) What are the first 4 names in dictionary order with just 2 vowels?

(d) How many names are possible?

*1.3.11. (*Multinomial Theorem*) Prove that the coefficient of $y_1^{m_1} y_2^{m_2} \cdots y_k^{m_k}$ in $(y_1 + y_2 + \cdots + y_k)^n$ is the multinomial coefficient $n!/m_1!m_2! \cdots m_k!$ when $n = m_1 + \cdots + m_k$ and zero otherwise.
Hint. Write

$$(y_1 + y_2 + \cdots + y_k)^n = \big((y_1 + y_2 + \cdots + y_{k-1}) + y_k\big)^n.$$

Now use the Binomial Theorem (Theorem 1.7) and induction on k.

1.3.12. Prove the following.

(a) $\binom{n}{k} = \binom{n}{n-k}$;

(b) $\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = 2^n$;

(c) $\binom{n}{0} - \binom{n}{1} + \cdots \pm \binom{n}{n} = 0 \quad \text{for} \quad n \geq 1$ (the signs alternate);

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

1.4 Recursions

Let's explore yet another approach to evaluating the binomial coefficient $C(n,k)$. As in the previous section, let $S = \{x_1, \ldots, x_n\}$. We'll think of $C(n,k)$ as counting k-subsets of S. Either the element x_n is in our subset or it is not. The cases where it is in the subset are all formed by taking the various $(k-1)$-subsets of $S - \{x_n\}$ and adding x_n to them. The cases where it is not in the subset are all formed by taking the various k-subsets of $S - \{x_n\}$. What we've done is describe how to build k-subsets of S from certain subsets of $S - \{x_n\}$. Since this gives each subset exactly once,

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

by the Rule of Sum.

The equation $C(n,k) = C(n-1,k-1) + C(n-1,k)$ is called a *recursion* because it tells how to compute $C(n,k)$ from values of the function with smaller arguments. This is a common approach which we can state in general form as follows.

Technique. Deriving recursions *Answering the question "How can I construct the things I want to count by using the same type of things of a smaller size?" usually gives a recursion.*

Sometimes it is easier to answer the question "How can I break the things I want to count up into smaller things of the same type?" This usually gives a recursion when it is turned around to answer the previous question.

Let's see how the second approach works for subsets. Given our collection of k-element subsets of S, throw out x_n if x_n is present. We obtain some $(k-1)$-element subsets of $S - \{x_n\}$ and some k-element subsets of $S - \{x_n\}$. In fact, you should be able to see that we obtain *all* $(k-1)$-element subsets and *all* k-element subsets exactly once. Turning this around gives us a way to build up k-element subsets of S.

We can use a recursion to compute a table of values by starting at the first row and computing new entries by adding previous ones. The arrows in Figure 1.3 show how this is done for the binomial coefficients. If the labels in this table are dropped, the rows are shifted slightly and a single 1 is added to the top row, then we obtain what is called *Pascal's triangle*. (See the figure.)

Actually, we've cheated a bit in all of this because the recursion only works when we have some values to start with. The correct statement of the recursion is either

$$\begin{aligned} C(0,0) &= 1, \\ C(0,k) &= 0 \quad \text{for } k \neq 0 \quad \text{and} \\ C(n,k) &= C(n-1,k-1) + C(n-1,k) \quad \text{for } n > 0; \end{aligned}$$

or

$$\begin{aligned} C(1,0) &= C(1,1) = 1, \\ C(1,k) &= 0 \quad \text{for } k \neq 0,1 \quad \text{and} \\ C(n,k) &= C(n-1,k-1) + C(n-1,k) \quad \text{for } n > 1; \end{aligned}$$

depending on whether we want to start with the row of Pascal's triangle consisting of 1 alone or the row consisting of 1,1. These starting values are called *initial conditions*. Note that, in either case, the last two conditions guarantee that $C(n,k) = 0$ for all $k < 0$.

