Mathematical Analysis for Computer Science
Part I – *Topics on Counting and Deterministic Analysis*

Edited/Compiled as Course Reader for CSE 4747

Jan-Aug 2021

Contents

Preface

This document is a compilation of study materials that form a big part of a course titled MATHEMATICAL ANALYSIS FOR COMPUTER SCIENCE. The topics covered in this part can be broadly categorized as *counting*. The rest of the course is concerned with *applied probability and stochastic processes.*

Counting plays a major role in much of the mathematical analyses undertaken within *computer science*. Determining the time and storage required to solve a computational problem—a central objective in computer science—often comes down to solving a counting problem. On the other hand, specialized topics such as password and encryption security counts on having a very large set of possible passwords and encryption keys. In addition, counting is the basis of probability theory, which plays a central role in all sciences, including computer science.

We begin our study of counting in Chapter 1 with a collection of rules and methods for finding closed-form expressions for commonly-occurring sums and products. We also review asymptotic notations such as $\sim$, O and Θ that are commonly used to express how a quantity such as the running time of a program grows with the size of the input. Special numbers often prop up in analysis of problems in computer science, and Chapter 2 presents an exposition on a few of the important and interesting items from the bunch.

Counting seems easy enough: 1, 2, 3, 4, etc. This direct approach works well for counting simple things—like your toes—and may be the only approach for extremely complicated things with no identifiable structure. However, subtler methods can help you count many things in the vast middle ground. Chapter 3 covers the most basic rules for determining the cardinality of a set. These rules characterize the fundamental structures of the analytic thinking required to address the counting problems that arise in sciences and engineering.

Chapter 4 shows how to solve a variety of recurrences that arise in computational problems. These methods are especially useful when you need to design or analyze recursive programs.

Finally, Chapter 5 introduces *generating functions* which allow many counting problems to be solved by simple algebraic formula simplification.

The materials are primarily drawn—in most cases verbatim—from different versions of the course reader titled “Mathematics for Computer Science” by Leighton, Meyer, and Lehman.[1] A few items are extracted from Graham, Knuth, and Patashnik,[2] which, in turn, laid the basis for a lot of the coverage in the former. We thank Leighton, Meyer, and Lehman[3] for releasing their materials with a Creative Commons License.

This particular version is a result of quite a bit of editing, occasional rewrites and compilation of the above materials. This process may have introduced inadvertent errors of its own – typographical or otherwise – in addition to those that may have been there from the source materials. Notifying any glitches as such via emails to `ashraf@cu.ac.bd` would be much appreciated.

[1]Tom Leighton, Albert Meyer, and Eric Lehman. *Mathematics for Computer Science.* MIT 6.042 Notes, 2017. URL: `https://courses.csail.mit.edu/6.042/fall17/mcs.pdf`.

[2]Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science.* 2nd ed. Addison-Wesley, 1994.

[3]Leighton, Meyer, and Lehman, *Mathematics for Computer Science.*

1 | Sums and Asymptotics

Sums and products arise regularly in the analysis of algorithms, financial applications, physical problems, and probabilistic systems. The following is an instance of a sum that you are supposed to have encountered as you studied *mathematical induction* in *discrete mathematics*

$$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}. \tag{1.1}$$

Of course, the left-hand sum could be expressed concisely as a subscripted summation

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

but the right-hand expression $n(n+1)/2$ is not only concise but also easier to evaluate. Furthermore, it more clearly reveals properties such as the growth rate of the sum. Expressions like $n(n+1)/2$ that do not make use of subscripted summations or products—or those handy but sometimes troublesome sequences of three dots—are called *closed forms.*

Another example is the closed form for a *geometric sum*

$$1 + x + x^2 + x^3 + \cdots + x^n = \frac{1 - x^{n+1}}{1 - x} \tag{1.2}$$

that you may also have proven and/or dealt with earlier. The sum as described on the left-hand side of (1.2) involves n additions and $1 + 2 + \cdots + (n-1) = (n-1)n/2$ multiplications, but its closed form on the right-hand side can be evaluated using fast exponentiation with at most $2 \log n$ multiplications, a division, and a couple of subtractions. Also, the closed form makes the growth and limiting behavior of the sum much more apparent.

Equations (1.1) and (1.2) were easy to verify by induction, but, as is often the case, the proofs by induction gave no hint about how these formulas were found in the first place. Finding them is part math and part art, which we'll start examining in this chapter.

Our first motivating example will be the value of a financial instrument known as an *annuity.* This value will be a large and nasty-looking sum. We will then describe several methods for finding closed forms for several sorts of sums, including those for annuities. In some cases, a closed form for a sum may not exist, and so we will provide a general method for finding closed forms for good upper and lower bounds on the sum.

The methods we develop for sums will also work for products, since any product can be converted into a sum by taking its logarithm. For instance, later in the chapter we will use this approach to find a good closed-form approximation to the *factorial function*

$$n! \triangleq 1 \cdot 2 \cdot 3 \cdots n.$$

We conclude the chapter with a discussion of asymptotic notation, especially "Big Oh" notation. Asymptotic notation is often used to bound the error terms when there is no exact closed form expression for a sum or product. It also provides a convenient way to express the growth rate or order of magnitude of a sum or product.

1.1 The Value of an Annuity

Would you prefer a million dollars today or $50,000 a year for the rest of your life? On the one hand, instant gratification is nice. On the other hand, the *total dollars* received at $50K per year is much larger if you live long enough.

Formally, this is a question about the value of an annuity. An *annuity* is a financial instrument that pays out a fixed amount of money at the beginning of every year for some specified number of years. In particular, an n-year, m-payment annuity pays m dollars at the start of each year for n years. In some cases, n is finite, but not always. Examples include lottery payouts, student loans, and home mortgages. There are even firms on Wall Street that specialize in trading annuities.[1]

A key question is, "What is an annuity worth?" For example, lotteries often pay out jackpots over many years. Intuitively, $50,000 a year for 20 years ought to be worth less than a million dollars right now. If you had all the cash right away, you could invest it and begin collecting returns on the investment. But what if the choice were between $50,000 a year for 20 years and a *half* million dollars today? Suddenly, it's not clear which option is better.

1.1.1 The Future Value of Money

In order to answer such questions, we need to know what a dollar paid out in the future is worth today. To model this, let's assume that money can be invested with an annual rate of return p on it on average. We'll assume an 8% rate[2] for the rest of the analysis and discussion, so $p = 0.08$.

Here is why the expected annual rate of return p matters. Ten dollars invested today at return rate p will become $(1+p)\cdot 10 = 10.80$ dollars in a year, $(1+p)^2 \cdot 10 \approx 11.66$ dollars in two years, and so forth. Looked at another way, ten dollars paid out a year from now is only really worth $1/(1+p)\cdot 10 \approx 9.26$ dollars today, because if we had the $9.26 today, we could invest it and would have $10.00 in a year anyway. Therefore, p determines the value of money paid out in the future.

So for an n-year, m-payment annuity, the first payment of m dollars is truly worth m dollars. But the second payment a year later is worth only $m/(1+p)$ dollars. Similarly, the third payment is worth $m/(1+p)^2$, and the n-th payment is worth only $m/(1+p)^{n-1}$. The total value V of the annuity is equal to the sum of the payment values. This gives:

$$\begin{aligned} V &= \sum_{i=1}^{n} \frac{m}{(1+p)^{i-1}} && (1.3) \\ &= m \cdot \sum_{j=0}^{n-1} \left(\frac{1}{1+p}\right)^j && \text{(substitute } j = i-1) \quad (1.4) \\ &= m \cdot \sum_{j=0}^{n-1} x^j && \text{(substitute } x = 1/(1+p)). \quad (1.5) \end{aligned}$$

The goal of the preceding substitutions was to get the summation into the form of a simple geometric sum. This leads us to an explanation of a way you could have discovered the closed form (1.2) in the first place using the *Perturbation Method*.

[1] Such trading ultimately led to the subprime mortgage disaster in 2008–2009. We'll probably talk more about that in a later chapter.

[2] Investment firms typically engage in *hedging*, that is, they diversify their investment in a number of areas so as to offset potential losses or gains that may be incurred by a companion investment. In a robust economy, an investment within a fund as such is poised to bring returns at a fairly steady rate, even if you are not stipulating a fixed rate prior. Importantly, as you seek to do the due diligence by assessing your risks in agreeing the investment contract, you would like to make projections based on an average of the prior annual rates of return. It is possible to form a more realistic yet complicated model by taking the rate of return as a random variable, but a first approximation from such a model would also typically require calculating with the expectation of the random variable in its place.

1.1.2 The Perturbation Method

Given a sum that has a nice structure, it is often useful to "perturb" the sum so that we can somehow combine the sum with the perturbation to get something much simpler. For example, suppose

$$S = 1 + x + x^2 + \cdots + x^n.$$

An example of a perturbation would be

$$xS = x + x^2 + \cdots + x^{n+1}.$$

The difference between S and xS is not so great, and so if we were to subtract xS from S, there would be massive cancellation:

$$\begin{aligned} S &= 1 + x + x^2 + x^3 + \cdots + x^n \\ -xS &= \quad\ - x - x^2 - x^3 - \cdots - x^n - x^{n+1}. \end{aligned}$$

The result of the subtraction is

$$S - xS = 1 - x^{n+1}.$$

Solving for S gives the desired closed-form expression in equation 1.2, namely,

$$S = \frac{1 - x^{n+1}}{1 - x}.$$

We'll see more examples of this method when we introduce *generating functions* (Chapter **??**) later in the course.

1.1.3 A Closed Form for the Annuity Value

Using equation 1.2, we can derive a simple formula for V, the value of an annuity that pays m dollars at the start of each year for n years.

$$\begin{aligned} V &= m\left(\frac{1 - x^n}{1 - x}\right) && \text{(by equations 1.5 and 1.2)} && (1.6) \\ &= m\left(\frac{1 + p - (1/(1+p))^{n-1}}{p}\right) && \text{(substituting } x = 1/(1+p)\text{).} && (1.7) \end{aligned}$$

Equation 1.7 is much easier to use than a summation with dozens of terms. For example, what is the real value of a winning lottery ticket that pays \$50,000 per year for 20 years? Plugging in $m = \$50{,}000$, $n = 20$ and $p = 0.08$ gives $V \approx \$530{,}180$. So because payments are deferred, the million dollar lottery is really only worth about a half million dollars! This is a good trick for the lottery advertisers.

1.1.4 Infinite Geometric Series

We began this chapter by asking whether you would prefer a million dollars today or \$50,000 a year for the rest of your life. Of course, this depends on how long you live, so optimistically assume that the second option is to receive \$50,000 a year *forever.* This sounds like infinite money! But we can compute the value of an annuity with an infinite number of payments by taking the limit of our geometric sum in equation 1.2 as n tends to infinity.

Theorem 1. *If* $|x| < 1$, *then*

$$\sum_{i=0}^{\infty} x^i = \frac{1}{1 - x}.$$

Proof.

$$\sum_{i=0}^{\infty} x^i \triangleq \lim_{n \to \infty} \sum_{i=0}^{n} x^i$$

$$
\begin{aligned}
&= \lim_{n\to\infty} \frac{1 - x^{n+1}}{1 - x} && \text{(by equation 1.2)} \\
&= \frac{1}{1-x}.
\end{aligned}
$$

The final line follows from the fact that $\lim_{n\to\infty} x^{n+1} = 0$ when $|x| < 1$. □

In our annuity problem $x = 1/(1+p) < 1$, so Theorem 1 applies, and we get

$$
\begin{aligned}
V &= m \cdot \sum_{j=0}^{\infty} x^j && \text{(by equation 1.5)} \\
&= m \cdot \frac{1}{1-x} && \text{(by Theorem 1)} \\
&= m \cdot \frac{1+p}{p} && (x = 1/(1+p)).
\end{aligned}
$$

Plugging in $m = \$50{,}000$ and $p = 0.08$, we see that the value V is only \$675,000. It seems amazing that a million dollars today is worth much more than \$50,000 paid every year for eternity! But on closer inspection, if we had a million dollars today invested in a fund that earns 8% on average, we could take out and spend \$80,000 a year, *forever.* So as it turns out, this answer really isn't so amazing after all.

1.1.5 Examples

Equation 1.2 and Theorem 1 are incredibly useful in computer science.

Here are some other common sums that can be put into closed form using equation 1.2 and Theorem 1:

$$1 + 1/2 + 1/4 + \cdots = \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i = \frac{1}{1 - (1/2)} = 2 \tag{1.8}$$

$$0.99999\cdots = 0.9 \sum_{i=0}^{\infty} \left(\frac{1}{10}\right)^i = 0.9 \left(\frac{1}{1 - 1/10}\right) = 0.9 \left(\frac{10}{9}\right) = 1 \tag{1.9}$$

$$1 - 1/2 + 1/4 - \cdots = \sum_{i=0}^{\infty} \left(\frac{-1}{2}\right)^i = \frac{1}{1 - (-1/2)} = \frac{2}{3} \tag{1.10}$$

$$1 + 2 + 4 + \cdots + 2^{n-1} = \sum_{i=0}^{n-1} 2^i = \frac{1 - 2^n}{1 - 2} = 2^n - 1 \tag{1.11}$$

$$1 + 3 + 9 + \cdots + 3^{n-1} = \sum_{i=0}^{n-1} 3^i = \frac{1 - 3^n}{1 - 3} = \frac{3^n - 1}{2} \tag{1.12}$$

If the terms in a geometric sum grow smaller, as in equation 1.8, then the sum is said to be *geometrically decreasing.* If the terms in a geometric sum grow progressively larger, as in equations 1.11 and 1.12, then the sum is said to be *geometrically increasing.* In either case, the sum is usually approximately equal to the term in the sum with the greatest absolute value. For example, in equations 1.8 and 1.10, the largest term is equal to 1 and the sums are 2 and $2/3$, both relatively close to 1. In equation 1.11, the sum is about twice the largest term. In equation 1.12, the largest term is 3^{n-1} and the sum is $(3^n - 1)/2$, which is only about a factor of 1.5 greater. You can see why this rule of thumb works by looking carefully at equation 1.2 and Theorem 1.

1.1.6 Variations of Geometric Sums

We now know all about geometric sums—if you have one, life is easy. But in practice one often encounters sums that cannot be transformed by simple variable substitutions to the form $\sum x^i$.

A non-obvious but useful way to obtain new summation formulas from old ones is by differentiating or integrating with respect to x. As an example, consider the following sum:

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

This is not a geometric sum. The ratio between successive terms is not fixed, and so our formula for the sum of a geometric sum cannot be directly applied. But differentiating equation 1.2 leads to:

$$\frac{d}{dx}\left(\sum_{i=0}^{n-1} x^i\right) = \frac{d}{dx}\left(\frac{1-x^n}{1-x}\right). \tag{1.13}$$

The left-hand side of equation 1.13 is simply

$$\sum_{i=0}^{n-1} \frac{d}{dx}(x^i) = \sum_{i=0}^{n-1} ix^{i-1}.$$

The right-hand side of equation 1.13 is

$$\begin{aligned}\frac{-nx^{n-1}(1-x) - (-1)(1-x^n)}{(1-x)^2} &= \frac{-nx^{n-1} + nx^n + 1 - x^n}{(1-x)^2}\\ &= \frac{1 - nx^{n-1} + (n-1)x^n}{(1-x)^2}.\end{aligned}$$

Hence, equation 1.13 means that

$$\sum_{i=0}^{n-1} ix^{i-1} = \frac{1 - nx^{n-1} + (n-1)x^n}{(1-x)^2}.$$

You would probably be asked in an exercise problem to show how the perturbation method could also be applied to derive this formula.

Often, differentiating or integrating messes up the exponent of x in every term. In this case, we now have a formula for a sum of the form $\sum ix^{i-1}$, but we want a formula for the series $\sum ix^i$. The solution is simple: multiply by x. This gives:

$$\sum_{i=1}^{n-1} ix^i = \frac{x - nx^n + (n-1)x^{n+1}}{(1-x)^2} \tag{1.14}$$

and we have the desired closed-form expression for our sum[3]. It seems a little complicated, but it's easier to work with than the sum.

