CHAPTER

THREE

Induction

The twin concepts of recursion and induction are fundamentally important in combinatorial mathematics and computer science. In this chapter, we give a number of examples of how recursive formulas arise naturally in combinatorial problems, and we explain how they can be used to make computations. We also introduce the Principle of Mathematical Induction and give several examples of how it is applied to prove combinatorial statements. Our treatment will include several code snippets that illustrate how functions are defined recursively in computer programs.

3.1 Introduction

A professor decides to liven up the next combinatorics class by giving a door prize. As students enter class (on time, because to be late is a bit insensitive to the rest of the class), they draw a ticket from a box. On each ticket, a positive integer has been printed. No information about the range of ticket numbers is given, although they are guaranteed to be distinct. The box of tickets was shaken, not stirred, before the drawing and the selection is done without looking inside the box.

After each student has selected a ticket, the professor announces that a cash prize of one dollar (this is a university, you know) will be awarded to the student holding the lowest numbered ticket—from among those drawn.

Must the prize be awarded? In other words, given a set of positive integers, in this case the set of ticket numbers chosen by the students, must there be a least one? More generally, is it true that in any set of positive integers, there is always a least one? What happens if there is an enrollment surge and there are infinitely many students in the class and each has a ticket?

3.2 The Positive Integers are Well Ordered

Most likely, you answered the questions posed above with an enthusiastic "yes", in part because you wanted the shot at the money, but more concretely because it seems so natural. But you may be surprised to learn that this is really a much more complex subject than you might think at first. In Appendix B, we discuss the development of the number systems starting from the Peano Postulates. Although we will not spend much class time on this topic, it is important to know that the positive integers come with "some assembly required." In particular, the basic operations of addition and multiplication don't come for free; instead they have to be defined.

As a by-product of this development, we get the following fundamentally important property of the set $\mathbb{N}$ of positive integers:

Well Ordered Property of the Positive Integers: Every non-empty set of positive integers has a least element.

An immediate consequence of the well ordered property is that the professor will indeed have to pay someone a dollar—even if there are infinitely many students in the class.

3.3 The Meaning of Statements

Have you ever taken standardized tests where they give you the first few terms of a sequence and then ask you for the next one? Here are some sample questions. In each case, see if you can determine a reasonable answer for the next term.

1. $2, 5, 8, 11, 14, 17, 20, 23, 26, \ldots$
2. $1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, \ldots$
3. $1, 2, 5, 14, 42, 132, 429, 1430, 4862, \ldots$
4. $2, 6, 12, 20, 30, 42, 56, 72, 90, 110, \ldots$
5. $2, 3, 6, 11, 18, 27, 38, 51, \ldots$

Pretty easy stuff! OK, now try the following somewhat more challenging sequence. Here, we'll give you a lot more terms and challenge you to find the next one.

$$1, 2, 3, 4, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 2, 3, 4, 5, 6, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 2, 3, 4, 5, 6, \ldots$$

Trust us when we say that we really have in mind something very concrete, and once it's explained, you'll agree that it's "obvious." But for now, it's far from it.

Here's another danger lurking around the corner when we encounter formulas like

$$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$$

What do the dots in this statement mean? In fact, let's consider a much simpler question. What is meant by the following expression:

$$1 + 2 + 3 + \cdots + 6$$

Are we talking about the sum of the first six positive integers, or are we talking about the sum of the first 19 terms from the more complicated challenge sequence given above? You are supposed to answer that you don't know, and that's the correct answer.

The point here is that without a clarifying comment or two, the notation $1 + 2 + 3 + \cdots + 6$ isn't precisely defined. Let's see how to make things right.

First, let $f : \mathbb{N} \longrightarrow \mathbb{N}$ be a function. Set

$$\sum_{i=1}^{1} f(i) = f(1)$$

and if $n > 1$, define

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

To see that these two statements imply that the expression $\sum_{i=1}^{n} f(i)$ is defined for all positive integers, simply apply the Well Ordered Property.