Values of k

		0	1	2	3	4	5	6
	0	1	0	0	0	0	0	0
V	1	1	1	0	0	0	0	0
a		↓ ↘	↓ ↘	↓ ↘	↓ ↘	↓ ↘	↓ ↘	↓
l	2	1	2	1	0	0	0	0
u		↓ ↘	↓ ↘	↓ ↘	↓ ↘	↓ ↘	↓ ↘	↓
e	3	1	3	3	1	0	0	0
s		↓ ↘	↓ ↘	↓ ↘	↓ ↘	↓ ↘	↓ ↘	↓
	4	1	4	6	4	1	0	0
o		↓ ↘	↓ ↘	↓ ↘	↓ ↘	↓ ↘	↓ ↘	↓
f	5	1	5	10	10	5	1	0
		↓ ↘	↓ ↘	↓ ↘	↓ ↘	↓ ↘	↓ ↘	↓
n	6	1	6	15	20	15	6	1

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1

Figure 1.3 Left: The binomial coefficients are computed recursively. The columns of zeroes for $k < 0$ are omitted. Right: The results are arranged to give Pascal's triangle.

Example 1.26 Alternating subsets Let t_n be the number of subsets of $\{1, 2, \ldots, n\}$ such that, when the elements of the subset are listed in increasing order, the first is odd, the second is even, the third is odd, and so forth. We will allow the empty subset. Thus $t_0 = 1$ and $t_1 = 2$ because of $\emptyset$ and $\{1\}$. When $n = 4$ the subsets are

$$\emptyset \quad \{1\} \quad \{1,2\} \quad \{1,2,3\} \quad \{1,2,3,4\} \quad \{1,4\} \quad \{3\} \quad \{3,4\},$$

and so $t_4 = 8$. Throwing out the subsets containing 4, we see that $t_3 = 5$. Throwing out those containing 3 or 4, we see that $t_2 = 3$.

How can we get a recursion for t_n? We can't simply take an acceptable subset for $n-1$ and either add n to it or not. Why? For example, adding 4 to the subset $\{1,2\}$ counted by t_3 would give $\{1,2,4\}$, which is not allowed. Of course, not adding an element is always safe. In other words, *every subset counted by t_n that does not contain n is counted by t_{n-1} and conversely.* If we can somehow figure a way to get the subsets counted by t_n that contain n, we'll be done.

Let's look again at the subsets for t_4 that contain 4. There are three of them:

$$\{1,2,3,4\}, \quad \{1,4\} \quad \text{and} \quad \{3,4\}.$$

Can we somehow reduce this to one or more groups of alternating subsets with $n < 4$? Since $t_2 = 3$, this might be a good place to start. To reduce our list to subsets counted by t_2, we'll need to throw out 3 and 4:

$$\{1,2\}, \quad \{1\} \quad \text{and} \quad \emptyset.$$

We've got the subsets counted by t_2. That's good, but can we reverse the process? Yes. Add 4 to each subset. If the resulting subset is not alternating, add 3 to it, too, and the result will be alternating. That's it!

Let's state it in general. We build the subsets counted by t_n in two ways.

(a) Take a subset counted by t_{n-1}.

(b) Take a subset S counted by t_{n-2}. Exactly one of the following will give a new alternating subset

 (i) Add n to S.

 (ii) Add $n-1$ and n to S.

 In fact, if the largest element of S and n have the same parity (i.e., both odd or both even), we use (ii); if different parity, we use (i).

This requires $n \geq 2$ since we need to have $n-1 \geq 0$ and $n-2 \geq 0$ for (a) and (b) to make sense. You should be able to see that the procedure gives every alternating subset of $\{1, 2, \ldots, n\}$ exactly once. We've proved that

$$a_0 = 1, \quad a_1 = 2 \quad \text{and, for } n \geq 2, \quad a_n = a_{n-1} + a_{n-2}.$$

These are the *Fibonacci numbers*, which can be found in the index. □

Example 1.27 Set partitions A *partition of a set B* is a collection of nonempty subsets of B such that each element of B appears in exactly one subset. Each subset is called a *block* of the partition. The 15 partitions of $\{1, 2, 3, 4\}$ by number of blocks are