Notice that if $|x| < 1$, then this series converges to a finite value even if there are infinitely many terms. Taking the limit of equation 1.14 as n tends to infinity gives the following theorem:

Theorem 2. *If* $|x| < 1$, *then*

$$\sum_{i=1}^{\infty} ix^i = \frac{x}{(1-x)^2}. \tag{1.15}$$

As a consequence, suppose that there is an annuity that pays im dollars at the end of each year i, forever. For example, if $m = \$50{,}000$, then the payouts are \$50,000 and then \$100,000 and then \$150,000 and so on. It is hard to believe that the value of this annuity is finite! But we can use Theorem 2 to compute the value:

$$V = \sum_{i=1}^{\infty} \frac{im}{(1+p)^i}$$

[3]Since we could easily have made a mistake in the calculation, it is always a good idea to go back and validate a formula obtained this way with a proof by induction.

$$\begin{aligned} &= m \cdot \frac{1/(1+p)}{(1-\frac{1}{1+p})^2} \\ &= m \cdot \frac{1+p}{p^2}. \end{aligned}$$

The second line follows by an application of Theorem 2. The third line is obtained by multiplying the numerator and denominator by $(1+p)^2$.

For example, if $m = \$50{,}000$, and $p = 0.08$ as usual, then the value of the annuity is $V = \$8{,}437{,}500$. Even though the payments increase every year, the increase is only additive with time; by contrast, dollars paid out in the future decrease in value exponentially with time. The geometric decrease swamps out the additive increase. Payments in the distant future are almost worthless, so the value of the annuity is finite.

The important thing to remember is the trick of taking the derivative (or integral) of a summation formula. Of course, this technique requires one to compute nasty derivatives correctly, but this is at least theoretically possible!

Practice Problems

Exercise 1.1.1 We begin with two large glasses. The first glass contains a pint of water, and the second contains a pint of wine. We pour 1/3 of a pint from the first glass into the second, stir up the wine/water mixture in the second glass, and then pour 1/3 of a pint of the mix back into the first glass and repeat this pouring back-and-forth process a total of n times.

(a) Describe a closed-form formula for the amount of wine in the first glass after n back-and-forth pourings.

(b) What is the limit of the amount of wine in each glass as n approaches infinity?

Exercise 1.1.2 You've seen this neat trick for evaluating a geometric sum:

$$\begin{aligned} S &= 1 + z + z^2 + \ldots + z^n \\ zS &= z + z^2 + \ldots + z^n + z^{n+1} \\ S - zS &= 1 - z^{n+1} \\ S &= \frac{1-z^{n+1}}{1-z} \quad \text{(where } z \neq 1\text{)} \end{aligned}$$

Use the same approach to find a closed-form expression for this sum:

$$T = 1z + 2z^2 + 3z^3 + \ldots + nz^n$$

Exercise 1.1.3 Sammy the Shark is a financial service provider who offers loans on the following terms.

- Sammy loans a client m dollars in the morning. This puts the client m dollars in debt to Sammy.
- Each evening, Sammy first charges a service fee which increases the client's debt by f dollars, and then Sammy charges interest, which multiplies the debt by a factor of p. For example, Sammy might charge a "modest" ten cent service fee and 1% interest rate per day, and then f would be 0.1 and p would be 1.01.

(a) What is the client's debt at the end of the first day?

(b) What is the client's debt at the end of the second day?

(c) Write a formula for the client's debt after d days and find an equivalent closed form.

(d) If you borrowed \$10 from Sammy for a year, how much would you owe him?

Exercise 1.1.4 Is a EEE degree really worth more than an CSE degree? Let us say that a person with a EEE degree starts with \$40,000 and gets a \$20,000 raise every year after graduation, whereas a person with an CSE degree starts with \$30,000, but gets a 20% raise every year. Assume inflation is a fixed 8% every year. That is, \$1.08 a year from now is worth \$1.00 today.

(a) How much is a EEE degree worth today if the holder will work for n years following graduation?

(b) How much is an CSE degree worth in this case?

(c) If you plan to retire after twenty years, which degree would be worth more?

Exercise 1.1.5 Suppose you deposit \$100 into your Credit Union account today, then \$99 at the end of the first month from now, \$98 at tne end of the second months from now, and so on. Given that the average *rate of return on investment* is 0.3% per month, how long will it take to save \$5,000?

1.2 Sums of Powers

You may have become adept at verifying formula like

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

by employing induction or *well ordering principle* but that does not reveal where the expression on the right came from in the first place. Even more inexplicable is the closed form expression for the sum of consecutive squares:

$$\sum_{i=1}^{n} i^2 = \frac{(2n+1)(n+1)n}{6}. \tag{1.16}$$

It turns out that there is a way to derive these expressions, but before we explain it, we thought it would be fun—OK, our definition of "fun" may be different than yours—to show you how Gauss is supposed to have proved equation 1.1 when he was a young boy.

Gauss's idea is related to the perturbation method we used in Section 1.1.2. Let

$$S = \sum_{i=1}^{n} i.$$

Then we can write the sum in two orders:

$$\begin{aligned} S &= 1 + \quad 2 \quad + \ldots + (n-1) + n, \\ S &= n + (n-1) + \ldots + \quad 2 \quad + 1. \end{aligned}$$

Adding these two equations gives

$$\begin{aligned} 2S &= (n+1) + (n+1) + \cdots + (n+1) + (n+1) \\ &= n(n+1). \end{aligned}$$

Hence,

$$S = \frac{n(n+1)}{2}.$$

Not bad for a young child—Gauss showed some potential. . . .

Unfortunately, the same trick does not work for summing consecutive squares. However, we can observe that the result might be a third-degree polynomial in n, since the sum contains n terms that average out to a value that grows quadratically in n. So we might guess that

$$\sum_{i=1}^{n} i^2 = an^3 + bn^2 + cn + d.$$

If our guess is correct, then we can determine the parameters a, b, c and d by plugging in a few values for n. Each such value gives a linear equation in a, b, c and d. If we plug in enough values, we may get a linear system with a unique solution. Applying this method to our example gives:

$$\begin{array}{lll} n = 0 & \text{implies} & 0 = d \\ n = 1 & \text{implies} & 1 = a + b + c + d \\ n = 2 & \text{implies} & 5 = 8a + 4b + 2c + d \\ n = 3 & \text{implies} & 14 = 27a + 9b + 3c + d. \end{array}$$

Solving this system gives the solution $a = 1/3$, $b = 1/2$, $c = 1/6$, $d = 0$. Therefore, *if* our initial guess at the form of the solution was correct, then the summation is equal to $n^3/3 + n^2/2 + n/6$, which matches equation 1.16.

The point is that if the desired formula turns out to be a polynomial, then once you get an estimate of the *degree* of the polynomial, all the coefficients of the polynomial can be found automatically.

Be careful! This method lets you discover formulas, but it doesn't guarantee they are right! After obtaining a formula by this method, it's important to go back and *prove* it by induction or some other method. If the initial guess at the solution was not of the right form, then the resulting formula will be completely wrong! A later chapter will describe a method based on generating functions that does not require any guessing at all.

Exercise Problems

Exercise 1.2.1 Find a closed form for each of the following sums:

(a)

$$\sum_{i=1}^{n} \left(\frac{1}{i + 2012} - \frac{1}{i + 2013} \right).$$

(b) Assuming the following sum equals a polynomial in n, find the polynomial. Then verify by induction that the sum equals the polynomial you find.

$$\sum_{i=1}^{n} i^3$$

1.3 Approximating Sums

Unfortunately, it is not always possible to find a closed-form expression for a sum. For example, no closed form is known for

$$S = \sum_{i=1}^{n} \sqrt{i}.$$

In such cases, we need to resort to approximations for S if we want to have a closed form. The good news is that there is a general method to find closed-form upper and lower bounds that works well for many sums. Even better, the method is simple and easy to remember. It works by replacing the sum by an integral and then adding either the first or last term in the sum.

Definition 3. A function $f : \mathbb{R}^+ \to \mathbb{R}^+$ is *strictly increasing* when

$$x < y \text{ IMPLIES } f(x) < f(y),$$

and it is *weakly increasing*[4] when

$$x < y \text{ IMPLIES } f(x) \leq f(y).$$

Similarly, f is *strictly decreasing* when

$$x < y \text{ IMPLIES } f(x) > f(y),$$

and it is *weakly decreasing*[5] when

$$x < y \text{ IMPLIES } f(x) \geq f(y).$$

For example, 2^x and $\sqrt{x}$ are strictly increasing functions, while $\max\{x, 2\}$ and $\lceil x \rceil$ are weakly increasing functions. The functions $1/x$ and 2^{-x} are strictly decreasing, while $\min\{1/x, 1/2\}$ and $\lfloor 1/x \rfloor$ are weakly decreasing.

Theorem 4. *Let $f : \mathbb{R}^+ \to \mathbb{R}^+$ be a weakly increasing function. Define*

$$S \triangleq \sum_{i=1}^{n} f(i) \tag{1.17}$$

and

$$I \triangleq \int_1^n f(x)\, dx.$$

Then

$$I + f(1) \leq S \leq I + f(n). \tag{1.18}$$

Similarly, if f is weakly decreasing, then

$$I + f(n) \leq S \leq I + f(1).$$

Proof: Suppose $f : \mathbb{R}^+ \to \mathbb{R}^+$ is weakly increasing. The value of the sum S in (1.17) is the sum of the areas of n unit-width rectangles of heights $f(1), f(2), \ldots, f(n)$. This area of these rectangles is shown shaded in Figure 1.1.

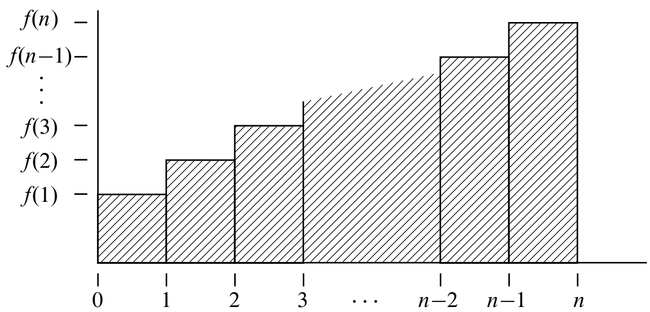


Figure 1.1: The area of the i^{th} rectangle is $f(i)$. The shaded region has area $\sum_{i=1}^{n} f(i)$.

The value of

$$I = \int_1^n f(x)\, dx$$

is the shaded area under the curve of $f(x)$ from 1 to n shown in Figure 1.2.

Comparing the shaded regions in Figures 1.1 and 1.2 shows that S is at least I plus the area of the leftmost rectangle. Hence,

$$S \geq I + f(1) \tag{1.19}$$

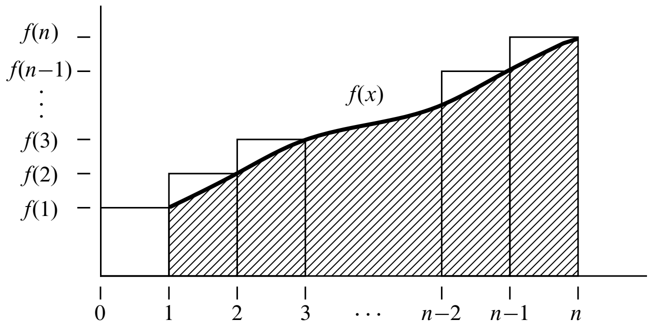


Figure 1.2: The shaded area under the curve of $f(x)$ from 1 to n (shown in bold) is $I = \int_1^n f(x)\,dx$.

This is the lower bound for S given in (1.18).

To derive the upper bound for S given in (1.18), we shift the curve of $f(x)$ from 1 to n one unit to the left as shown in Figure 1.3.

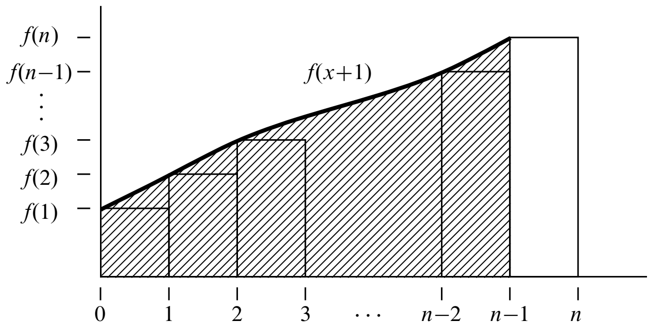


Figure 1.3: This curve is the same as the curve in Figure 1.2 shifted left by 1.

Comparing the shaded regions in Figures 1.1 and 1.3 shows that S is at most I plus the area of the rightmost rectangle. That is,

$$S \leq I + f(n),$$

which is the upper bound for S given in (1.18).

The very similar argument proves the weakly decreasing case[6]. ■

Theorem 4 provides good bounds for most sums. At worst, the bounds will be off by the largest term in the sum. For example, we can use Theorem 4 to bound the sum

$$S = \sum_{i=1}^{n} \sqrt{i}$$

as follows.

We begin by computing

$$I = \int_1^n \sqrt{x}\,dx = \left.\frac{x^{3/2}}{3/2}\right|_1^n = \frac{2}{3}(n^{3/2} - 1).$$

We then apply Theorem 4 to conclude that

$$\frac{2}{3}(n^{3/2} - 1) + 1 \;\leq\; S \;\leq\; \frac{2}{3}(n^{3/2} - 1) + \sqrt{n}$$

[4]Weakly increasing functions are usually called *nondecreasing* functions. We will avoid this terminology to prevent confusion between being a nondecreasing function and the much weaker property of *not* being a decreasing function.

[5]Weakly decreasing functions are usually called *nonincreasing*.

[6]left as an exercise for the reader.

and thus that

$$\frac{2}{3}n^{3/2} + \frac{1}{3} \le S \le \frac{2}{3}n^{3/2} + \sqrt{n} - \frac{2}{3}.$$

In other words, the sum is very close to $\frac{2}{3}n^{3/2}$. We'll define several ways that one thing can be "very close to" something else at the end of this chapter.

In an application of Theorem 4, we explain in a later chapter how it helps in resolving a classic paradox in structural engineering.

Practice Problems

Exercise 1.3.1 Let

$$S \triangleq \sum_{n=1}^{5} \sqrt{3n}\,.$$

Using the *Integral Method* of Section 1.3, we can find integers a, b, c, d and a real number e such that

$$\int_a^b x^e\,dx \le S \le \int_c^d x^e\,dx$$

What are appropriate values for $a, \ldots, e$?

Exercise 1.3.2

Let $f : \mathbb{R} \to \mathbb{R}$ be a continuous, weakly increasing function. Say that f *grows slowly* when

$$f(n) = o\left(\int_1^n f(x)\,dx\right).$$

(a) Prove that the function $f_a(n) \triangleq n^a$ grows slowly for any $a > 0$.

(b) Prove that the function e^n does not grow slowly.

(c) Prove that if f grows slowly, then

$$\int_1^n f(x)\,dx \sim \sum_{i=1}^{n} f(i)\,.$$

Exercise 1.3.3 Assume n is an integer larger than 1. Circle all the correct inequalities below.

Explanations are not required, but partial credit for wrong answers will not be given without them. *Hint:* You may find the graphs in Figure 1.4 helpful.

- $\sum_{i=1}^{n} \ln(i+1) \le \ln 2 + \int_1^n \ln(x+1)dx$
- $\sum_{i=1}^{n} \ln(i+1) \le \int_0^n \ln(x+2)dx$
- $\sum_{i=1}^{n} \frac{1}{i} \ge \int_0^n \frac{1}{x+1}dx$

Exercise 1.3.4 Let $f : \mathbb{R}^+ \to \mathbb{R}^+$ be a weakly decreasing function. Define

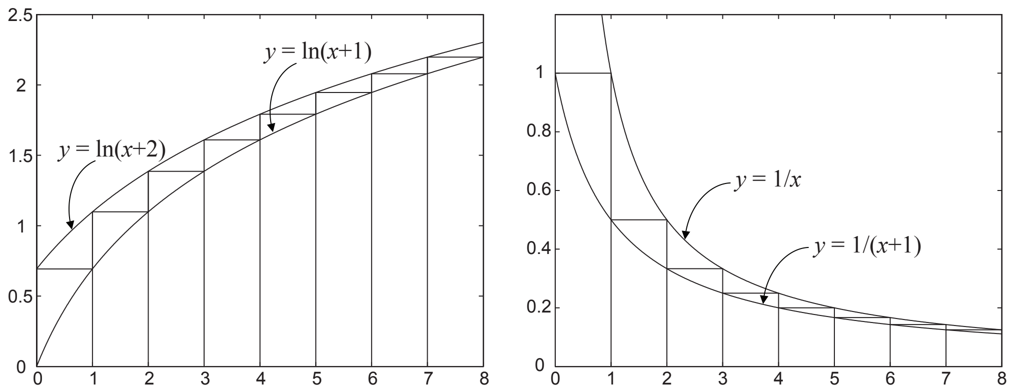


Figure 1.4: Integral bounds for two sums

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

and

$$I \triangleq \int_{1}^{n} f(x)\,dx.$$

Prove that

$$I + f(n) \leq S \leq I + f(1).$$

(Proof by very clear picture is OK.)