So if we want to talk about the sum of the first six positive integers, then we should write:

$$\sum_{i=1}^{6} i$$

Now it is clear that we are talking about a computation that yields 21 as an answer.

A second example: previously, we defined $n!$ by writing

$$n! = n \times n - 1 \times n - 2 \times \cdots \times 3 \times 2 \times 1$$

By this point, you should realize that there's a problem here. Multiplication, like addition, is a binary operation. And what do those dots mean? Here's a way to do the job more precisely. Define $n!$ to be 1 if $n = 1$. And when $n > 1$, set $n! = n(n-1)!$.

Definitions like these are called *recursive* definitions. They can be made with different starting points. For example, we could have set $n! = 1$ when $n = 0$, and when $n > 0$, set $n! = n(n-1)!$.

Here's a code snippet using the C-programming language:

```
int sumrecursive(int n) {
 if (n == 1) return 2;
 else return sumrecursive(n-1)+(n*n -2*n+3);
}
```

What is the value of `sumrecursive(4)`? Does it make sense to you to say that `sumrecursive(n)` is defined for all positive integers n? Did you recognize that this program provides a precise meaning to the expression:

$$2 + 3 + 6 + 11 + 18 + 27 + 38 + 51 + \cdots + (n^2 - 2n + 3)$$

3.4 Binomial Coefficients Revisited

The binomial coefficient $\binom{n}{k}$ was originally defined in terms of the factorial notation, and with our recursive definitions of the factorial notation, we also have a complete and legally correct definition of binomial coefficients. The following recursive formula provides an efficient computational scheme.

Let n and k be integers with $0 \le k \le n$. If $k = 0$ or $k = n$, set $\binom{n}{k} = 1$. If $0 < k < n$, set

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

This recursion has a natural combinatorial interpretation. Both sides count the number of k-element subsets of $\{1, 2, \ldots, n\}$, with the right-hand side first grouping them into those which contain the element n and then those which don't. The traditional form of displaying this recursion is shown in Figure 3.1. This pattern is called "Pascal's triangle."

3.5 Solving Combinatorial Problems Recursively

In this section, we present examples of combinatorial problems for which solutions can be computed recursively. In chapter 8, we return to these problems and obtain even more compact solutions. Our first problem is one discussed in our introductory chapter.

Example 3.1. A family of n lines is drawn in the plane with (1) each pair of lines crossing and (2) no three lines crossing in the same point. Let $r(n)$ denote the number of regions into which the plane is partitioned by these lines. Evidently, $r(1) = 2$, $r(2) = 4$, $r(3) = 7$ and $r(4) = 11$. To determine $r(n)$ for all positive integers, it is enough to note that $r(1) = 1$, and when $n > 1$, $r(n) = n + r(n-1)$. This formula follows from the observation that if we label the lines as $L_1, L_2, \ldots, L_n$, then the $n-1$ points on line L_n where it crosses the other lines in the family divide L_n into n segments, two of which are infinite. Each

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

1 7 21 35 35 21 7 1

1 8 28 56 70 56 28 8 1

FIGURE 3.1: PASCAL'S TRIANGLE

of these segments is associated with a region determined by the first $n-1$ lines that has now been subdivided into two, giving us n more regions than were determined by $n-1$ lines. Thus $r(5) = 5 + 11 = 16$, $r(6) = 6 + 16 = 22$ and $r(7) = 7 + 22 = 29$.

Even by hand, it wouldn't be all that much trouble to calculate $r(100)$. We could do it before lunch.