1 block:	$\{1,2,3,4\}$			
2 blocks:	$\{\{1,2,3\},\{4\}\}$	$\{\{1,2,4\},\{3\}\}$	$\{\{1,2\},\{3,4\}\}$	$\{\{1,3,4\},\{2\}\}$
	$\{\{1,3\},\{2,4\}\}$	$\{\{1,4\},\{2,3\}\}$	$\{\{1\},\{2,3,4\}\}$	
3 blocks:	$\{\{1,2\},\{3\},\{4\}\}$	$\{\{1,3\},\{2\},\{4\}\}$	$\{\{1,4\},\{2\},\{3\}\}$	$\{\{1\},\{2,3\},\{4\}\}$
	$\{\{1\},\{2,4\},\{3\}\}$	$\{\{1\},\{2\},\{3,4\}\}$		
4 blocks:	$\{\{1\},\{2\},\{3\},\{4\}\}$			

Let $S(n, k)$ be the number of partitions of an n-set having exactly k blocks. These are called *Stirling numbers of the second kind*.

> Do not confuse $S(n, k)$ with $C(n, k) = \binom{n}{k}$. In both cases we have an n-set. For $C(n, k)$ we want to *choose a subset* containing k elements and for $S(n, k)$ we want to *partition the set* into k blocks.

What is the value of $S(n, k)$? Let's try to get a recursion using the two questions in our technique.

How can we build partitions of $S = \{1, 2, \ldots, n\}$ with k blocks out of smaller cases? Using the approach we used for binomial coefficients, we'll take a partition of $S - \{n\}$ and add n to it somehow to get a k-block partition of S. If we take partitions of $\{1, 2, \ldots, n-1\}$ with $k-1$ blocks, we can simply add the block $\{n\}$. If we take partitions of $\{1, 2, \ldots, n-1\}$ with k blocks, we can add the element n to one of the k blocks. You should convince yourself that all k block partitions of $\{1, 2, \ldots, n\}$ arise in exactly one way when we do this. This gives us a recursion for $S(n, k)$. Putting n in a block by itself contributes $S(n-1, k-1)$. Putting n in a block with other elements contributes $S(n-1, k) \times k$ by the Rule of Product. By the Rule of Sum

$$S(n, k) \;=\; S(n-1, k-1) + k\, S(n-1, k). \qquad 1.9$$

We leave it to you to determine the values of n and k for which this is valid and to determine the initial conditions. You can construct the analog of Figure 1.3 as an exercise.

Now let's take the second question approach: How can we tear down a set partition into something smaller. As we did with subsets, we can simply remove n from our partition of $\{1, 2, \ldots, n\}$. You should convince yourself that this gives (1.9). There is another approach to tearing down: Instead of simply throwing out n, we can throw out the entire block containing n. If there are j elements in that block, throwing it out gives us a partition of an $(n-j)$-subset of $\{1, 2, \ldots, n-1\}$ into $k-1$ blocks. This gives all such partitions exactly once. Since there are $\binom{n-1}{n-j}$ ways to choose the subset, we have

$$S(n, k) \;=\; \sum_{j=1}^{n} \binom{n-1}{n-j} S(n-j, k-1) \quad \text{for } k > 1. \qquad 1.10$$

The initial conditions are $S(n, 1) = 1$ for $n \geq 1$ and $S(n, 1) = 0$ for $n \leq 0$.

At this point you may well expect us to come up with an explicit formula for $S(n, k)$ by a direct counting argument or a generating function argument since we did both for $C(n, k)$. These can both be done; however, more tools are required. They are developed in later chapters. Explicit formulas for $S(n, k)$ are not as nice as $C(n, k) = \frac{n!}{k!\,(n-k)!}$ since the simplest formula for $S(n, k)$ involves summation. □

So far all we've done is find recursions for various numbers and use the recursions to construct values. This is not the only way recursions can be used. Here are some others:

- Prove a formula, usually by induction: We'll see an example of this in a minute.
- Discover that two sets of numbers are the same because they have the same recursion. (Remember to include the initial conditions!)
- Study the numbers by looking directly at the recursion or by using generating functions: More on this in Part IV.

To illustrate a proof by induction, let's do Exercise 1.3.12(b), namely $\sum_{k=0}^{n}\binom{n}{k} = 2^n$ when $n \geq 0$. It's easy to check it for $n = 0$. Suppose $n > 0$ and the result is true for all values less than n. By the recursion

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

Since the terms $\binom{n-1}{-1}$ and $\binom{n-1}{n}$ are zero, each of the last two sums is 2^{n-1} by the induction hypothesis and we are done since $2^{n-1}+2^{n-1}=2^n$.