Exercise 1.3.5 Use integration to find upper and lower bounds that differ by at most 0.1 for the following sum. (You may need to add the first few terms explicitly and then use integrals to bound the sum of the remaining terms.)

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

1.4 Products

We've covered several techniques for finding closed forms for sums but no methods for dealing with products. Fortunately, we do not need to develop an entirely new set of tools when we encounter a product such as

$$n! \triangleq \prod_{i=1}^{n} i. \tag{1.20}$$

That's because we can convert any product into a sum by taking a logarithm. For example, if

$$P = \prod_{i=1}^{n} f(i),$$

then

$$\ln(P) = \sum_{i=1}^{n} \ln(f(i)).$$

We can then apply our summing tools to find a closed form (or approximate closed form) for $\ln(P)$ and then exponentiate at the end to undo the logarithm.

For example, let's see how this works for the *factorial* function $n!$. We start by taking the logarithm:

$$\begin{aligned}\ln(n!) &= \ln(1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n) \\ &= \ln(1) + \ln(2) + \ln(3) + \cdots + \ln(n-1) + \ln(n) \\ &= \sum_{i=1}^{n} \ln(i).\end{aligned}$$

Unfortunately, no closed form for this sum is known. However, we can apply Theorem 4 to find good closed-form bounds on the sum. To do this, we first compute

$$\begin{aligned}\int_1^n \ln(x)\,dx &= x\ln(x) - x\Big|_1^n \\ &= n\ln(n) - n + 1.\end{aligned}$$

Plugging into Theorem 4, this means that

$$n\ln(n) - n + 1 \;\le\; \sum_{i=1}^{n} \ln(i) \;\le\; n\ln(n) - n + 1 + \ln(n).$$

Exponentiating then gives

$$\frac{n^n}{e^{n-1}} \;\le\; n! \;\le\; \frac{n^{n+1}}{e^{n-1}}. \tag{1.21}$$

This means that $n!$ is within a factor of n of n^n/e^{n-1}.

1.4.1 Stirling's Formula

The most commonly used product in discrete mathematics is probably $n!$, and mathematicians have workedto find tight closed-form bounds on its value. The most useful bounds are given in Theorem 5.

Theorem 5 (*Stirling's Formula*)**.** *For all $n \ge 1$,*

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\epsilon(n)}$$

where

$$\frac{1}{12n+1} \le \epsilon(n) \le \frac{1}{12n}.$$

Theorem 5 can be proved by induction (with some pain), and there are lots of proofs using elementary calculus, but we won't go into them.

There are several important things to notice about Stirling's Formula. First, $\epsilon(n)$ is always positive. This means that

$$n! > \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \tag{1.22}$$

for all $n \in \mathbb{N}^+$.

Second, $\epsilon(n)$ tends to zero as n gets large. This means that

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \tag{1.23}$$

which is impressive. After all, who would expect both π and e to show up in a closed-form expression that is asymptotically equal to $n!$?

Third, $\epsilon(n)$ is small even for small values of n. This means that Stirling's Formula provides good approximations for $n!$ for most all values of n. For example, if we use

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

Approximation	$n \geq 1$	$n \geq 10$	$n \geq 100$	$n \geq 1000$
$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n$	$< 10\%$	$< 1\%$	$< 0.1\%$	$< 0.01\%$
$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{1/12n}$	$< 1\%$	$< 0.01\%$	$< 0.0001\%$	$< 0.000001\%$

Table 1.1: Error bounds on common approximations for $n!$ from Theorem 5. For example, if $n \geq 100$, then $\sqrt{2\pi n}\left(\frac{n}{e}\right)^n$ approximates $n!$ to within 0.1%.

as the approximation for $n!$, as many people do, we are guaranteed to be within a factor of

$$e^{\epsilon(n)} \leq e^{\frac{1}{12n}}$$

of the correct value. For $n \geq 10$, this means we will be within 1% of the correct value. For $n \geq 100$, the error will be less than 0.1%.

If we need an even closer approximation for $n!$, then we could use either

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{1/12n}$$

or

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{1/(12n+1)}$$

depending on whether we want an upper, or a lower, bound. By Theorem 5, we know that both bounds will be within a factor of

$$e^{\frac{1}{12n}-\frac{1}{12n+1}} = e^{\frac{1}{144n^2+12n}}$$

of the correct value. For $n \geq 10$, this means that either bound will be within 0.01% of the correct value. For $n \geq 100$, the error will be less than 0.0001%.

For quick future reference, these facts are summarized in Corollary 6 and Table 1.1.

Corollary 6.

$$n! < \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \cdot \begin{cases} 1.09 & \text{for } n \geq 1, \\ 1.009 & \text{for } n \geq 10, \\ 1.0009 & \text{for } n \geq 100. \end{cases}$$

1.5 Double Trouble

Sometimes we have to evaluate sums of sums, otherwise known as *double summations.* This sounds hairy, and sometimes it is. But usually, it is straightforward—you just evaluate the inner sum, replace it with a closed form, and then evaluate the outer sum (which no longer has a summation inside it). For example,[7]

$$\begin{aligned}
\sum_{n=0}^{\infty}\left(y^n \sum_{i=0}^{n} x^i\right) &= \sum_{n=0}^{\infty}\left(y^n \frac{1-x^{n+1}}{1-x}\right) && \text{equation 1.2} \\
&= \left(\frac{1}{1-x}\right)\sum_{n=0}^{\infty} y^n - \left(\frac{1}{1-x}\right)\sum_{n=0}^{\infty} y^n x^{n+1} \\
&= \frac{1}{(1-x)(1-y)} - \left(\frac{x}{1-x}\right)\sum_{n=0}^{\infty}(xy)^n && \text{Theorem 1} \\
&= \frac{1}{(1-x)(1-y)} - \frac{x}{(1-x)(1-xy)} && \text{Theorem 1}
\end{aligned}$$

[7] OK, so maybe this one is a little hairy, but it is also fairly straightforward. Wait till you see the next one!

$$= \frac{(1-xy)-x(1-y)}{(1-x)(1-y)(1-xy)}$$
$$= \frac{1-x}{(1-x)(1-y)(1-xy)}$$
$$= \frac{1}{(1-y)(1-xy)}.$$

When there's no obvious closed form for the inner sum, a special trick that is often useful is to try *exchanging the order of summation.* To illustrate this, we bring in a sum that involves *harmonic numbers*, which we cover in some detail in Section 2.1. For now it suffices to look at its definition

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} = \sum_{i=1}^{n} \frac{1}{i}$$

Now, suppose we want to compute the sum of the first n *harmonic numbers*

$$\sum_{k=1}^{n} H_k = \sum_{k=1}^{n} \sum_{j=1}^{k} \frac{1}{j} \tag{1.24}$$

For intuition about this sum, we can apply Theorem 4 to equation 2.3 to conclude that the sum is close to

$$\int_1^n \ln(x)\,dx = x\ln(x) - x \Big|_1^n = n\ln(n) - n + 1.$$

Now let's look for an exact answer. If we think about the pairs (k, j) over which we are summing, they form a triangle:

		j 1	2	3	4	5	$\dots$	n
k	1	1						
	2	1	1/2					
	3	1	1/2	1/3				
	4	1	1/2	1/3	1/4			
		$\dots$						
	n	1	1/2		$\dots$			$1/n$

The summation in equation 1.24 is summing each row and then adding the row sums. Instead, we can sum the columns and then add the column sums. Inspecting the table we see that this double sum can be written as

$$\sum_{k=1}^{n} H_k = \sum_{k=1}^{n} \sum_{j=1}^{k} \frac{1}{j}$$
$$= \sum_{j=1}^{n} \sum_{k=j}^{n} \frac{1}{j}$$
$$= \sum_{j=1}^{n} \frac{1}{j} \sum_{k=j}^{n} 1$$
$$= \sum_{j=1}^{n} \frac{1}{j}(n-j+1)$$
$$= \sum_{j=1}^{n} \frac{n+1}{j} - \sum_{j=1}^{n} \frac{j}{j}$$
$$= (n+1)\sum_{j=1}^{n} \frac{1}{j} - \sum_{j=1}^{n} 1$$

$$= (n+1)H_n - n. \tag{1.25}$$

1.6 Asymptotic Notation

Asymptotic notation is a shorthand used to give a quick measure of the behavior of a function $f(n)$ as n grows large. For example, the asymptotic notation $\sim$ of Definition 22 is a binary relation indicating that two functions grow at the *same* rate. There is also a binary relation "little oh" indicating that one function grows at a significantly *slower* rate than another and "Big Oh" indicating that one function grows not much more rapidly than another.

1.6.1 Little O

Definition 7. For functions $f, g : \mathbb{R} \to \mathbb{R}$, with g nonnegative, we say f is*asymptotically smaller* than g, in symbols,

$$f(x) = o(g(x)),$$

iff

$$\lim_{x\to\infty} f(x)/g(x) = 0.$$

For example, $1000x^{1.9} = o(x^2)$ because $1000x^{1.9}/x^2 = 1000/x^{0.1}$ and since $x^{0.1}$ goes to infinity with x and 1000 is constant, we have $\lim_{x\to\infty} 1000x^{1.9}/x^2 = 0$. This argument generalizes directly to yield

Lemma 8. *$x^a = o(x^b)$ for all nonnegative constants $a < b$.*

Using the familiar fact that $\log x < x$ for all $x > 1$, we can prove

Lemma 9. *$\log x = o(x^\epsilon)$ for all $\epsilon > 0$.*

Proof. Choose $\epsilon > \delta > 0$ and let $x = z^\delta$ in the inequality $\log x < x$. This implies

$$\log z < z^\delta/\delta = o(z^\epsilon) \qquad \text{by Lemma 8.} \tag{1.26}$$

□

Corollary 10. *$x^b = o(a^x)$ for any $a, b \in \mathbb{R}$ with $a > 1$.*

Lemma 9 and Corollary 10 can also be proved using l'Hôpital's Rule or the Maclaurin Series for $\log x$ and e^x. Proofs can be found in most calculus texts.

1.6.2 Big O

"Big Oh" is the most frequently used asymptotic notation. It is used to give an upper bound on the growth of a function, such as the running time of an algorithm. There is a standard definition of Big Oh given below in 15, but we'll begin with an alternative definition that makes apparent several basic properties of Big Oh.

Definition 11. Given functions $f, g : \mathbb{R} \to \mathbb{R}$ with g nonnegative, we say that

$$f = O(g)$$

iff

$$\limsup_{x\to\infty} |f(x)|/g(x) < \infty.$$

Here we're using the technical notion of *limit superior*[8] instead of just limit. But because limits and lim sup's are the same when limits exist, this formulation makes it easy to check basic properties of Big Oh. We'll take the following Lemma for granted.

[8]The precise definition of lim sup is

$$\limsup_{x\to\infty} h(x) \triangleq \lim_{x\to\infty} \text{lub}_{y\geq x} h(y),$$

where "lub" abbreviates "least upper bound."

Lemma 12. *If a function $f : \mathbb{R} \to \mathbb{R}$ has a finite or infinite limit as its argument approaches infinity, then its limit and limit superior are the same.*

Now Definition 11 immediately implies:

Lemma 13. *If $f = o(g)$ or $f \sim g$, then $f = O(g)$.*

Proof. $\lim f/g = 0$ or $\lim f/g = 1$ implies $\lim f/g < \infty$, so by Lemma 12, $\limsup f/g < \infty$. □

Note that the converse of Lemma 13 is not true. For example, $2x = O(x)$, but $2x \not\sim x$ and $2x \neq o(x)$.

We also have:

Lemma 14. *If $f = o(g)$, then it is* not *true that $g = O(f)$.*

Proof.

$$\lim_{x\to\infty} \frac{g(x)}{f(x)} = \frac{1}{\lim_{x\to\infty} f(x)/g(x)} = \frac{1}{0} = \infty,$$

so by Lemma 12, $g \neq O(f)$. □

We need lim sup's in Definition 11 to cover cases when limits don't exist. For example, if $f(x)/g(x)$ oscillates between 3 and 5 as x grows, then $\lim_{x\to\infty} f(x)/g(x)$ does not exist, but $f = O(g)$ because $\limsup_{x\to\infty} f(x)/g(x) = 5$.

An equivalent, more usual formulation of big O does not mention lim sup's:

Definition 15. Given functions $f, g : \mathbb{R} \to \mathbb{R}$ with g nonnegative, we say

$$f = O(g)$$

iff there exists a constant $c \geq 0$ and an x_0 such that for all $x \geq x_0$, $|f(x)| \leq cg(x)$.

This definition is rather complicated, but the idea is simple: $f(x) = O(g(x))$ means $f(x)$ is less than or equal to $g(x)$, except that we're willing to ignore a constant factor, namely c and to allow exceptions for small x, namely, $x < x_0$. So in the case that $f(x)/g(x)$ oscillates between 3 and 5, $f = O(g)$ according to Definition 15 because $f \leq 5g$.

Proposition 16. $100x^2 = O(x^2)$.

Proof. Choose $c = 100$ and $x_0 = 1$. Then the proposition holds, since for all $x \geq 1$, $|100x^2| \leq 100x^2$. □

Proposition 17. $x^2 + 100x + 10 = O(x^2)$.

Proof. $(x^2+100x+10)/x^2 = 1+100/x+10/x^2$ and so its limit as x approaches infinity is $1+0+0 = 1$. So in fact, $x^2 + 100x + 10 \sim x^2$, and therefore $x^2 + 100x + 10 = O(x^2)$. Indeed, it's conversely true that $x^2 = O(x^2 + 100x + 10)$. □

Proposition 17 generalizes to an arbitrary polynomial:

Proposition 18. $a_k x^k + a_{k-1}x^{k-1} + \cdots + a_1 x + a_0 = O(x^k)$.

We'll omit the routine proof.

Big O notation is especially useful when describing the running time of an algorithm. For example, the usual algorithm for multiplying $n \times n$ matrices uses a number of operations proportional to n^3 in the worst case. This fact can be expressed concisely by saying that the running time is $O(n^3)$. So this asymptotic notation allows the speed of the algorithm to be discussed without reference to constant factors or lower-order terms that might be machine specific. It turns out that there is another *matrix multiplication* procedure that uses $O(n^{2.55})$ operations. The fact that this procedure is asymptotically faster indicates that it involves new ideas that go beyond a simply more efficient implementation of the $O(n^3)$ method.

Of course the asymptotically faster procedure will also definitely be much more efficient on large enough matrices, but being asymptotically faster does not mean that it is a better choice. The $O(n^{2.55})$-operation multiplication procedure is almost never used in practice because it only becomes more efficient than the usual $O(n^3)$ procedure on matrices of impractical size.[9]

1.6.3 Theta $\Theta()$

Sometimes we want to specify that a running time $T(n)$ is precisely quadratic up to constant factors (both upper bound *and* lower bound). We could do this by saying that $T(n) = O(n^2)$ and $n^2 = O(T(n))$, but rather than say both, mathematicians have devised yet another symbol Θ to do the job.

Definition 19.

$$f = \Theta(g) \quad \text{iff} \quad f = O(g) \text{ and } g = O(f).$$

The statement $f = \Theta(g)$ can be paraphrased intuitively as "f and g are equal to within a constant factor."

The Theta notation allows us to highlight growth rates and suppress distracting factors and low-order terms. For example, if the running time of an algorithm is

$$T(n) = 10n^3 - 20n^2 + 1,$$

then we can more simply write

$$T(n) = \Theta(n^3).$$

In this case, we would say that T *is of order* n^3 or that $T(n)$ *grows cubically*, which is often the main thing we really want to know. Another such example is

$$\pi^2 3^{x-7} + \frac{(2.7x^{113} + x^9 - 86)^4}{\sqrt{x}} - 1.08^{3x} = \Theta(3^x).$$

Just knowing that the running time of an algorithm is $\Theta(n^3)$, for example, is useful, because if n doubles we can predict that the running time will *by and large*[10] increase by a factor of at most 8 for large n. In this way, Theta notation preserves information about the scalability of an algorithm or system. Scalability is, of course, a big issue in the design of algorithms and systems.

1.6.4 Pitfalls with Asymptotic Notation

There is a long list of ways to make mistakes with asymptotic notation. This section presents some of the ways that big O notation can lead to trouble. With minimal effort, you can cause just as much chaos with the other symbols.

The Exponential Fiasco

Sometimes relationships involving big O are not so obvious. For example, one might guess that $4^x = O(2^x)$ since 4 is only a constant factor larger than 2. This reasoning is incorrect, however; 4^x actually grows as the square of 2^x.