Example 3.2. A $2 \times n$ checkboard will be tiled with rectangles of size 2×1 and 1×2. Find a recursive formula for the number $t(n)$ of tilings. Clearly, $t(1) = 1$ and $t(2) = 2$. When $n > 2$, consider the rectangle that covers the square in the upper right corner. If it is vertical, then preceding it, we have a tiling of the first $n-1$ columns. If it is horizontal, then so is the rectangle immediately underneath it, and proceeding them is a tiling of the first $n-2$ columns. This shows that $t(n) = t(n-1) + t(n-2)$. In particular, $t(3) = 1 + 2 = 3$, $t(4) = 2 + 3 = 5$ and $t(5) = 3 + 5 = 8$.

Again, if compelled, we could get $t(100)$ by hand, and Maple could get $t(1000)$.

Example 3.3. Call a ternary string *good* if it never contains a 2 followed immediately by a 0; otherwise, call it *bad*. Let $g(n)$ be the number of good strings of length n. Obviously $g(1) = 3$, since all strings of length 1 are good. Also, $g(2) = 8$ since the only bad string of length 2 is $(2, 0)$. Now consider a value of n larger than 2.

Partition the set of good strings of length n into three parts, according to the last character. Good strings ending in 1 can be preceded by any good string of length $n-1$, so there are $g(n-1)$ such strings. The same applies for good strings ending in 2. For good strings ending in 0, however, we have to be more careful. We can precede the 0 by a good string of length $n-1$ provided that the string does not end in 2. There are $g(n-1)$ good strings of length $n-1$ and of these, exactly $g(n-2)$ end in a 2. Therefore there are $g(n-1) - g(n-2)$ good strings of length n that end in a 0. Hence the total number of good strings of length n satisfies the recursive formula $g(n) = 3g(n-1) - g(n-2)$. Thus $g(3) = 3 \cdot 8 - 3 = 21$ and $g(4) = 3 \cdot 21 - 8 = 55$.

Once more, $g(100)$ is doable, while even a modest computer can be coaxed into giving us $g(5000)$.

3.6 Finding Greatest Common Divisors

There is more meat than you might think to the following elementary theorem, which seems to simply state a fact that you've known since second grade.

Theorem 3.4 (Division Theorem)**.** *Let m and n be positive integers. Then there exist unique integers q and r so that*

$$m = q \cdot n + r \quad \textit{and} \quad 0 \leq r < n.$$

We call q the quotient *and r the* remainder*.*

Proof. We settle the claim for existence. The uniqueness part is just high-school algebra.

Fix a positive integer n. If $n = 1$, then we may take $q = m$ and $r = 0$, i.e., $m = m \cdot 1 + 0$. Thus we may suppose that $n > 1$. If $m = 1$, we may take $q = 0$ and $r = 1$, noting that $1 = 0 \cdot n + 1$ and $0 \leq r = 1 < n$. Now suppose that we have a positive integer $m > 1$. Find integers q and r with

$$m - 1 = q \cdot n + r \quad \text{and} \quad 0 \leq r < n.$$

Since $r < n$, we know that $r + 1 \leq n$. If $r + 1 < n$, then

$$m = q \cdot n + (r + 1) \quad \text{and} \quad 0 \leq r + 1 < n.$$

On the other hand, if $r + 1 = n$, then

$$m = q \cdot n + (r + 1) = nq + n = (q + 1)n = (q + 1)n + 0. \qquad \square$$

Recall that an integer n is a *divisor* of an integer m if there is an integer q such that $m = qn$. (We write $n|m$ and read "n divides m".) An integer d is a *common divisor* of integers m and n if d is a divisor of both m and n. The *greatest common divisor* of m and n, written $\gcd(m, n)$, is the largest of all the common divisors of m and n.

Here's a particularly elegant application of the preceding basic theorem:

Theorem 3.5 (Euclidean Algorithm)**.** *Let m, n be positive integers and let q and r be the unique integers for which*

$$m = q \cdot n + r \quad \textit{and} \quad 0 \leq r < n.$$

If $r > 0$, then $\gcd(m, n) = \gcd(n, r)$.