Exercises

1.4.1. Calculate the next two rows in Pascal's Triangle.

1.4.2. Equation (1.9) gives a recursion for $S(n,k)$, but it is incomplete: initial conditions and the values of n and k for which it holds were omitted. Determine the values of n and k for which it is valid. Determine the initial conditions. Construct a table of values for $S(n,k)$ up through $n = 5$.

1.4.3. Derive a recursion like $S(n,k) = S(n-1,k-1) + kS(n-1,k)$ for ordered k-lists without repetitions that can be made from an n-set. Derive the recursion using an argument like that for $S(n,k)$; *do not* get the recursion using the formula $n!/(n-k)!$ that we found earlier. Since "like" is rather vague, there can be more than one solution to this exercise.

1.4.4. Exercise 1.3.12(c) you were asked to prove

$$\sum_{k=0}^{n}(-1)^k\binom{n}{k} = 0 \quad \text{for} \quad n \geq 1.$$

Prove it by induction on n using the recursion $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$.

1.4.5. For $n > 0$, prove the following formulas for $S(n,k)$.

$$S(n,n)=1 \qquad S(n,n-1)=\binom{n}{2} \qquad S(n,1)=1 \qquad S(n,2)=(2^n-2)/2$$

1.4.6. How can the initial conditions be set up to make (1.10) true for $n \geq 1$?

1.4.7. "Marking" something can help us derive a recursion. How many ways can we construct a k-subset of $\{1,2,\ldots,n\}$ and mark an element in the subset? You can do this in two ways:

- choose the subset and mark the element or
- choose the marked element and then choose the rest of the subset.

By counting these two ways, obtain the recursion $\binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1}$ for $k > 0$.

1.4.8. Let B_n be the total number of partitions of an n element set. Thus

$$B_n = S(n,0) + S(n,1) + \cdots + S(n,n).$$

(a) Prove that

$$B_{n+1} = \sum_{i=0}^{n} \binom{n}{i} B_{n-i},$$

where B_0 is defined to be 1.
Hint. Construct the block containing $n+1$ and then construct the rest of the partition.

(b) Calculate B_n for $n \le 5$.

*1.4.9. Return to Exercise 1.2.13 (p. 18). You should have done (a) and (d) previously. Now you should be able to do (b) and (c) and obtain a recursion for (e). (Later, we will see how to use the "Principle of Inclusion and Exclusion" to obtain another solution for (e).)

1.4.10. We want to count the number of n digit sequences that have no adjacent zeroes. The digits must be chosen from the set $\{0, 1, \ldots, d-1\}$. For example, with $d = 3$ and $n = 4$, the sequences 0,2,1,0 and 2,1,2,2 are valid but 1,0,0,2 and 1,3,2,3 are not. Let the number of such sequences be A_n. (The case $d = 2$ is called the *Fibonacci numbers*.)

(a) From an n-sequence, remove the last digit if it is nonzero and the last two digits if the last digit is zero. By reversing this process, describe a way to build up all acceptable sequences by adding elements one or two at a time.

(b) Use (a) to obtain a recursion of the form $A_n = aA_{n-1} + bA_{n-2}$. What are a and b? For what n is the recursion valid? What are the initial conditions?

(c) Compute A_n for $n \le 5$ when $d = 10$.

1.5 Multisets

Let $M(n,k)$ be the number of ways to choose k elements from an n-set when repetition is allowed and order doesn't matter. Will any of our three methods for handling $C(n,k)$ work for $M(n,k)$? Let's examine them.