[9] It is even conceivable that there is an $O(n^2)$ matrix multiplication procedure, but none is known.

[10] Since $\Theta(n^3)$ only implies that the running time $T(n)$ is between cn^3 and dn^3 for constants $0 < c < d$, the time $T(2n)$ could regularly exceed $T(n)$ by a factor as large as $8d/c$. The factor is sure to be close to 8 for all large n only if $T(n) \sim n^3$.

Constant Confusion

Every constant is $O(1)$. For example, $17 = O(1)$. This is true because if we let $f(x) = 17$ and $g(x) = 1$, then there exists a $c > 0$ and an x_0 such that $|f(x)| \leq cg(x)$. In particular, we could choose $c = 17$ and $x_0 = 1$, since $|17| \leq 17 \cdot 1$ for all $x \geq 1$. We can construct a false theorem that exploits this fact.

False Theorem 20.

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

Bogus proof. Define $f(n) = \sum_{i=1}^{n} i = 1 + 2 + 3 + \cdots + n$. Since we have shown that every constant i is $O(1)$, $f(n) = O(1) + O(1) + \cdots + O(1) = O(n)$. $\square$

Of course in reality $\sum_{i=1}^{n} i = n(n+1)/2 \neq O(n)$.

The error stems from confusion over what is meant in the statement $i = O(1)$. For any *constant* $i \in \mathbb{N}$ it is true that $i = O(1)$. More precisely, if f is any constant function, then $f = O(1)$. But in this False Theorem, i is not constant—it ranges over a set of values $0, 1, \ldots, n$ that depends on n.

And anyway, we should not be adding $O(1)$'s as though they were numbers. We never even defined what $O(g)$ means by itself; it should only be used in the context "$f = O(g)$" to describe a relation between functions f and g.

Equality Blunder

The notation $f = O(g)$ is too firmly entrenched to avoid, but the use of "=" is regrettable. For example, if $f = O(g)$, it seems quite reasonable to write $O(g) = f$. But doing so might tempt us to the following blunder: because $2n = O(n)$, we can say $O(n) = 2n$. But $n = O(n)$, so we conclude that $n = O(n) = 2n$, and therefore $n = 2n$. To avoid such nonsense, we will never write "$O(f) = g$."

Similarly, you will often see statements like

$$H_n = \ln(n) + \gamma + O\left(\frac{1}{n}\right)$$

or

$$n! = (1 + o(1))\sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

In such cases, the true meaning is

$$H_n = \ln(n) + \gamma + f(n)$$

for some $f(n)$ where $f(n) = O(1/n)$, and

$$n! = (1 + g(n))\sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

where $g(n) = o(1)$. These last transgressions are OK as long as you (and your reader) know what you mean.

Operator Application Blunder

It's tempting to assume that familiar operations preserve asymptotic relations, but it ain't necessarily so. For example, $f \sim g$ in general does not even imply that $3^f = \Theta(3^g)$. On the other hand, some operations preserve and even strengthen asymptotic relations, for example,

$$f = \Theta(g) \quad \text{IMPLIES} \quad \ln f \sim \ln g.$$

The case of ω is symmetric to o and is omitted for brevity here.

Practice Problems

Exercise 1.6.1 Find the least nonnegative integer n such that $f(x)$ is $O(x^n)$ when f is defined by each of the expressions below.

(a) $2x^3 + (\log x)x^2$

(b) $2x^2 + (\log x)x^3$

(c) $(1.1)^x$

(d) $(0.1)^x$

(e) $(x^4 + x^2 + 1)/(x^3 + 1)$

(f) $(x^4 + 5\log x)/(x^4 + 1)$

(g) $2^{(3\log_2 x^2)}$

Exercise 1.6.2 Let $f(n) = n^3$. For each function $g(n)$ in the table below, indicate which of the indicated *asymptotic relations* hold.

$g(n)$	$f = O(g)$	$f = o(g)$	$g = O(f)$	$g = o(f)$
$6 - 5n - 4n^2 + 3n^3$				
$n^3 \log n$				
$(\sin(\pi n/2) + 2)\, n^3$				
$n^{\sin(\pi n/2)+2}$				
$\log n!$				
$e^{0.2n} - 100n^3$				

Exercise 1.6.3 Circle each of the true statements below.

Explanations are not required, but partial credit for wrong answers will not be given without them.

- $n^2 \sim n^2 + n$
- $3^n = O(2^n)$
- $n^{\sin(n\pi/2)+1} = o\left(n^2\right)$
- $n = \Theta\left(\dfrac{3n^3}{(n+1)(n-1)}\right)$

Exercise 1.6.4 Show that

$$\ln(n^2!) = \Theta(n^2 \ln n)$$

Hint: Stirling's formula for $(n^2)!$.

Exercise 1.6.5 The quantity

$$\frac{(2n)!}{2^{2n}(n!)^2} \tag{1.27}$$

will come up later in the course (it is the probability that in 2^{2n} flips of a fair coin, exactly n will be Heads). Show that it is asymptotically equal to $\dfrac{1}{\sqrt{\pi n}}$.

Exercise 1.6.6 Indicate which of the following holds for each pair of functions $(f(n), g(n))$ in the table below. Assume $k \geq 1$, $\epsilon > 0$, and $c > 1$ are constants. Pick the four table entries you consider to be the most challenging or interesting and justify your answers to these.

$f(n)$	$g(n)$	$f = O(g)$	$f = o(g)$	$g = O(f)$	$g = o(f)$	$f = \Theta(g)$	$f \sim g$
2^n	$2^{n/2}$						
$\sqrt{n}$	$n^{\sin(n\pi/2)}$						
$\log(n!)$	$\log(n^n)$						
n^k	c^n						
$\log^k n$	n^ϵ						

Exercise 1.6.7 Arrange the following functions in a sequence f_1, f_2, ... f_{24} so that $f_i = O(f_{i+1})$. Additionally, if $f_i = \Theta(f_{i+1})$, indicate that too:

1. $n \log n$
2. $2^{100} n$
3. n^{-1}
4. $n^{-1/2}$
5. $(\log n)/n$
6. $\binom{n}{64}$
7. $n!$
8. $2^{2^{100}}$
9. 2^{2^n}
10. 2^n
11. 3^n
12. $n2^n$
13. 2^{n+1}
14. $2n$
15. $3n$
16. $\log(n!)$
17. $\log_2 n$
18. $\log_{10} n$
19. $2.1^{\sqrt{n}}$
20. 2^{2n}
21. 4^n
22. n^{64}
23. n^{65}
24. n^n

Exercise 1.6.8 Let f, g be nonnegative real-valued functions such that $\lim_{x \to \infty} f(x) = \infty$ and $f \sim g$.

(a) Give an example of f, g such that $\text{NOT}(2^f \sim 2^g)$.

(b) Prove that $\log f \sim \log g$.

(c) Use Stirling's formula to prove that in fact $\log(n!) \sim n \log n$

Exercise 1.6.9 Determine which of these choices

$$\Theta(n), \quad \Theta(n^2 \log n), \quad \Theta(n^2), \quad \Theta(1), \quad \Theta(2^n), \quad \Theta(2^{n \ln n}), \quad \text{none of these}$$

describes each function's asymptotic behavior. Full proofs are not required, but briefly explain your answers.

(a) $n + \ln n + (\ln n)^2$

(b) $\dfrac{n^2 + 2n - 3}{n^2 - 7}$

(c) $\displaystyle\sum_{i=0}^{n} 2^{2i+1}$

(d) $\ln(n^2!)$

(e) $\displaystyle\sum_{k=1}^{n} k \left(1 - \frac{1}{2^k}\right)$

Exercise 1.6.10 Answer the folloing two questions.

(a) Either prove or disprove each of the following statements.

- $n! = O((n+1)!)$
- $(n+1)! = O(n!)$
- $n! = \Theta((n+1)!)$
- $n! = o((n+1)!)$
- $(n+1)! = o(n!)$

(b) Show that $\left(\frac{n}{3}\right)^{n+e} = o(n!)$.

Exercise 1.6.11 Prove that $\displaystyle\sum_{k=1}^{n} k^6 = \Theta(n^7)$.

Exercise 1.6.12 Give an elementary proof (without appealing to Stirling's formula) that $\log(n!) = \Theta(n \log n)$.

Exercise 1.6.13 Recall that for functions f, g on $\mathbb{N}$, $f = O(g)$ iff

$$\exists c \in \mathbb{N}\, \exists n_0 \in \mathbb{N}\, \forall n \geq n_0 \quad c \cdot g(n) \geq |f(n)|\,. \tag{1.28}$$

For each pair of functions below, determine whether $f = O(g)$ and whether $g = O(f)$. In cases where one function is $O()$ of the other, indicate the *smallest nonnegative integer* c and for that smallest c, the *smallest corresponding nonnegative integer* n_0 ensuring that condition (1.28) applies.

(a) $f(n) = n^2, g(n) = 3n$.

$f = O(g)$	YES	NO	If YES, $c =$ ______, $n_0 =$ ______
$g = O(f)$	YES	NO	If YES, $c =$ ______, $n_0 =$ ______

(b) $f(n) = (3n - 7)/(n + 4), g(n) = 4$

$f = O(g)$	YES	NO	If YES, $c =$ ______, $n_0 =$ ______
$g = O(f)$	YES	NO	If YES, $c =$ ______, $n_0 =$ ______

(c) $f(n) = 1 + (n \sin(n\pi/2))^2, g(n) = 3n$

$f = O(g)$	YES	NO	If yes, $c =$ ______ $n_0 =$ ______
$g = O(f)$	YES	NO	If yes, $c =$ ______ $n_0 =$ ______

Exercise 1.6.14 Explain why the following claim is false.

False Claim.

$$2^n = O(1). \tag{1.29}$$

Then identify and explain the mistake in the following bogus proof.

Bogus proof. The proof is by induction on n where the induction hypothesis $P(n)$ is the assertion (1.29).

base case: $P(0)$ holds trivially.

inductive step: We may assume $P(n)$, so there is a constant $c > 0$ such that $2^n \leq c \cdot 1$. Therefore,

$$2^{n+1} = 2 \cdot 2^n \leq (2c) \cdot 1,$$

which implies that $2^{n+1} = O(1)$. That is, $P(n+1)$ holds, which completes the proof of the inductive step.

We conclude by induction that $2^n = O(1)$ for all n. That is, the exponential function is bounded by a constant.

□

Exercise 1.6.15 Give an example of a pair of strictly increasing total functions, $f : \mathbb{N}^+ \to \mathbb{N}^+$ and $g : \mathbb{N}^+ \to \mathbb{N}^+$, that satisfy $f \sim g$ but **not** $3^f = O(3^g)$.

Exercise 1.6.16 Let f, g be real-valued functions such that $f = \Theta(g)$ and $\lim_{x\to\infty} f(x) = \infty$. Prove that

$$\ln f \sim \ln g.$$

Exercise 1.6.17 Let f, g be positive real-valued functions on finite, *connected*, simple graphs. We will extend the $O()$ notation to such graph functions as follows: $f = O(g)$ iff there is a constant $c > 0$ such that

$$f(G) \leq c \cdot g(G) \text{ for all connected simple graphs } G \text{ with more than one vertex.}$$

For each of the following assertions, state whether it is **True** or **False** and briefly explain your answer. You are **not** expected to offer a careful proof or detailed counterexample.

Reminder: $V(G)$ is the set of vertices and $E(G)$ is the set of edges of G, and G is connected.

(a) $|V(G)| = O(|E(G)|)$.

(b) $|E(G)| = O(|V(G)|)$.

(c) $|V(G)| = O(\chi(G))$, where $\chi(G)$ is the chromatic number of G.

(d) $\chi(G) = O(|V(G)|)$.

2 | Special Numbers

Some sequences of "numbers" arise so often in mathematical analysis that we recognize them instantly and give them special names. For example, everybody who learns arithmetic knows the sequence of square numbers $\langle 1, 4, 9, 16, \ldots \rangle$. The prime numbers $\langle 2, 3, 5, 7, \ldots \rangle$, of course, are studied a lot, and you may have encountered the Catalan numbers $\langle 1, 2, 5, 14, \ldots \rangle$, a bit of exotic variety even among the special numbers.

In the present chapter we'll get to know a few other important sequences. First on our agenda will be the harmonic numbers H_n, followed by an exposition of the fascinating Fibonacci numbers F_n.

2.1 Harmonic Numbers

Harmonic numbers are remarkable as they often arise naturally in simple situations. To highlight this point of relevance, we set the stage with the following illustrative case of an analytic problem.

2.1.1 Hanging Out Over the Edge

Suppose you have a bunch of books and you want to stack them up, one on top of another in some off-center way, so the top book sticks out past books below it without falling over. If you moved the stack to the edge of a table, how far past the edge of the table do you think you could get the top book to go? Could the top book stick out completely beyond the edge of table? You're not supposed to use glue or any other support to hold the stack in place.

Most people's first response to the Book Stacking Problem—sometimes also their second and third responses—is "No, the top book will never get completely past the edge of the table." But in fact, you can get the top book to stick out as far as you want: one booklength, two booklengths, any number of booklengths!

Formalizing the Problem

We'll approach this problem recursively. How far past the end of the table can we get one book to stick out? It won't tip as long as its center of mass is over the table, so we can get it to stick out half its length, as shown in Figure 2.1.

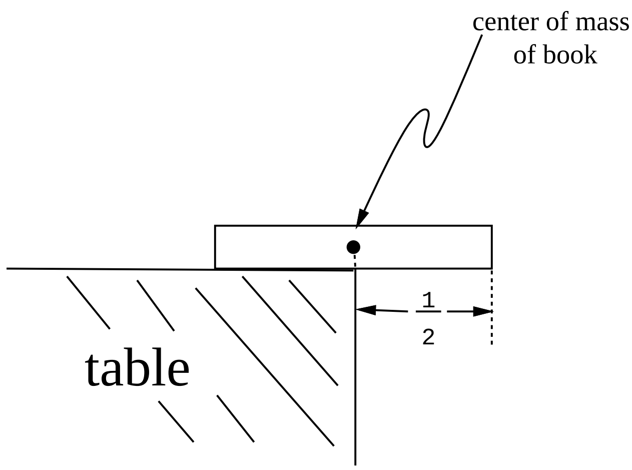


Figure 2.1: One book can overhang half a book length.

Now suppose we have a stack of books that will not tip over if the bottom book rests on the table—call that a *stable stack*. Let's define the *overhang* of a stable stack to be the horizontal distance from the center of mass of the stack to the furthest edge of the top book. So the overhang is purely a property of the stack, regardless of its placement on the table. If we place the center of mass of the stable stack at the edge of the table as in Figure 2.2, the overhang is how far we can get the top book in the stack to stick out past the edge.

In general, a stack of n books will be stable if and only if the center of mass of the top i books sits over the $(i+1)$st book for $i = 1, 2, \ldots, n-1$.

So we want a formula for the maximum possible overhang B_n achievable with a stable stack of n books.

We've already observed that the overhang of one book is 1/2 a book length. That is,

$$B_1 = \frac{1}{2}.$$

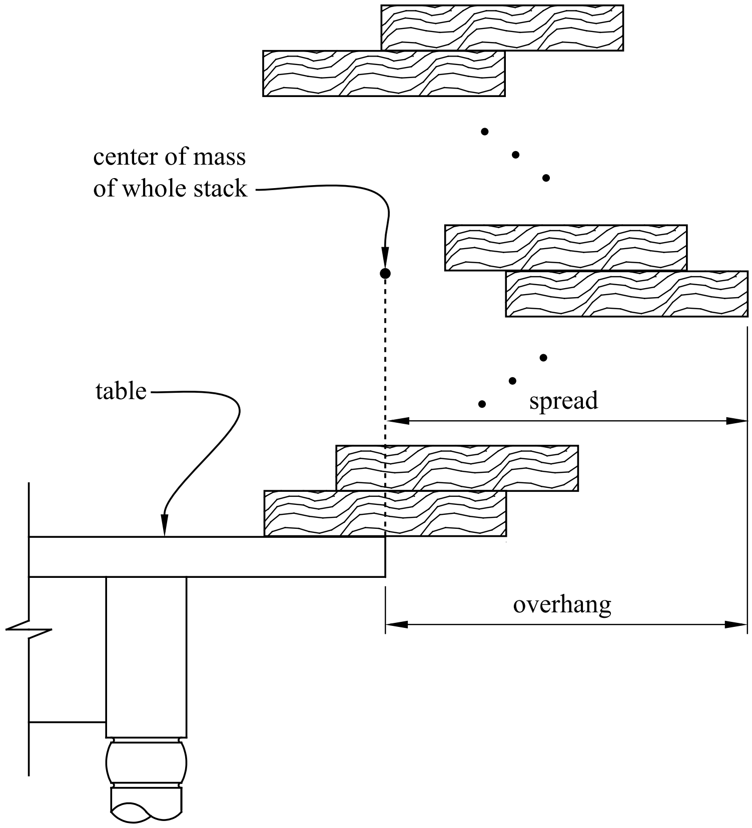


Figure 2.2: Overhanging the edge of the table.