Proof. Consider the expression $m = q \cdot n + r$. If a number d is a divisor of m and n, then it must also divide r. Similarly, if d is a divisor of n and r, it must also divide m. $\square$

Here is a code snippet that computes the greatest common divisor of m and n when m and n are positive integers with $m \geq n$. We use the familiar notation $m\%n$ to denote the remainder r in the expression $m = q \cdot n + r$.

```
int gcd(int m, int n) {
 if (m%n == 0) return n;
 else return gcd(n, m%n);
}
```

The disadvantage of this approach is the somewhat wasteful use of memory due to recursive function calls. On our course website, you will find code for computing the greatest common divisor of m and n using only a loop, i.e., there are no calls to memory. This code also solves the following diophantine equation problem.

Theorem 3.6. *Let m, n, and c be positive integers. Then there exist integers a and b, not necessarily non-negative, so that $am + bn = c$ if and only if c is a multiple of the greatest common divisor of m and n.*

Example 3.7. Find the greatest common divisor d of 3920 and 252 and find integers a and b such that $d = 3920a + 252b$.

In solving the problem, we demonstrate how to perform the Euclidean algorithm so that we can find a and b by working backward. First, we note that

$$3920 = 15 \cdot 252 + 140.$$

Now the Euclidean algorithm tells us that $\gcd(3920, 252) = \gcd(252, 140)$, so we write

$$252 = 1 \cdot 140 + 112.$$

Continuing, we have $140 = 1 \cdot 112 + 28$ and $112 = 4 \cdot 28 + 0$, so $d = 28$. Now we have

$$28 = 140 - 1 \cdot 112.$$

But we know that $112 = 252 - 1 \cdot 140$, so

$$28 = 140 - 1(252 - 1 \cdot 140) = 2 \cdot 140 - 1 \cdot 252.$$

Finally, $140 = 3920 - 15 \cdot 252$, so now we have

$$28 = 2(3920 - 15 \cdot 252) - 1 \cdot 252 = 2 \cdot 3920 - 31 \cdot 252.$$

Therefore $a = 2$ and $b = -31$.

3.7 Mathematical Induction

Now we move on the induction, the powerful twin of recursion.

Let n be a positive integer. Consider the following mathematical statements, each of which involve n:

1. $2n + 7 = 13$.
2. $3n - 5 = 9$.
3. $n^2 - 5n + 9 = 3$.
4. $8n - 3 < 48$.
5. $8n - 3 > 0$.
6. $(n + 3)(n + 2) = n^2 + 5n + 6$.
7. $n^2 - 6n + 13 \geq 0$.

Such statements are called *open* statements. Open statements can be considered as *equations*, i.e., statements that are valid for certain values of n. Statement 1 is valid only when $n = 3$. Statement 2 is never valid, i.e., it has no solutions among the positive integers. Statement 3 has exactly two solutions, and Statement 4 has six solutions. On the other hand, Statements 5, 6 and 7 are valid for all positive integers.

At this point, you are probably scratching your head, thinking that this discussion is trivial. But let's consider some statements that are a bit more complex.

1. The sum of the first n positive integers is $n(n + 1)/2$.
2. The sum of the first n odd positive integers is n^2.
3. $n^n \geq n! + 4,000,000,000n2^n$ when $n \geq 14$.

How can we establish the validity of such statements, provided of course that they are actually true? The starting point for providing an answer is the following property:

Principle of Mathematical Induction Let S_n be an open statement involving a positive integer n. If S_1 is true, and for every positive integer k, the statement S_{k+1} is true whenever S_k is true, then S_n is true for every positive integer n.

It is easy to see that the Principle of Mathematical Induction is logically equivalent to the Well Ordered Property of Positive Integers. If you haven't already done so, now might be a good time to look over Appendix A on set theory.

3.8 Inductive Definitions

Although it is primarily a matter of taste, recursive definitions can also be recast in an inductive setting. As a first example, set $1! = 1$ and whenever $k!$ has been defined, set $(k+1)! = (k+1)k!$.