- Imposing an order: The critical observation for our first method was that an unordered list can be ordered in $k!$ ways. This is not true if repetitions are allowed. To see this, note that the extreme case of k repetitions of one element has only one ordering.
- Using a recursion: We might be able to obtain a recursion, but we would still be faced with the problem of solving it.
- Using generating functions: To use the generating functions we have to allow for repetitions. This can be done very easily: Simply replace $(1 + x_i)$ in Example 1.14 (p. 19) with the infinite sum

  $$1 + x_i + x_i^2 + x_i^3 + \cdots,$$

  a geometric series which sums to $(1-x_i)^{-1}$. Why does this replacement work? When we studied $C(n,k)$ in Example 1.14, the two terms in the factor $1+x_i$ corresponded to not choosing the ith element OR choosing it, respectively. Now we need more terms: $x_i x_i$ for when the ith element is chosen to appear twice in our unordered list, $x_i x_i x_i$ for three appearances, and so forth. The distributive law still takes care of producing all possible combinations. As in Example 1.14, if we replace x_i by x for all i, the coefficient of x^k will be the number of multisets of size k. Thus $M(n,k)$ is the coefficient of x^k in $(1-x)^{-n}$. You should be able to use this fact and Taylor's Theorem to obtain $M(n,k) = (n+k-1)!/(n-1)!\,k!$. Thus

Theorem 1.8 Multiset formula *The number of k-multisets that can be made from an n-set is*

$$M(n,k) = \binom{n+k-1}{k}.$$

We can stop here since we have the answer; however, someone with an inquiring mind is likely not to. Such a person might ask "Why is $M(n,k)$ the number of ways to choose a k element subset of an $n-1+k$ element set?" Here "why" means an explanation that proves the two numbers are equal without actually counting. Posing and answering questions like this improve our understanding of a topic and improve our abilities to use the tools. We'll give one answer now. Another appears in the exercises.

Given a k-multiset of positive integers, list them in nondecreasing order, [2] say $a_1 \le a_2 \le \ldots \le a_k$. For each i, increase a_i by $i-1$ to obtain a new list. The new list consists of k distinct postive integers in increasing order. This sets up a one-to-one correspondence between multisets of positive integers and sets of positive integers.

What do the k-multisets formed from $\{1,2,\ldots,n\}$ correspond to? Since the largest element in the multiset is increased by $k-1$, each such multiset corresponds to a k-subset of $T = \{1,2,\ldots,n+k-1\}$. Conversely, every k-subset X of T corresponds to such a multiset: Simply list the elements of X in increasing order and subtract $i-1$ from the ith element for each i.

We have proved that in our one-to-one correspondence the multisets counted by $M(n,k)$ correspond to the sets counted by $C(n+k-1,k)$. Thus, these two numbers must be equal.

Example 1.28 Balls in boxes We are given 4 labeled boxes each of which can hold 2 balls and are also given 4 identical red balls and 4 identical green balls. How many ways can the balls be placed in the boxes?

This is not a problem that fits into our multiset model easily, although it can be made to fit. Nevertheless, it is the sort of problem that our methods work for. Indeed, it is very similar to the card hand problems. We'll look at it as if we hadn't seen those problems to emphasize the need to be able to translate problem descriptions without needing to force them into particular frameworks.

To begin with, we observe that once the red balls have been placed into boxes, there is only one way to place the green balls. (This is because there are exactly as many positions available as there are balls.) Thus, we can simply focus on placement of the red balls. Since there aren't very many ways to do that, we could simply list all of them. There is another approach that requires less work: First do the problem with unlabeled boxes and then label them. The unlabeled solutions are simply partitions of the number 4 into 4 parts, with zeroes allowed and no part exceeding 2. (A *partition of a number* is an unordered list that sums to the number.) The solutions are

$$1+1+1+1 \quad 0+1+1+2 \quad \text{and} \quad 0+0+2+2.$$

The first of these can be labeled in one way. The second can be labeled in 12 ways: choose the label for the empty box (4 ways) and then the label for the box containing 2 red balls (3 ways). The third solution can be labeled in $\binom{4}{2} = 6$ ways. Thus, there are $1+12+6=19$ solutions to the original problem. □

---

[2] A sequence is in nondecreasing order if the elements do not decrease as we move along the sequence. It is in increasing order if the elements increase. Thus $-7,3,5,6$ is both increasing and nondecreasing, $3,4,4,6$ is nondecreasing but not increasing, and $3,5,6,4$ is neither.