Now suppose we have a stable stack of $n+1$ books with maximum overhang. If the overhang of the n books on top of the bottom book was not maximum, we could get a book to stick out further by replacing the top stack with a stack of n books with larger overhang. So the maximum overhang B_{n+1} of a stack of $n+1$ books is obtained by placing a maximum overhang stable stack of n books on top of the bottom book. And we get the biggest overhang for the stack of $n+1$ books by placing the center of mass of the n books right over the edge of the bottom book as in Figure 2.3.

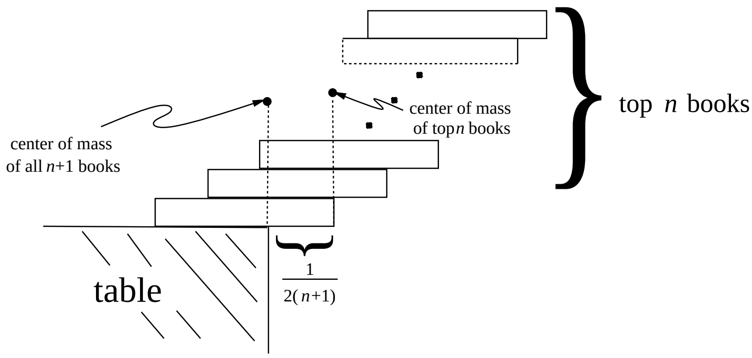


Figure 2.3: Additional overhang with $n+1$ books.

So we know where to place the $n+1$st book to get maximum overhang. In fact, the reasoning above actually shows that this way of stacking $n+1$ books is the *unique* way to build a stable stack where the top book extends as far as possible. All we have to do is calculate what this extension is.

The simplest way to do that is to let the center of mass of the top n books be the origin. That way the horizontal coordinate of the center of mass of the whole stack of $n+1$ books will equal the increase in the overhang. But now the center of mass of the bottom book has horizontal coordinate 1/2, so the horizontal coordinate of center of mass of the whole stack of $n+1$ books is

$$\frac{0 \cdot n + (1/2) \cdot 1}{n+1} = \frac{1}{2(n+1)}.$$

In other words,

$$B_{n+1} = B_n + \frac{1}{2(n+1)}, \tag{2.1}$$

as shown in Figure 2.3.

Expanding equation (2.1), we have

$$\begin{aligned} B_{n+1} &= B_{n-1} + \frac{1}{2n} + \frac{1}{2(n+1)} \\ &= B_1 + \frac{1}{2 \cdot 2} + \cdots + \frac{1}{2n} + \frac{1}{2(n+1)} \\ &= \frac{1}{2} \sum_{i=1}^{n+1} \frac{1}{i}. \end{aligned} \tag{2.2}$$

So our next task is to examine the behavior of B_n as n grows.

2.1.2 Properties of Harmonic Numbers

Definition 21. The nth *harmonic number* H_n is

$$H_n \triangleq \sum_{i=1}^{n} \frac{1}{i}.$$

So (2.2) means that

$$B_n = \frac{H_n}{2}.$$

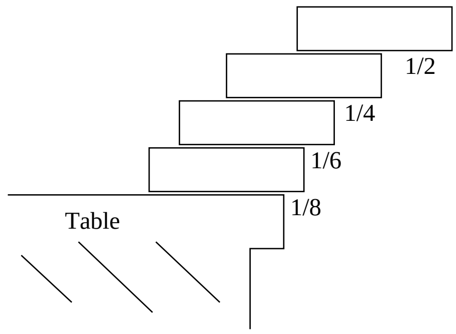


Figure 2.4: Stack of four books with maximum overhang.

The first few harmonic numbers are easy to compute. For example, $H_4 = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = \frac{25}{12} > 2$. The fact that H_4 is greater than 2 has special significance: it implies that the total extension of a 4-book stack is greater than one full book! This is the situation shown in Figure 2.4.

There is good news and bad news about harmonic numbers. The bad news is that there is no known closed-form expression for the harmonic numbers. The good news is that we can use Theorem 4 to get close upper and lower bounds on H_n. In particular, since

$$\int_1^n \frac{1}{x}\, dx = \ln(x) \Big|_1^n = \ln(n),$$

Theorem 4 means that

$$\ln(n) + \frac{1}{n} \le H_n \le \ln(n) + 1. \tag{2.3}$$

In other words, the nth harmonic number is very close to $\ln(n)$.

Because the harmonic numbers frequently arise in practice, mathematicians have worked hard to get even better approximations for them. In fact, it is now known that

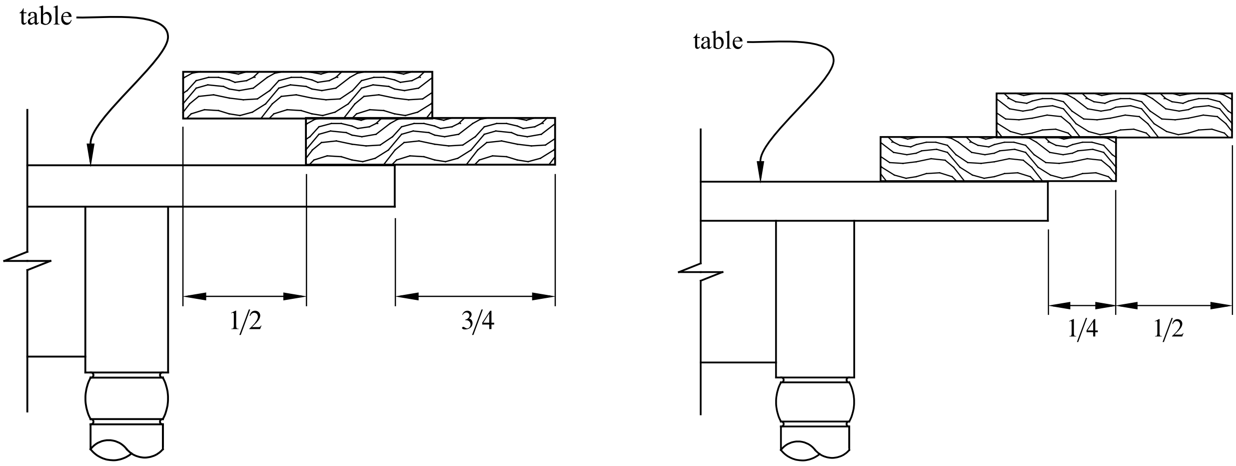
$$H_n = \ln(n) + \gamma + \frac{1}{2n} + \frac{1}{12n^2} + \frac{\epsilon(n)}{120n^4} \tag{2.4}$$

Here γ is a value $0.577215664\ldots$ called *Euler's constant*, and $\epsilon(n)$ is between 0 and 1 for all n. We will not prove this formula.

We are now finally done with our analysis of the book stacking problem. Plugging the value of H_n into (2.2), we find that the maximum overhang for n books is very close to $\ln(n)/2$. Since $\ln(n)$ grows to infinity as n increases, this means that if we are given enough books , we can get a book to hang out arbitrarily far over the edge of the table. Of course, the number of books we need will grow as an exponential function of the overhang; it will take 227 books just to achieve an overhang of 3, never mind an overhang of 100.

2.1.3 Extending Further Past the End of the Table

The overhang we analyzed above was the furthest out the *top* book could extend past the table. This leaves open the question of if there is some better way to build a stable stack where some book other than the top stuck out furthest. For example, Figure 2.5 shows a stable stack of two books where the bottom book extends further out than the top book. Moreover, the bottom book extends 3/4 of a book length past the end of the table, which is the same as the maximum overhang for the top book in a two book stack.



(a) **(b)**

Figure 2.5: Figure (a) shows a stable stack of two books where the bottom book extends the same amount past the end of the table as the maximum overhang two-book stack shown in Figure (b).

Since the two book arrangement in Figure 2.5(a) ties the maximum overhang stack in Figure 2.5(b), we could take the unique stable stack of n books where the top book extends furthest, and switch the top two books to look like Figure 2.5(a). This would give a stable stack of n books where the second from the top book extends the same maximum overhang distance. So for $n > 1$, there are at least two ways of building a stable stack of n books which both extend the maximum overhang distance—one way where the top book is furthest out, and another way where the second from the top book is furthest out.

It is not too hard to prove that these are the *only* two ways to get a stable stack of books that achieves maximum overhang, providing we stick to stacking only *one* book on top of another. But there is more to the story. Building book piles with more than one book resting on another—think of an inverted pyramid—it is possible to get a stack of n books to extend proportional to $\sqrt[3]{n}$—much more than $\ln n$—book lengths without falling over. See[1], *Maximum Overhang.*

2.1.4 Asymptotic Equality

For cases like equation 2.4 where we understand the growth of a function like H_n up to some (unimportant) error terms, we use a special notation, $\sim$, to denote the leading term of the function. For example, we say that $H_n \sim \ln(n)$ to indicate that the leading term of H_n is $\ln(n)$. More precisely:

Definition 22. For functions $f, g : \mathbb{R} \to \mathbb{R}$, we say f is*asymptotically equal* to g, in symbols,

$$f(x) \sim g(x)$$

iff

$$\lim_{x \to \infty} f(x)/g(x) = 1.$$

[1]Michael Paterson *et al.* “Maximum Overhang”. In: *MAA Monthly* 116 (2009), pp. 763–787.

Although it is tempting to write $H_n \sim \ln(n) + \gamma$ to indicate the two leading terms, this is not really right. According to Definition 22, $H_n \sim \ln(n) + c$ where c is *any constant*. The correct way to indicate that γ is the second-largest term is $H_n - \ln(n) \sim \gamma$.

The reason that the $\sim$ notation is useful is that often we do not care about lower order terms. For example, if $n = 100$, then we can compute $H(n)$ to great precision using only the two leading terms:

$$|H_n - \ln(n) - \gamma| \le \left| \frac{1}{200} - \frac{1}{120000} + \frac{1}{120 \cdot 100^4} \right| < \frac{1}{200}.$$

We will find your study of asymptotic notations in Section 1.6 quite useful in making sense of such simplifications.

Practice Problems

Exercise 2.1.1 There is a number a such that $\sum_{i=1}^{\infty} i^p$ converges iff $p < a$. What is the value of a? Prove it.

Hint: Find a value for a you think that works, then apply the integral bound.

Exercise 2.1.2 An infinite sum of nonnegative terms will converge to the same value—or diverge—no matter the order in which the terms are summed. This may not be true when there are an infinite number of both nonnegative and negative terms. An extreme example is

$$\sum_{i=0}^{\infty} (-1)^i = 1 + (-1) + 1 + (-1) + \cdots$$

because by regrouping the terms we can deduce:

$$\begin{aligned} [1 + (-1)] + [1 + (-1)] + \cdots &= 0 + 0 + \cdots = 0, \\ 1 + [(-1) + 1] + [(-1) + 1] + \cdots &= 1 + 0 + 0 + \cdots = 1. \end{aligned}$$

The problem here with this infinite sum is that the sum of the first n terms oscillates between 0 and 1, so the sum does not approach any limit.

But even for convergent sums, rearranging terms can cause big changes when the sum contains positive and negative terms. To illustrate the problem, we look at the Alternating Harmonic Series:

$$1 - 1/2 + 1/3 - 1/4 + \cdots \pm .$$

A standard result of elementary calculus, Tom M. Apostol. *Calculus, Vol. 1.* Wiley & Sons, 1967. ISBN: 0-471-00005-1, p.403, is that this series converges to $\ln 2$, but things change if we reorder the terms in the series.

Explain for example how to reorder terms in the Alternating Harmonic Series so that the reordered series converges to 7. Then explain how to reorder so it diverges.

Exercise 2.1.3 There is a bug on the edge of a 1-meter rug. The bug wants to cross to the other side of the rug. It crawls at 1 cm per second. However, at the end of each second, a malicious first-grader named Mildred Anderson *stretches* the rug by 1 meter. Assume that her action is instantaneous and the rug stretches uniformly. Thus, here's what happens in the first few seconds:

- The bug walks 1 cm in the first second, so 99 cm remain ahead.
- Mildred stretches the rug by 1 meter, which doubles its length. So now there are 2 cm behind the bug and 198 cm ahead.
- The bug walks another 1 cm in the next second, leaving 3 cm behind and 197 cm ahead.
- Then Mildred strikes, stretching the rug from 2 meters to 3 meters. So there are now $3 \cdot (3/2) = 4.5$ cm behind the bug and $197 \cdot (3/2) = 295.5$ cm ahead.

- The bug walks another 1 cm in the third second, and so on.

Your job is to determine this poor bug's fate.

(a) During second i, what *fraction* of the rug does the bug cross?

(b) Over the first n seconds, what fraction of the rug does the bug cross altogether? Express your answer in terms of the Harmonic number H_n.

(c) The known universe is thought to be about $3 \cdot 10^{10}$ light years in diameter. How many universe diameters must the bug travel to get to the end of the rug? (This distance is NOT the inflated distance caused by the stretching but only the actual walking done by the bug).

Exercise 2.1.4 Prove that the Alternating Harmonic Series

$$1 - 1/2 + 1/3 - 1/4 + \cdots \pm$$

converges.

Exercise 2.1.5 Show that $\sum_{i=1}^{\infty} i^p$ converges to a finite value iff $p < -1$.

Exercise 2.1.6 Suppose $f, g : \mathbb{N}^+ \to \mathbb{N}^+$ and $f \sim g$.

(a) Prove that $2f \sim 2g$.

(b) Prove that $f^2 \sim g^2$.

(c) Give examples of f and g such that $2^f \not\sim 2^g$.

2.2 Fibonacci Numbers

Now we come to a special sequence of numbers that is perhaps the most pleasant of all, the Fibonacci sequence $\langle F_n \rangle$:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
F_n	0	1	1	2	3	5	8	13	21	34	55	89	144	233	377

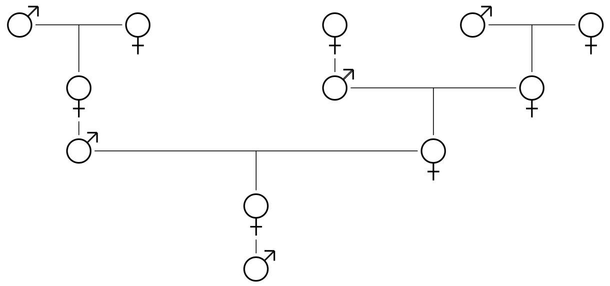
Unlike the harmonic numbers and the Bernoulli numbers, the “Fibonacci numbers" are nice simple integers. They are defined by the recurrence

$$\begin{aligned} F_0 &= 0\,; \\ F_1 &= 1\,; \\ F_n &= F_{n-1} + F_{n-2}\,, \qquad \text{for } n > 1. \end{aligned} \tag{2.5}$$

The simplicity of this rule—the simplest possible recurrence in which each number depends on the previous two—accounts for the fact that Fibonacci numbers occur in a wide variety of situations.

“Bee trees" provide a good example of how Fibonacci numbers can arise naturally. Let's consider the pedigree of a male bee. Each male (also known as a drone) is produced asexually from a female (also known as a queen); each female, however, has two parents, a male and a female. Here are the first few levels of the tree:

The "drone" has one grandfather and one grandmother; he has one great-grandfather and two great-grandmothers; he has two great-great-grandfathers and three great-great-grandmothers.

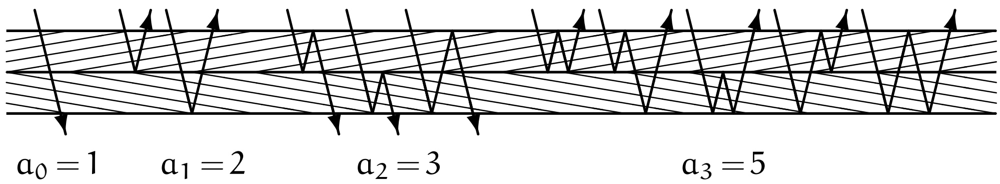


In general, it is easy to see by induction that he has exactly F_{n+1} great^n-grandpas and F_{n+2} great^n-grandmas.

Fibonacci numbers are often found in nature, perhaps for reasons similar to the bee-tree law. For example, a typical sunflower has a large head that contains spirals of tightly packed florets, usually with 34 winding in one direction and 55 in another.