As a second example, set

$$\sum_{i=1}^{1} f(i) = f(1) \quad \text{and} \quad \sum_{i=1}^{k+1} f(i) = \sum_{i=1}^{k} f(i) + f(k+1)$$

In this second example, we are already using an abbreviated form, as we have omitted some English phrases. But the meaning should be clear.

Now let's back up and give an example which would really be part of the development of number systems. Suppose you knew everything there was to know about the *addition* of positive integers but had never heard anything about *multiplication*. Here's how this operation can be defined.

Let m be a positive integer. Then set

$$m \cdot 1 = m \quad \text{and} \quad m \cdot (k+1) = m \cdot k + m$$

You should see that this *defines* multiplication but doesn't do anything in terms of establishing such familiar properties as the commutative and associative properties. Check out some of the details in Appendix B.

3.9 Proofs by Induction

No discussion of recursion and induction would be complete without some obligatory examples of proofs using induction. We start with the "Hello World" example.

Proposition 3.8. *For every positive integer n, the sum of the first n positive integers is $n(n+1)/2$, i.e.,*

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

Proof. We first prove the assertion when $n = 1$. For this value of n, the left hand side is just 1, while the right hand side evaluates to $1(1+1)/2 = 1$.

Now assume that for some positive integer k, the formula holds when $n = k$, i.e., assume that

$$\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$$

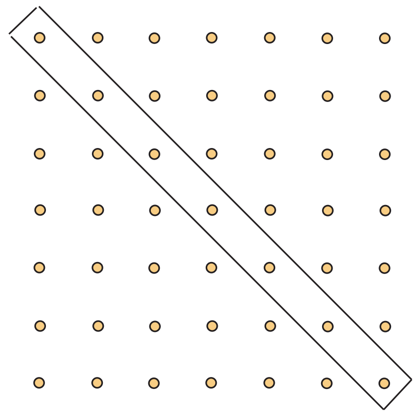

FIGURE 3.2: THE SUM OF THE FIRST n INTEGERS

Then it follows that

$$\sum_{i=1}^{k+1} i = \Big(\sum_{i=1}^{k} i\Big) + (k+1) = \frac{k(k+1)}{2} + (k+1) = \frac{k^2+3k+2}{2} = \frac{(k+1)(k+2)}{2}$$

Thus the formula also holds when $n = k+1$. By the Principle of Mathematical Induction, it holds for all positive integers n. $\square$

The preceding argument is 100% correct... but some combinatorial mathematicians would argue that it may actually hide what is really going on. Here's a much more concrete explanation for the formula.

Consider an $(n+1) \times (n+1)$ array of dots. There are $(n+1)^2$ dots altogether, with exactly $n+1$ on the main diagonal. As illustrated in Figure 3.2, the off-diagonal entries split naturally into two equal size parts, those above and those below the diagonal.

Furthermore, each of those two parts has $\sum_{i=1}^{n} i = 1+2+3+\cdots+n$ dots. It follows that

$$\sum_{i=1}^{n} i = 1+2+3+\cdots+n = \frac{(n+1)^2-(n+1)}{2}$$

and this is obvious! Now a little algebra on the right hand side of this expression produces the formula given earlier for the sum.

So to really understand an identity, you should be able both to give a formal proof by mathematical induction as well as give a combinatorial explanation of its meaning.

Here's a second example, also quite a classic.