Given a set S, forming a k-subset or a k-multiset from S are two extremes: for a k-subset, no element can be repeated; for a k-multiset elements can be repeated as much as desired (as long as the total equals k). If we want something between the extremes, the counting is more difficult. For example, there's no simple formula for the number of k-multisets if each element appears at most j times except for $j = 1$ and $j \geq k$.

Exercises

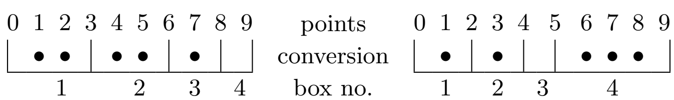
1.5.1. How many multisets can be formed from a set S if each element can appear at most j times? Your answer should be a simple formula.

1.5.2. It was stated in the preceding paragraph that "there's no simple formula for the number of k-multisets if each element appears at most j times except for $j = 1$ and $j \geq k$." What are the formulas for $j = 1$ and $j \geq k$?

1.5.3. Without using the formula for $M(n,k)$, prove that $M(n,k) = M(n-1,k) + M(n,k-1)$. What are the initial conditions for this recursion?

1.5.4. Prove that $M(n,k)$ is the number of ways to place k indistinguishable balls into n boxes.
Hint. If you have $n = 7$ boxes and $k = 8$ balls, the list 1,1,1,2,4,4,4,7 can be interpreted as "Place three balls in box 1, one ball in box 2, three balls in box 4 and one ball in box 7."

1.5.5. Imagine $\{1, 2, \ldots, n+k-1\}$ represented as points on a line in the usual way. Convert $n-1$ of the points to vertical bars and convert 0 and $n+k$ to vertical bars. Combine this with the previous problem to prove that $M(n,k) = C(n+k-1, n-1)$. This gives one answer the the question of "why" the two numbers are equal.
Hint. Here are examples of a correspondence with 5 balls and 4 boxes



1.5.6. Prove that the number of unordered k-lists made from n different items and using each item at most twice is the coefficient of x^k in $(1 + x + x^2)^n$. Generalize this.

1.5.7. Let $T(n,k)$ be the the number of k-multisets made from n different items, using each item at most twice in a multiset. Prove that

$$T(n,k) = T(n-1,k) + T(n-1,k-1) + T(n-1,k-2).$$

Relate this problem to the previous exercise and generalize it.

1.5.8. Prove by induction on n and k that the number of k-multisets that can be formed from an n-set is $\binom{n+k-1}{k}$. Let the answer be $M(n,k)$. To start the induction, verify the formula for $M(1,k)$ and for $M(n,1)$ for all n and k. For the induction step, use $M(n,k-1)$ and $M(n-1,k)$ to derive $M(n,k)$.

1.5.9. Let $f(b,t)$ be the number of ways to put b labeled balls into t labeled tubes. When balls are put into tubes the order matters: because the diameter of the tube is only slightly larger than that of the balls, the balls end up stacked on top of each other in a tube.

(a) Prove by induction on b that $f(b,t) = t(t+1)\cdots(t+b-1)$. (To do this, you will first need a recursion for $f(b,t)$.)
Hint. There are at least two ways to get a recursion on b: (i) insert $b-1$ balls and then the last or (ii) insert the first ball and then the remaining $b-1$.

(b) Give a noninductive combinatorial proof of the formula for $f(b,t)$.

*1.5.10. Let $f(n,k)$ be the number of ways to partition an n-set into k nonempty blocks where the order of the entries in a block matters but the order of the blocks does not matter.

(a) Prove by induction that

$$f(n,k) = \frac{n!}{k!}\binom{n-1}{k-1}.$$

Hint. An argument like the one leading to (1.9) can be used for the induction step.

(b) Give a noninductive combinatorial proof of the formula for $f(n,k)$.

Notes and References

We conclude this chapter with a table of the numbers of each of the four basic types of lists; i.e., ordered and unordered with repetitions allowed or forbidden. It is given in Figure 1.4. We are selecting k things from an n-set. The rules governing the selections are listed at the top and the left of the figure. As indicated in the figure, these numbers can also be interpreted in terms of placing balls into labeled boxes.