Smaller heads will have 21 and 34, or 13 and 21; a gigantic “sunflower" with 89 and 144 spirals was once exhibited in England. Similar patterns are found in some species of pine cones.

And here’s an example of a different nature: Suppose we put two panes of glass back-to-back. How many ways a_n are there for light rays to pass through or be reflected after changing direction n times? The first few cases are:

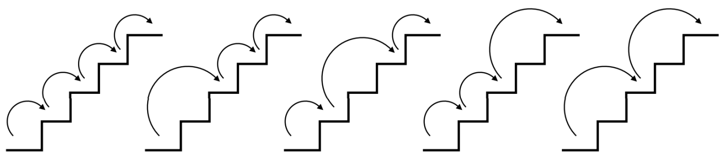


When n is even, we have an even number of bounces and the ray passes through; when n is odd, the ray is reflected and it re-emerges on the same side it entered. The a_n’s seem to be Fibonacci numbers, and a little staring at the figure tells us why: For $n \geq 2$, the n-bounce rays either take their first bounce off the opposite surface and continue in a_{n-1} ways, or they begin by bouncing off the middle surface and then bouncing back again to finish in a_{n-2} ways. Thus we have the Fibonacci recurrence $a_n = a_{n-1} + a_{n-2}$. The initial conditions are different, but not very different, because we have $a_0 = 1 = F_2$ and $a_1 = 2 = F_3$; therefore everything is simply shifted two places, and $a_n = F_{n+2}$.

As an exercise in a formulation that turns out to be a close analog of the above, consider this: How many different ways s_n are there to climb n stairs, if you can either step up one stair or hop up two?

For example, there are five different ways to climb four stairs:

1. step, step, step, step
2. hop, step, step
3. step, hop step
4. step, step, hop
5. hop, hop



As special cases, there is 1 way to climb 0 stairs (do nothing) and 1 way to climb 1 stair (step up). So $s_0 = 1$ and $s_1 = 1$. In general, an ascent of n stairs consists of either a step followed by an ascent of the remaining $n - 1$ stairs or a hop followed by an ascent of $n - 2$ stairs. So the total number of ways s_n to climb n stairs is equal to the number of ways to climb $n - 1$ plus the number of ways to climb $n - 2$. These observations define a recurrence $s_n = s_{n-1} + s_{n-2}$ for denoting the number of ways to climb n stairs.

Leonardo “Fibonacci" introduced these numbers in 1202, and mathematicians gradually began to discover more and more interesting things about them. Édouard “Lucas", the perpetrator of the Tower of Hanoi puzzle (discussed elsewhere), worked with them extensively in the last half of the nineteenth century (in fact it was Lucas who popularized the name “Fibonacci numbers”). One of his amazing results was to use properties of Fibonacci numbers to prove that the 39-digit “Mersenne number" $2^{127}-1$ is prime.

One of the oldest theorems about Fibonacci numbers, due to the French astronomer Jean-Dominique "Cassini" in 1680, is the identity

$$F_{n+1}F_{n-1}-F_n^2=(-1)^n\,,\qquad \text{for } n>0. \tag{2.6}$$

When $n=6$, for example, "Cassini's identity" correctly claims that $13\cdot 5-8^2=1$.

A polynomial formula that involves Fibonacci numbers of the form $F_{n\pm k}$ for small values of k can be transformed into a formula that involves only F_n and F_{n+1}, because we can use the rule

$$F_m=F_{m+2}-F_{m+1} \tag{2.7}$$

to express F_m in terms of higher Fibonacci numbers when $m<n$, and we can use

$$F_m=F_{m-2}+F_{m-1} \tag{2.8}$$

to replace F_m by lower Fibonacci numbers when $m>n+1$. Thus, for example, we can replace F_{n-1} by $F_{n+1}-F_n$ in (2.6) to get "Cassini's identity" in the form

$$F_{n+1}^2-F_{n+1}F_n-F_n^2=(-1)^n\,. \tag{2.9}$$

Moreover, Cassini's identity reads

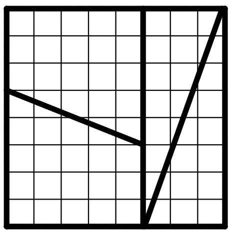
$$F_{n+2}F_n-F_{n+1}^2=(-1)^{n+1}$$

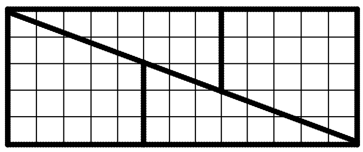
when n is replaced by $n+1$; this is the same as $(F_{n+1}+F_n)F_n-F_{n+1}^2=(-1)^{n+1}$, which is the same as (2.9). Thus Cassini(n) is true if and only if Cassini(n+1) is true; equation (2.6) holds for all n by induction.

Cassini's identity is the basis of a geometrical paradox that was one of Lewis "Carroll"'s favorite puzzles

The idea is to take a chessboard and cut it into four pieces as shown here, then to reassemble the pieces into a rectangle:

Presto: The original area of $8\times 8=64$ squares has been rearranged to yield $5\times 13=65$ squares!





A similar construction dissects any $F_n\times F_n$ square into four pieces, using F_{n+1}, F_n, F_{n-1}, and F_{n-2} as dimensions wherever the illustration has 13, 8, 5, and 3 respectively. The result is an $F_{n-1}\times F_{n+1}$ rectangle; by (2.6), one square has therefore been gained or lost, depending on whether n is even or odd.

Strictly speaking, we can't apply the reduction (2.8) unless $m\geq 2$, because we haven't defined F_n for negative n. A lot of maneuvering becomes easier if we eliminate this boundary condition and use (2.7) and (2.8) to define Fibonacci numbers with negative indices. For example, F_{-1} turns out to be $F_1-F_0=1$; then F_{-2} is $F_0-F_{-1}=-1$. In this way we deduce the values

n	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11
F_n	0	1	-1	2	-3	5	-8	13	-21	34	-55	89

and it quickly becomes clear (by induction) that

$$F_{-n}=(-1)^{n-1}F_n\,,\qquad \text{integer } n. \tag{2.10}$$

"Cassini's identity" (2.6) is true for *all* integers n, not just for $n > 0$, when we extend the Fibonacci sequence in this way.

The process of reducing $F_{n\pm k}$ to a combination of F_n and F_{n+1} by using (2.8) and (2.7) leads to the sequence of formulas

$$\begin{aligned} F_{n+2} &= F_{n+1} + F_n & F_{n-1} &= F_{n+1} - F_n \\ F_{n+3} &= 2F_{n+1} + F_n & F_{n-2} &= -F_{n+1} + 2F_n \\ F_{n+4} &= 3F_{n+1} + 2F_n & F_{n-3} &= 2F_{n+1} - 3F_n \\ F_{n+5} &= 5F_{n+1} + 3F_n & F_{n-4} &= -3F_{n+1} + 5F_n \end{aligned}$$

in which another pattern becomes obvious:

$$F_{n+k} = F_k F_{n+1} + F_{k-1} F_n \,. \tag{2.11}$$

This identity, easily proved by induction, holds for all integers k and n (positive, negative, or zero).

If we set $k = n$ in (2.11), we find that

$$F_{2n} = F_n F_{n+1} + F_{n-1} F_n \,; \tag{2.12}$$

hence F_{2n} is a multiple of F_n. Similarly,

$$F_{3n} = F_{2n} F_{n+1} + F_{2n-1} F_n \,, \tag{2.13}$$

and we may conclude that F_{3n} is also a multiple of F_n. By induction,

$$F_{kn} \text{ is a multiple of } F_n \,, \tag{2.14}$$

for all integers k and n. This explains, for example, why F_{15} (which equals 610) is a multiple of both F_3 and F_5 (which are equal to 2 and 5). Even more is true, in fact it can be shown[2] that

$$\gcd(F_m, F_n) = F_{\gcd(m,n)} \,. \tag{2.15}$$

For example, $\gcd(F_{12}, F_{18}) = \gcd(144, 2584) = 8 = F_6$.

We can now prove a converse of (2.14): If $n > 2$ and if F_m is a multiple of F_n, then m is a multiple of n. For if $F_n \mid F_m$ then $F_n \mid \gcd(F_m, F_n) = F_{\gcd(m,n)} \le F_n$. This is possible only if $F_{\gcd(m,n)} = F_n$; and our assumption that $n > 2$ makes it mandatory that $\gcd(m, n) = n$. Hence $n \mid m$.

An extension of these divisibility ideas was used by Yuri Matijasevich in his famous proof that there is no algorithm to decide if a given multivariate polynomial equation with integer coefficients has a solution in integers. Matijasevich's lemma states that, if $n > 2$, the Fibonacci number F_m is a multiple of F_n^2 if and only if m is a multiple of nF_n.

Let's prove this by looking at the sequence $\langle F_{kn} \bmod \mathrm{F_n^2} \rangle$ for $k = 1, 2, 3, \ldots$, and seeing when $F_{kn} \bmod \mathrm{F_n^2} = 0$. (We know that m must have the form kn if $F_m \bmod \mathrm{F_n} = 0$.) First we have $F_n \bmod \mathrm{F_n^2} = \mathrm{F_n}$; that's not zero. Next we have

$$F_{2n} = F_n F_{n+1} + F_{n-1} F_n \equiv 2F_n F_{n+1} \pmod{F_n^2}, \tag{2.16}$$

by (2.11), since $F_{n+1} \equiv F_{n-1} \pmod{F_n}$. Similarly

$$F_{2n+1} = F_{n+1}^2 + F_n^2 \equiv F_{n+1}^2 \pmod{F_n^2}. \tag{2.17}$$

This congruence allows us to compute

$$\begin{aligned} F_{3n} &= F_{2n+1} F_n + F_{2n} F_{n-1} \\ &\equiv F_{n+1}^2 F_n + (2F_n F_{n+1}) F_{n+1} = 3F_{n+1}^2 F_n \qquad \pmod{F_n^2}; \end{aligned}$$

[2]You are meant to put in some solid effort to come to terms with this rather convincingly as you would be asked to prove this in an exercise problem.

$$\begin{aligned} F_{3n+1} &= F_{2n+1}F_{n+1} + F_{2n}F_n \\ &\equiv F_{n+1}^3 + (2F_nF_{n+1})F_n \equiv F_{n+1}^3 \qquad \pmod{F_n^2}. \end{aligned}$$

In general, we find by induction on k that

$$F_{kn} \equiv kF_nF_{n+1}^{k-1} \text{ and } F_{kn+1} \equiv F_{n+1}^k \pmod{F_n^2}.$$

Now F_{n+1} is relatively prime to F_n, so

$$\begin{aligned} F_{kn} \equiv 0 \pmod{F_n^2} &\longleftrightarrow kF_n \equiv 0 \pmod{F_n^2} \\ &\longleftrightarrow k \equiv 0 \pmod{F_n}. \end{aligned}$$

We have proved Matijasevich's lemma.

One of the most important properties of the Fibonacci numbers is the special way in which they can be used to represent integers. Let's write

$$j \gg k \qquad \longleftrightarrow \qquad j \geq k+2. \tag{2.18}$$

Then *every positive integer has a unique representation of the form*

$$n = F_{k_1} + F_{k_2} + \cdots + F_{k_r}, \qquad k_1 \gg k_2 \gg \cdots \gg k_r \gg 0. \tag{2.19}$$

(This is "Zeckendorf's theorem".) For example, the representation of one million turns out to be

$$\begin{aligned} 1000000 &= 832040 + 121393 + 46368 + 144 + 55 \\ &= F_{30} + F_{26} + F_{24} + F_{12} + F_{10}. \end{aligned}$$

We can always find such a representation by using a "greedy" approach, choosing F_{k_1} to be the largest Fibonacci number $\leq n$, then choosing F_{k_2} to be the largest that is $\leq n - F_{k_1}$, and so on. (More precisely, suppose that $F_k \leq n < F_{k+1}$; then we have $0 \leq n - F_k < F_{k+1} - F_k = F_{k-1}$. If n is a Fibonacci number, (2.19) holds with $r = 1$ and $k_1 = k$. Otherwise $n - F_k$ has a Fibonacci representation $F_{k_2} + \cdots + F_{k_r}$, by induction on n; and (2.19) holds if we set $k_1 = k$, because the inequalities $F_{k_2} \leq n - F_k < F_{k-1}$ imply that $k \gg k_2$.)

Conversely, any representation of the form (2.19) implies that

$$F_{k_1} \leq n < F_{k_1+1},$$

because the largest possible value of $F_{k_2} + \cdots + F_{k_r}$ when $k \gg k_2 \gg \cdots \gg k_r \gg 0$ is

$$F_{k-2} + F_{k-4} + \cdots + F_{k \bmod 2 + 2} = F_{k-1} - 1, \qquad \text{if } k \geq 2. \tag{2.20}$$

(This formula is easy to prove by induction on k; the left-hand side is zero when k is 2 or 3.) Therefore k_1 is the greedily chosen value described earlier, and the representation must be unique.

Any unique system of representation is a "number system"; therefore Zeckendorf's theorem leads to the *"Fibonacci number system"*. We can represent any nonnegative integer n as a sequence of 0's and 1's, writing

$$n = (b_m b_{m-1} \ldots b_2)_F \longleftrightarrow n = \sum_{k=2}^{m} b_k F_k. \tag{2.21}$$

This number system is something like binary (radix 2) notation, except that there never are two adjacent 1's. For example, here are the numbers from 1 to 20, expressed Fibonacci-wise:

$$\begin{array}{llll} 1 = (000001)_F & 6 = (001001)_F & 11 = (010100)_F & 16 = (100100)_F \\ 2 = (000010)_F & 7 = (001010)_F & 12 = (010101)_F & 17 = (100101)_F \\ 3 = (000100)_F & 8 = (010000)_F & 13 = (100000)_F & 18 = (101000)_F \\ 4 = (000101)_F & 9 = (010001)_F & 14 = (100001)_F & 19 = (101001)_F \\ 5 = (001000)_F & 10 = (010010)_F & 15 = (100010)_F & 20 = (101010)_F \end{array}$$

The Fibonacci representation of a million, shown a minute ago, can be contrasted with its binary representation $2^{19} + 2^{18} + 2^{17} + 2^{16} + 2^{14} + 2^9 + 2^6$:

$$\begin{aligned}(1000000)_{10} &= (10001010000000000010100000000)_F \\ &= (11110100001001000000)_2 \,.\end{aligned}$$

The Fibonacci representation needs a few more bits because adjacent 1's are not permitted; but the two representations are analogous.

To add 1 in the Fibonacci number system, there are two cases: If the "units digit" is 0, we change it to 1; that adds $F_2 = 1$, since the units digit refers to F_2. Otherwise the two least significant digits will be 01, and we change them to 10 (thereby adding $F_3 - F_2 = 1$). Finally, we must "carry" as much as necessary by changing the digit pattern '011' to '100' until there are no two 1's in a row. (This carry rule is equivalent to replacing $F_{m+1} + F_m$ by F_{m+2}.) For example, to go from $5 = (1000)_F$ to $6 = (1001)_F$ or from $6 = (1001)_F$ to $7 = (1010)_F$ requires no carrying; but to go from $7 = (1010)_F$ to $8 = (10000)_F$ we must carry twice.

So far we've been discussing lots of properties of the Fibonacci numbers, but we haven't come up with a closed formula for them. No closed forms are known for such special numbers as Stirling numbers, Eulerian numbers, or Bernoulli numbers, whereas we were able to obtain something akin to the closed form for harmonic numbers H_n. Is there a relation between F_n and other quantities we know? Can we "solve" the recurrence that defines F_n?

We are going to find out soon.

Practice Problems

Exercise 2.2.1 An explorer has left a pair of baby "rabbits" on an island. If baby rabbits become adults after one month, and if each pair of adult rabbits produces one pair of baby rabbits every month, how many pairs of rabbits are present after n months? (After two months there are two pairs, one of which is newborn.) Find a connection between this problem and the "bee tree" in the text.

Exercise 2.2.2 Show that "Cassini's identity" (2.6) is a special case of (2.11).

Exercise 2.2.3 Use the "Fibonacci number system" to convert 65 mile/hr into an approximate number of km/hr.

Exercise 2.2.4 Prove the gcd law (2.15) for Fibonacci numbers.

3 | Cardinality Rules for Counting

The present chapter describes the most basic rules for determining the cardinality of a set. These rules are actually theorems, but our focus here will be less on their proofs than on teaching their use in simple counting as a practical skill, like integration.