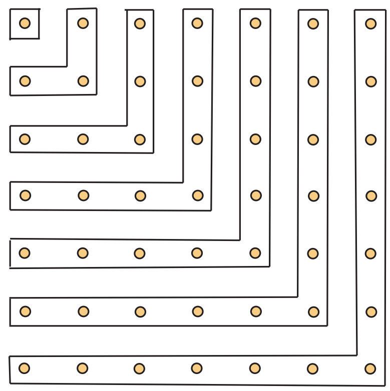

FIGURE 3.3: THE SUM OF THE FIRST n ODD INTEGERS

Proposition 3.9. *For each positive integer n, the sum of the first n odd positive integers is n^2, i.e.,*

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

Proof. First, that the formula holds when $n = 1$. Now suppose that k is a positive integer and that the formula holds when $n = k$, i.e., assume

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

Then

$$\sum_{i=1}^{k+1}(2i-1) = \big(\sum_{i=1}^{k} 2i-1\big) + 2k+1 = k^2 + (2k+1) = (k+1)^2. \qquad \square$$

Now for a combinatorial argument. As suggested in Figure 3.3, the sum of the first n odd positive integers is clearly equal to n^2.

Here is a much more general version of the first result in this section.

Proposition 3.10. *Let n and k be non-negative integers with $n \geq k$. Then*

$$\sum_{i=k}^{n}\binom{i}{k} = \binom{n+1}{k+1}.$$

Proof. Fix a non-negative integer k. We then prove the formula by induction on n. If $n = k$, note that the left hand side is just $\binom{k}{k} = 1$, while the right hand side is $\binom{k+1}{k+1}$ which is also 1. Now assume that m is a non-negative integer, with $m \geq k$, and that the formula holds when $n = m$, i.e., assume that

$$\sum_{i=k}^{m} \binom{i}{k} = \binom{m+1}{k+1}.$$

Then

$$\begin{aligned}\sum_{i=k}^{m+1} \binom{i}{k} &= \sum_{i=k}^{m} \binom{i}{k} + \binom{m+1}{k} \\ &= \binom{m+1}{k+1} + \binom{m+1}{k} \\ &= \binom{m+2}{k+1}.\end{aligned}$$

□

To make sure that we understand combinatorially what the preceding result is saying, note that both sides count the number of $k+1$ element subsets of $\{1, 2, 3, \ldots, n+1\}$ with the left hand side first grouping them according to the largest element.

3.10 Exercises

1. Find $d = \gcd(5544, 910)$ as well as integers a and b such that $5544a + 910b = d$.
2. Find $\gcd(827, 249)$ as well as integers a and b such that $827a + 249b = 6$.
3. Let a, b, m, and n be integers and suppose that $am + bn = 36$. What can you say about $\gcd(m, n)$?
4. Each of the following formulas admits an inductive proof. For each formula, give both a formal proof using the Principle of Mathematical Induction and a combinatorial proof.

   a) $1^2 + 2^2 + 3^2 + \cdots + n^2 = \dfrac{n(n+1)(2n+1)}{6}$

   b) $\binom{n}{0}\binom{n}{n} + \binom{n}{1}\binom{n}{n-1} + \binom{n}{2}\binom{n}{n-2} + \cdots + \binom{n}{n}\binom{n}{0} = \binom{2n}{n}$

   c) $\binom{n}{0}2^0 + \binom{n}{1}2^1 + \binom{n}{2}2^2 + \cdots + \binom{n}{n}2^n = 3^n$

5. Show that for all positive integers n,

$$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1.$$

6. Consider the recursion given by $f(1) = 1$, $f(2) = 1$, and $f(n) = f(n-1) + f(n-2)$ for $n \geq 3$. Show that $f(n)$ is divisible by 3 if and only if n is divisible by 4.

7. Give a recursion for the number $g(n)$ of ternary strings of length n that do not contain 102 as a substring. Be sure to give enough initial values to get the recursion going.

8. Let S be the set of strings on the alphabet $\{0, 1, 2, 3\}$ that do not contain 12 or 20 as a substring. Give a recursion for the number $h(n)$ of strings in S of length n. *Hint*: Check your recursion by manually computing $h(1)$, $h(2)$, $h(3)$, and $h(4)$.

9. Suppose that $x \in \mathbb{R}$ and $x > -1$. Prove that for all integers $n \geq 0$, $(1+x)^n \geq 1+nx$.