The concepts in the Rules of Sum and Product were known in ancient times. Until fairly recently, combinatorics has been synonymous with counting. This may be due to its connections with probability theory. You can learn about this in many books, but it is hard to do better than Feller's classic text [7]. We will focus on enumeration problems again in Chapters 4, 10 and 11.

Enumeration is still an active area of research in combinatorics. Although much of the research uses more sophisticated tools (See the notes to Chapters 4, 10 and 11.), some current research relies only on clever elementary arguments. You may be able to find some papers of this sort by browsing through such combinatorial journals as the *Journal of Combinatorial Theory, Series A*, the *European Journal of Combinatorics*, and *The Electronic Journal of Combinatorics* (at `http://www.combinatorics.org`). Unfortunately, proofs are often given rather tersely and careless authors sometimes neglect to explain terminology. As examples of short papers that you may be able to read now, you may want to look at [6] and [10]. Lubell's proof, which was used in Example 1.22, appeared in [9].

Because of the fundamental importance of counting, it is discussed in almost every text whose title refers to combinatorics or discrete mathematics. A few of the texts with material around the level of this book are those by Biggs [2; Ch.3], Bogart [3; Chs.1, 2], Cohen [4; Chs.2, 4], Stanton and White [12; Ch.1] and Tucker [13; Ch.5]. More advanced treatments can be found in the books by Comtet [5], Goulden and Jackson [8] and Stanley [11]. Anderson [1] starts off with Lubell's proof of Sperner's Theorem (Example 1.22) and then continues with other topics related to subsets of sets. His text is an example of the breadth of combinatorics—it does not discuss enumeration and has practically no overlap with our text.

Many papers have been written on Catalan numbers. Stanley [11, v.2] lists sixty-six things counted by Catalan numbers in his Exercise 6.1.9 (pp.219–229) and gives a partial solution to the exercise (pp.256–265).

Derivations of Stirling's formula (Theorem 1.5 (p. 12)) can be found in many places, including Feller's text [7; II.9, VII.2].

1. Ian Anderson, *Combinatorics of Finite Sets*, Dover (2002).
2. Norman L. Biggs, *Discrete Mathematics*, 2nd ed., Oxford Univ. Press (2003).
3. Kenneth P. Bogart, *Introductory Combinatorics*, 3rd ed., Brooks/Cole (2000).
4. Daniel I.A. Cohen *Basic Techniques of Combinatorial Theory*, John Wiley (1978).
5. Louis Comtet, *Advanced Combinatorics: The Art of Finite and Infinite Expansions*, Reidel (1974).

	Ordered (labeled balls)	Unordered (unlabeled balls)
Repetitions Allowed	Lists with repetition n^k	Multisets $\frac{(n+k-1)!}{k!(n-1)!}$
Repetitions Forbidden $\left(\begin{array}{c}\text{at most one}\\ \text{ball per box}\end{array}\right)$	Lists of distinct elements $\frac{n!}{(n-k)!}$	Sets $\frac{n!}{k!(n-k)!}$

Figure 1.4 The four basic list enumerators for k-lists made from n-sets. They can also be interpreted as placing k balls (either labeled or unlabeled) into n labeled boxes. The ball and box interpretation is indicated parenthetically.

6. Paul Erdős and Joel Spencer, Monochromatic sumsets, *J. Combinatorial Theory, Series A* **50** (1989), 162–163.
7. William Feller, *An Introduction to Probability Theory and Its Applications*, 3rd ed., John Wiley (1968).
8. Ian P. Goulden and David M. Jackson, *Combinatorial Enumeration*, Dover (2004). Reprint of John Wiley edition (1983).
9. David Lubell, A short proof of Sperner's Lemma, *J. Combinatorial Theory* **1** (1966), 299.
10. Albert Nijenhuis and Herbert S. Wilf, A method and two algorithms on the theory of partitions, *J. Combinatorial Theory, Series A* **18** (1975), 219–222.
11. Richard P. Stanley, *Enumerative Combinatorics*, vols. 1 and 2, Cambridge Univ. Press (1999, 2001).
12. Dennis Stanton and Dennis White, *Constructive Combinatorics*, Springer-Verlag (1986).
13. Alan C. Tucker, *Applied Combinatorics*, 3rd ed., John Wiley (2001).