But counting can be tricky, and people make counting mistakes all the time, so a crucial part of counting skill is being able to verify a counting argument. Sometimes this can be done simply by finding an alternative way to count and then comparing answers—they better agree. But most elementary counting arguments reduce to finding a bijection between objects to be counted and easy-to-count sequences. The chapter shows how explicitly defining these bijections—and verifying that they are bijections—is another useful way to verify counting arguments. The material presented here is simple yet powerful, and it provides a great tool set for use in analyzing a myriad of problems in *computer science.*

3.1 Counting One Thing by Counting Another

How do you count the number of people in a crowded room? You could count heads, since for each person there is exactly one head. Alternatively, you could count ears and divide by two. Of course, you might have to adjust the calculation if someone lost an ear in a pirate raid or someone was born with three ears. The point here is that you can often *count one thing by counting another*, though some fudging may be required. This is a central theme of counting, from the easiest problems to the hardest.

3.1.1 The Bijection Rule

The most direct way to count one thing by counting another is to find a bijection between them, since if there is a bijection between two sets, then the sets have the same size. This important fact is commonly known as the *Bijection Rule.*

Rule 23 (Bijection Rule)**.** *If there is a bijection $f : A \to B$ between A and B, then $|A| = |B|$.*

As an application of the bijection rule 23 for the special case of the finite sets, we can give an easy proof of:

Theorem 24. *There are 2^n subsets[1] of an n-element set. That is,*

$$|A| = n \quad \textit{implies} \quad |\operatorname{pow}(A)| = 2^n.$$

For example, the three-element set $\{a_1, a_2, a_3\}$ has eight different subsets:

$$\begin{array}{cccc} \emptyset & \{a_1\} & \{a_2\} & \{a_1, a_2\} \\ \{a_3\} & \{a_1, a_3\} & \{a_2, a_3\} & \{a_1, a_2, a_3\} \end{array}$$

Theorem 24 follows from the fact that there is a simple bijection from subsets of A to $\{0,1\}^n$, the n-bit sequences. Namely, let $a_1, a_2, \ldots, a_n$ be the elements of A. The bijection maps each subset of $S \subseteq A$ to the bit sequence $(b_1, \ldots, b_n)$ defined by the rule that

$$b_i = 1 \quad \text{iff} \quad a_i \in S.$$

[1]Recall that the power set $\operatorname{pow}(A)$ denotes the set of all subsets of A.

For example, if $n = 10$, then the subset $\{a_2, a_3, a_5, a_7, a_{10}\}$ maps to a 10-bit sequence as follows:

$$\begin{array}{rcccccccccccc} \text{subset:} & \{ & & a_2, & a_3, & & a_5, & & a_7, & & & a_{10} & \} \\ \text{sequence:} & (& 0, & 1, & 1, & 0, & 1, & 0, & 1, & 0, & 0, & 1 &) \end{array}$$

Now by bijection rule 23,

$$|\operatorname{pow}(A)| = |\{0,1\}^n|.$$

But every computer scientist knows[2] that there are 2^n n-bit sequences! So we've proved Theorem 24!

The Bijection Rule acts as a magnifier of counting ability; if you figure out the size of one set, then you can immediately determine the sizes of many other sets via bijections. For example, let's look at the following two sets:

$$\begin{aligned} A &= \text{all ways to select a dozen donuts when five varieties are available} \\ B &= \text{all 16-bit sequences with exactly 4 ones} \end{aligned}$$

An example of an element of set A is:

$$\underbrace{0\,0}_{\text{chocolate}} \quad \underbrace{}_{\text{lemon-filled}} \quad \underbrace{0\,0\,0\,0\,0\,0}_{\text{sugar}} \quad \underbrace{0\,0}_{\text{glazed}} \quad \underbrace{0\,0}_{\text{plain}}$$

Here, we've depicted each donut with a 0 and left a gap between the different varieties. Thus, the selection above contains two chocolate donuts, no lemon-filled, six sugar, two glazed, and two plain. Now let's put a 1 into each of the four gaps:

$$\underbrace{0\,0}_{\text{chocolate}} \quad 1 \quad \underbrace{}_{\text{lemon-filled}} \quad 1 \quad \underbrace{0\,0\,0\,0\,0\,0}_{\text{sugar}} \quad 1 \quad \underbrace{0\,0}_{\text{glazed}} \quad 1 \quad \underbrace{0\,0}_{\text{plain}}$$

and close up the gaps:

$$0011000000100100\,.$$

We've just formed a 16-bit number with exactly 4 ones—an element of B!

This example suggests a bijection from set A to set B: map a dozen donuts consisting of:

$$c \text{ chocolate, } l \text{ lemon-filled, } s \text{ sugar, } g \text{ glazed, and } p \text{ plain}$$

to the sequence:

$$\underbrace{0\ldots0}_{c} \quad 1 \quad \underbrace{0\ldots0}_{l} \quad 1 \quad \underbrace{0\ldots0}_{s} \quad 1 \quad \underbrace{0\ldots0}_{g} \quad 1 \quad \underbrace{0\ldots0}_{p}$$

The resulting sequence always has 16 bits and exactly 4 ones, and thus is an element of B. Moreover, the mapping is a bijection: every such bit sequence comes from exactly one order of a dozen donuts. Therefore, $|A| = |B|$ by the Bijection Rule. More generally,

Lemma 25. *The number of ways to select n donuts when k flavors are available is the same as the number of binary sequences with exactly n zeroes and $k - 1$ ones.*

This example demonstrates the power of the bijection rule. We managed to prove that two very different sets are actually the same size—even though we don't know exactly how big either one is. But as soon as we figure out the size of one set, we'll immediately know the size of the other.

This particular bijection might seem frighteningly ingenious if you've not seen it before. But you'll use essentially this same argument over and over, and soon you'll consider it routine.

---

[2] In case you're someone who doesn't know how many n-bit sequences there are, you'll find the 2^n explained in Section 3.2.2.

3.2 Counting Sequences

The Bijection Rule lets us count one thing by counting another. This suggests a general strategy: get really good at counting just a few things, then use bijections to count everything else! This is the strategy we'll follow. In particular, we'll get really good at counting *sequences*. When we want to determine the size of some other set T, we'll find a bijection from T to a set of sequences S. Then we'll use our super-ninja sequence-counting skills to determine $|S|$, which immediately gives us $|T|$. We'll need to hone this idea somewhat as we go along, but that's pretty much it!

3.2.1 The Product Rule

The *Product Rule* gives the size of a product of sets. Recall that if $P_1, P_2, \ldots, P_n$ are sets, then

$$P_1 \times P_2 \times \cdots \times P_n$$

is the set of all sequences whose first term is drawn from P_1, second term is drawn from P_2 and so forth.

Rule 26 (Product Rule)**.** *If $P_1, P_2, \ldots P_n$ are finite sets, then:*

$$|P_1 \times P_2 \times \cdots \times P_n| = |P_1| \cdot |P_2| \cdots |P_n|$$

For example, suppose a *daily diet* consists of a breakfast selected from set B, a lunch from set L, and a dinner from set D where:

$$\begin{aligned} B &= \{\text{pancakes}, \text{bacon and eggs}, \text{bagel}, \text{Doritos}\} \\ L &= \{\text{burger and fries}, \text{garden salad}, \text{Doritos}\} \\ D &= \{\text{macaroni}, \text{pizza}, \text{frozen burrito}, \text{pasta}, \text{Doritos}\} \end{aligned}$$

Then $B \times L \times D$ is the set of all possible daily diets. Here are some sample elements:

$$\begin{gathered} (\text{pancakes}, \text{burger and fries}, \text{pizza}) \\ (\text{bacon and eggs}, \text{garden salad}, \text{pasta}) \\ (\text{Doritos}, \text{Doritos}, \text{frozen burrito}) \end{gathered}$$

The Product Rule tells us how many different daily diets are possible:

$$\begin{aligned} |B \times L \times D| &= |B| \cdot |L| \cdot |D| \\ &= 4 \cdot 3 \cdot 5 \\ &= 60. \end{aligned}$$

3.2.2 Subsets of an n-element Set

The fact that there are 2^n subsets of an n-element set was proved in Theorem 24 by setting up a bijection between the subsets and the length-n bit-strings. So the original problem about subsets was tranformed into a question about sequences—*exactly according to plan*! Now we can fill in the missing explanation of why there are 2^n length-n bit-strings: we can write the set of all n-bit sequences as a product of sets:

$$\{0,1\}^n \triangleq \underbrace{\{0,1\} \times \{0,1\} \times \cdots \times \{0,1\}}_{n \text{ terms}}.$$

Then Product Rule gives the answer:

$$|\{0,1\}^n| = |\{0,1\}|^n = 2^n.$$

3.2.3 The Sum Rule

There are 55 humid days, 45 hot days and 38 dusty days a year in your climate. On how many days you would encounter uncomfortable weather? Let set M be the humid days, T be the hot days, and D be the dusty days. In these terms, the answer to the question is $|M \cup T \cup D|$. Now assuming that each days is categorized into at most one weather type, the size of this union of sets is given by the *Sum Rule*:

Rule 27 (Sum Rule). *If $A_1, A_2, \ldots, A_n$ are* disjoint *sets, then:*

$$|A_1 \cup A_2 \cup \ldots \cup A_n| = |A_1| + |A_2| + \ldots + |A_n|$$

Thus, according to the climate statistics, you can be in uncomfortable weather for:

$$\begin{aligned}|M \cup T \cup D| &= |M| + |T| + |D| \\ &= 55 + 45 + 38 \\ &= 138 \text{ days}\end{aligned}$$

Notice that the Sum Rule holds only for a union of *disjoint* sets. Finding the size of a union of overlapping sets is a more complicated problem that we'll take up in Section 3.9.

3.2.4 Counting Passwords

Few counting problems can be solved with a single rule. More often, a solution is a flurry of sums, products, bijections, and other methods.

For solving problems involving passwords, telephone numbers, and license plates, the sum and product rules are useful together. For example, on a certain computer system, a valid password is a sequence of between six and eight symbols. The first symbol must be a letter (which can be lowercase or uppercase), and the remaining symbols must be either letters or digits. How many different passwords are possible?

Let's define two sets, corresponding to valid symbols in the first and subsequent positions in the password.

$$\begin{aligned}F &= \{a, b, \ldots, z, A, B, \ldots, Z\} \\ S &= \{a, b, \ldots, z, A, B, \ldots, Z, 0, 1, \ldots, 9\}\end{aligned}$$

In these terms, the set of all possible passwords is:[3]

$$(F \times S^5) \cup (F \times S^6) \cup (F \times S^7)$$

Thus, the length-six passwords are in the set $F \times S^5$, the length-seven passwords are in $F \times S^6$, and the length-eight passwords are in $F \times S^7$. Since these sets are disjoint, we can apply the Sum Rule and count the total number of possible passwords as follows:

$$\begin{aligned}&|(F \times S^5) \cup (F \times S^6) \cup (F \times S^7)| \\ &\quad = |F \times S^5| + |F \times S^6| + |F \times S^7| && \text{Sum Rule} \\ &\quad = |F| \cdot |S|^5 + |F| \cdot |S|^6 + |F| \cdot |S|^7 && \text{Product Rule} \\ &\quad = 52 \cdot 62^5 + 52 \cdot 62^6 + 52 \cdot 62^7 \\ &\quad \approx 1.8 \cdot 10^{14} \text{ different passwords.}\end{aligned}$$

[3]The notation S^5 means $S \times S \times S \times S \times S$.

Practice Problems

Exercise 3.2.1 Alice is thinking of a number between 1 and 1000.

What is the least number of yes/no questions you could ask her and be guaranteed to discover what it is? (Alice always answers truthfully.)

Exercise 3.2.2 In how many different ways is it possible to answer the next chapter's practice problems if:

- the first problem has four *true/false* questions,
- the second problem requires choosing one of four alternatives, and
- the answer to the third problem is an integer ≥ 15 and ≤ 20?

Exercise 3.2.3 How many total functions are there from set A to set B if $|A| = 3$ and $|B| = 7$?

Exercise 3.2.4 Let X be the six element set $\{x_1, x_2, x_3, x_4, x_5, x_6\}$.

(a) How many subsets of X contain x_1?

(b) How many subsets of X contain x_2 and x_3 but do not contain x_6?

Exercise 3.2.5 A license plate consists of either:

- 3 letters followed by 3 digits (standard plate)
- 5 letters (vanity plate)
- 2 characters—letters or numbers (big shot plate)

Let L be the set of all possible license plates.

(a) Express L in terms of

$$\mathcal{A} = \{A, B, C, \ldots, Z\}$$
$$\mathcal{D} = \{0, 1, 2, \ldots, 9\}$$

using unions ($\cup$) and set products ($\times$).

(b) Compute $|L|$, the number of different license plates, using the sum and product rules.

Exercise 3.2.6 Answer the followings.

(a) How many of the billion numbers in the integer interval $[1..10^9]$ contain the digit 1? (*Hint:* How many don't?)

(b) There are 20 books arranged in a row on a shelf. Describe a bijection between ways of choosing 6 of these books so that no two adjacent books are selected, and 15-bit strings with exactly 6 ones.

Exercise 3.2.7 Answer the followings.

(a) Let $\mathcal{S}_{n,k}$ be the possible nonnegative integer solutions to the inequality

$$x_1 + x_2 + \cdots + x_k \leq n. \tag{3.1}$$

That is $\mathcal{S}_{n,k} \triangleq \{(x_1, x_2, \ldots, x_k) \in \mathbb{N}^k \mid (3.1) \text{ is true}\}$.

Describe a bijection between $\mathcal{S}_{n,k}$ and the set of binary strings with n zeroes and k ones.

(b) Let $\mathcal{L}_{n,k}$ be the length k weakly increasing sequences of nonnegative integers $\leq n$. That is $\mathcal{L}_{n,k} \triangleq \{(y_1, y_2, \ldots, y_k) \in \mathbb{N}^k \mid y_1 \leq y_2 \leq \cdots \leq y_k \leq n\}$.

Describe a bijection between $\mathcal{L}_{n,k}$ and $\mathcal{S}_{n,k}$.

Exercise 3.2.8 The following problems give you an idea of how many relations and functions there are given two sets.

Let X and Y be finite sets.

(a) How many binary relations from X to Y are there?

(b) Define a bijection between the set $[X \to Y]$ of all total functions from X to Y and the set $Y^{|X|}$. (Recall Y^n is the Cartesian product of Y with itself n times.) Based on that, what is $|\,[X \to Y]\,|$?

(c) Using the previous part, how many *functions*, not necessarily total, are there from X to Y? How does the fraction of functions vs. total functions grow as the size of X grows? Is it $O(1)$, $O(|X|)$, $O(2^{|X|})$,...?

(d) Show a bijection between the powerset $\text{pow}(X)$ and the set $[X \to \{0, 1\}]$ of 0-1-valued total functions on X.

(e) Let X be a set of size n and B_X be the set of all bijections from X to X. Describe a bijection from B_X to the set of permutations of X.[4] This implies that there are how many bijections from X to X?

3.3 The Generalized Product Rule

In how many ways can, say, a Nobel prize, a Japan prize, and a Pulitzer prize be awarded to n people? This is easy to answer using our strategy of translating the problem about awards into a problem about sequences. Let P be the set of n people taking the course. Then there is a bijection from ways of awarding the three prizes to the set $P^3 \triangleq P \times P \times P$. In particular, the assignment:

"Barack wins a Nobel, George wins a Japan, and Bill wins a Pulitzer prize"

maps to the sequence $(\text{Barack}, \text{George}, \text{Bill})$. By the Product Rule, we have $|P^3| = |P|^3 = n^3$, so there are n^3 ways to award the prizes to a class of n people. Notice that P^3 includes triples like $(\text{Barack}, \text{Bill}, \text{Barack})$ where one person wins more than one prize.

But what if the three prizes must be awarded to *different* students? As before, we could map the assignment to the triple $(\text{Bill}, \text{George}, \text{Barack}) \in P^3$. But this function is *no longer a bijection.* For example, no valid assignment maps to the triple $(\text{Barack}, \text{Bill}, \text{Barack})$ because now we're not allowing Barack to receive two prizes. However, there *is* a bijection from prize assignments to the set:

$$S = \{(x, y, z) \in P^3 \mid x,\ y \text{ and } z \text{ are different people}\}$$

This reduces the original problem to a problem of counting sequences. Unfortunately, the Product Rule does not apply directly to counting sequences of this type because the entries depend on one another; in particular, they must all be different. However, a slightly sharper tool does the trick.

Rule 28 (*Generalized Product Rule*)**.** *Let S be a set of length-k sequences. If there are:*

[4] A sequence in which all the elements of a set X appear exactly once is called a *permutation* of X (see Section 3.3.3).

- n_1 *possible first entries,*
- n_2 *possible second entries for each first entry,*$\vdots$
- n_k *possible kth entries for each sequence of first* $k-1$ *entries,*

then:

$$|S| = n_1 \cdot n_2 \cdot n_3 \cdots n_k$$

In the awards example, S consists of sequences (x, y, z). There are n ways to choose x, the recipient of prize #1. For each of these, there are $n-1$ ways to choose y, the recipient of prize #2, since everyone except for person x is eligible. For each combination of x and y, there are $n-2$ ways to choose z, the recipient of prize #3, because everyone except for x and y is eligible. Thus, according to the Generalized Product Rule, there are

$$|S| = n \cdot (n-1) \cdot (n-2)$$

ways to award the 3 prizes to different people.

3.3.1 Defective Dollar Bills

A dollar bill (*i.e.*, note) is *defective* if some digit appears more than once in the 8-digit serial number. If you check your wallet, you'll be sad to discover that defective bills are all-too-common. In fact, how common are *nondefective* bills? Assuming that the digit portions of serial numbers all occur equally often, we could answer this question by computing

$$\begin{aligned}&\text{fraction of nondefective bills}\\ &= \frac{|\{\text{ serial \#'s with all digits different}\}|}{|\{\text{serial numbers}\}|}\end{aligned} \tag{3.2}$$

Let's first consider the denominator. Here there are no restrictions; there are 10 possible first digits, 10 possible second digits, 10 third digits, and so on. Thus, the total number of 8-digit serial numbers is 10^8 by the Product Rule.

Next, let's turn to the numerator. Now we're not permitted to use any digit twice. So there are still 10 possible first digits, but only 9 possible second digits, 8 possible third digits, and so forth. Thus, by the Generalized Product Rule, there are

$$10 \cdot 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 = \frac{10!}{2} = 1{,}814{,}400$$

serial numbers with all digits different. Plugging these results into Equation 3.2, we find:

$$\text{fraction of nondefective bills} = \frac{1{,}814{,}400}{100{,}000{,}000} = 1.8144\%$$

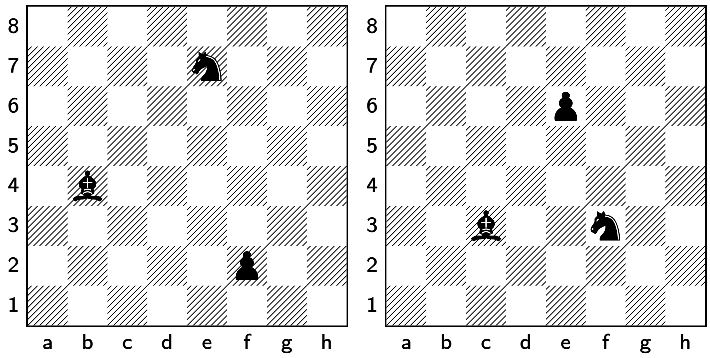
3.3.2 A Chess Problem

In how many different ways can we place a pawn (P), a knight (N), and a bishop (B) on a chessboard so that no two pieces share a row or a column? A valid and an invalid configuration are shown in Figure 3.1.

First, we map this problem about chess pieces to a question about sequences. There is a bijection from configurations to sequences

$$(r_P, c_P, r_N, c_N, r_B, c_B)$$

where r_P, r_N and r_B are distinct rows and c_P, c_N and c_B are distinct columns. In particular, r_P is the pawn's row c_P is the pawn's column r_N is the knight's row, etc. Now we can count the number of such sequences using the Generalized Product Rule:



(a) valid **(b)** invalid

Figure 3.1: Two ways of placing a pawn (♙), a knight (♘), and a bishop (♗) on a chessboard. The configuration shown in (b) is invalid because the bishop and the knight are in the same row.

- r_P is one of 8 rows
- c_P is one of 8 columns
- r_N is one of 7 rows (any one but r_P)
- c_N is one of 7 columns (any one but c_P)
- r_B is one of 6 rows (any one but r_P or r_N)
- c_B is one of 6 columns (any one but c_P or c_N)

Thus, the total number of configurations is $(8 \cdot 7 \cdot 6)^2$.

3.3.3 Permutations

A *permutation* of a set S is a sequence that contains every element of S exactly once. For example, here are all the permutations of the set $\{a, b, c\}$:

$$\begin{array}{ccc} (a, b, c) & (a, c, b) & (b, a, c) \\ (b, c, a) & (c, a, b) & (c, b, a) \end{array}$$

How many permutations of an n-element set are there? Well, there are n choices for the first element. For each of these, there are $n - 1$ remaining choices for the second element. For every combination of the first two elements, there are $n - 2$ ways to choose the third element, and so forth. Thus, there are a total of

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

permutations of an n-element set. In particular, this formula says that there are $3! = 6$ permutations of the 3-element set $\{a, b, c\}$, which is the number we found above.

Permutations will come up again in this course approximately 1.6 bazillion times. In fact, permutations are the reason why factorial comes up so often and why we taught you Stirling's approximation:

$$n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n .$$

3.4 The Division Rule

Counting ears and dividing by two is a silly way to count the number of people in a room, but this approach is representative of a powerful counting principle.

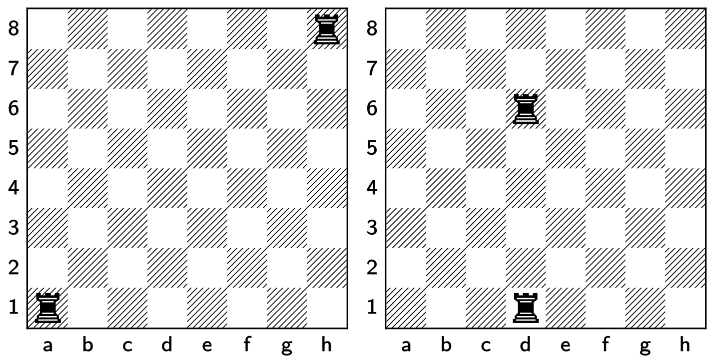
A *k-to-1 function* maps exactly k elements of the domain to every element of the codomain. For example, the function mapping each ear to its owner is 2-to-1. Similarly, the function mapping each finger to its owner is 10-to-1, and the function mapping each finger and toe to its owner is 20-to-1. The general rule is:

Rule 29 (Division Rule)**.** *If $f : A \to B$ is k-to-1, then $|A| = k \cdot |B|$.*

For example, suppose A is the set of ears in the room and B is the set of people. There is a 2-to-1 mapping from ears to people, so by the Division Rule, $|A| = 2 \cdot |B|$. Equivalently, $|B| = |A|/2$, expressing what we knew all along: the number of people is half the number of ears. Unlikely as it may seem, many counting problems are made much easier by initially counting every item multiple times and then correcting the answer using the Division Rule. Let's look at some examples.

3.4.1 Another Chess Problem

In how many different ways can you place two identical rooks on a chessboard so that they do not share a row or column? A valid and an invalid configuration are shown in Figure 3.2.



(a) valid **(b)** invalid

Figure 3.2: Two ways to place 2 rooks (♖) on a chessboard. The configuration in (b) is invalid because the rooks are in the same column.

Let A be the set of all sequences

$$(r_1, c_1, r_2, c_2)$$

where r_1 and r_2 are distinct rows and c_1 and c_2 are distinct columns. Let B be the set of all valid rook configurations. There is a natural function f from set A to set B; in particular, f maps the sequence (r_1, c_1, r_2, c_2) to a configuration with one rook in row r_1, column c_1 and the other rook in row r_2, column c_2.

But now there's a snag. Consider the sequences:

$$(1, a, 8, h) \qquad \text{and} \qquad (8, h, 1, a)$$

The first sequence maps to a configuration with a rook in the lower-left corner and a rook in the upper-right corner. The second sequence maps to a configuration with a rook in the upper-right corner and a rook in the lower-left corner. The problem is that those are two different ways of describing the *same* configuration! In fact, this arrangement is shown in Figure 3.2(a).

More generally, the function f maps exactly two sequences to *every* board configuration; f is a 2-to-1 function. Thus, by the quotient rule, $|A| = 2 \cdot |B|$. Rearranging terms gives:

$$|B| = \frac{|A|}{2} = \frac{(8 \cdot 7)^2}{2}.$$

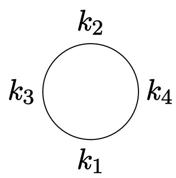
In the second equality, we've computed the size of A using the General Product Rule just as in the earlier chess problem.

3.4.2 Knights of the Round Table

In how many ways can King Arthur arrange to seat his n different knights at his round table? A seating defines who sits where. Two seatings are considered to be the same *arrangement* if each knight sits with the same knight on his left in both seatings. An equivalent way to say this is that two seatings yield the same arrangement when they yield the same sequence of knights starting at knight number 1 and going clockwise around the table. For example, the following two seatings determine the same arrangement:

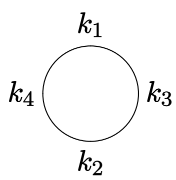
A seating is determined by the sequence of knights going clockwise around the table starting at the top seat. So seatings correspond to permutations of the knights, and there are $n!$ of them. For example,

$(k_2, k_4, k_1, k_3) \longrightarrow$



Two seatings determine the same arrangement if they are the same when the table is rotated so knight 1 is at the top seat. For example with $n = 4$, there are 4 different sequences that correspond to the seating arrangement:

$$\begin{array}{c} (k_2, k_4, k_1, k_3) \\ (k_4, k_1, k_3, k_2) \\ (k_1, k_3, k_2, k_4) \\ (k_3, k_2, k_4, k_1) \end{array} \longrightarrow$$



This mapping from seating to arrangments is actually an n-to-1 function, since all n cyclic shifts of the sequence of knights in the seating map to the same arrangement. Therefore, by the division rule, the number of circular seating arrangements is:

$$\frac{\#\,\text{seatings}}{n} = \frac{n!}{n} = (n-1)!\,.$$

Practice Problems

Exercise 3.4.1 Use induction to prove that there are 2^n subsets of an n-element set.

3.5 Counting Subsets

How many k-element subsets of an n-element set are there? This question arises all the time in various guises:

- In how many ways can I select 5 books from my collection of 100 to bring on vacation?
- How many different 13-card bridge hands can be dealt from a 52-card deck?
- In how many ways can I select 5 toppings for my pizza if there are 14 available toppings?

This number comes up so often that there is a special notation for it:

$$\binom{n}{k} \triangleq \text{the number of } k\text{-element subsets of an } n\text{-element set.}$$

The expression $\binom{n}{k}$ is read "n choose k." Now we can immediately express the answers to all three questions above:

- I can select 5 books from 100 in $\binom{100}{5}$ ways.
- There are $\binom{52}{13}$ different bridge hands.
- There are $\binom{14}{5}$ different 5-topping pizzas, if 14 toppings are available.

3.5.1 The Subset Rule

We can derive a simple formula for the n choose k number using the Division Rule. We do this by mapping any permutation of an n-element set $\{a_1, \ldots, a_n\}$ into a k-element subset simply by taking the first k elements of the permutation. That is, the permutation $a_1 a_2 \ldots a_n$ will map to the set $\{a_1, a_2, \ldots, a_k\}$.

Notice that any other permutation with the same first k elements $a_1, \ldots, a_k$ *in any order* and the same remaining elements $n-k$ elements *in any order* will also map to this set. What's more, a permutation can only map to $\{a_1, a_2, \ldots, a_k\}$ if its first k elements are the elements $a_1, \ldots, a_k$ in some order. Since there are $k!$ possible permutations of the first k elements and $(n-k)!$ permutations of the remaining elements, we conclude from the Product Rule that exactly $k!(n-k)!$ permutations of the n-element set map to the particular subset S. In other words, the mapping from permutations to k-element subsets is $k!(n-k)!$-to-1.

But we know there are $n!$ permutations of an n-element set, so by the Division Rule, we conclude that

$$n! = k!(n-k)!\binom{n}{k}$$

which proves:

Rule 30 (Subset Rule)**.** *The number of k-element subsets of an n-element set is*

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

Notice that this works even for 0-element subsets: $n!/0!n! = 1$. Here we use the fact that $0!$ is a *product* of 0 terms, which by convention[5] equals 1.

[5] We don't use it here, but a *sum* of zero terms equals 0.

3.5.2 Bit Sequences

How many n-bit sequences contain exactly k ones? We've already seen the straightforward bijection between subsets of an n-element set and n-bit sequences. For example, here is a 3-element subset of $\{x_1, x_2, \ldots, x_8\}$ and the associated 8-bit sequence:

$$\begin{array}{cccccccccc} \{ & x_1, & & & x_4, & x_5 & & & & \} \\ (& 1, & 0, & 0, & 1, & 1, & 0, & 0, & 0 &) \end{array}$$

Notice that this sequence has exactly 3 ones, each corresponding to an element of the 3-element subset. More generally, the n-bit sequences corresponding to a k-element subset will have exactly k ones. So by the Bijection Rule,

Corollary 31. *The number of n-bit sequences with exactly k ones is* $\binom{n}{k}$.

Also, the bijection between selections of flavored donuts and bit sequences of Lemma 25 now implies,

Corollary 32. *The number of ways to select n donuts when k flavors are available is*

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

Practice Problems

Exercise 3.5.1 Eight students—Anna, Brian, Caine,...—are to be seated around a circular table in a circular room. Two seatings are regarded as defining the same *arrangement* if each student has the same student on his or her right in both seatings: it does not matter which way they face. We'll be interested in counting how many arrangements there are of these 8 students, given some restrictions.

(a) As a start, how many different arrangements of these 8 students around the table are there without any restrictions?

(b) How many arrangements of these 8 students are there with Anna sitting next to Brian?

(c) How many arrangements are there with if Brian sitting next to both Anna AND Caine?

(d) How many arrangements are there with Brian sitting next to Anna *OR* Caine?

Exercise 3.5.2 How many different ways are there to select an unordered bundle of three dozen colored roses if red, yellow, pink, white, purple and orange roses are available? Please explain your answer (you may leave it un-simplified).

Exercise 3.5.3 Suppose n books are lined up on a shelf. The number of selections of m of the books so that selected books are separated by at least three unselected books is the same as the number of *all* length k binary strings with exactly m ones.

(a) What is the value of k?

(b) Describe a bijection between between the set of all length k binary strings with exactly m ones and such book selections.

Exercise 3.5.4 Six women and nine men are on the faculty of a university's CS department. The individuals are distinguishable. How many ways are there to select a committee of 5 members if at least 1 woman must be on the committee?

Exercise 3.5.5 Your class tutorial has 12 students, who are supposed to break up into 4 groups of 3 students each. Your instructor has observed that the students waste too much time trying to form balanced groups, so he decided to pre-assign students to groups and email the group assignments to his students.

(a) Your instructor has a list of the 12 students in front of him, so he divides the list into consecutive groups of 3. For example, if the list is ABCDEFGHIJKL, the instructor would define a sequence of four groups to be $(\{A, B, C\}, \{D, E, F\}, \{G, H, I\}, \{J, K, L\})$. This way of forming groups defines a mapping from a list of twelve students to a sequence of four groups. This is a k-to-1 mapping for what k?

(b) A group assignment specifies which students are in the same group, but not any order in which the groups should be listed. If we map a sequence of 4 groups,

$$(\{A, B, C\}, \{D, E, F\}, \{G, H, I\}, \{J, K, L\}),$$

into a group assignment

$$\{\{A, B, C\}, \{D, E, F\}, \{G, H, I\}, \{J, K, L\}\},$$

this mapping is j-to-1 for what j?

(c) How many group assignments are possible?

(d) In how many ways can $3n$ students be broken up into n groups of 3?

Exercise 3.5.6 Answer the following quesions using the Generalized Product Rule.

(a) Next week, I'm going to get really fit! On day 1, I'll exercise for 5 minutes. On each subsequent day, I'll exercise 0, 1, 2, or 3 minutes more than the previous day. For example, the number of minutes that I exercise on the seven days of next week might be 5, 6, 9, 9, 9, 11, 12. How many such sequences are possible?

(b) An *r-permutation* of a set is a sequence of r distinct elements of that set. For example, here are all the 2-permutations of $\{a, b, c, d\}$:

$$\begin{array}{ccc}
(a, b) & (a, c) & (a, d) \\
(b, a) & (b, c) & (b, d) \\
(c, a) & (c, b) & (c, d) \\
(d, a) & (d, b) & (d, c)
\end{array}$$

How many r-permutations of an n-element set are there? Express your answer using factorial notation.

(c) How many $n \times n$ matrices are there with *distinct* entries drawn from $\{1, \ldots, p\}$, where $p \geq n^2$?

Exercise 3.5.7 Answer the followings.

(a) There are 30 books arranged in a row on a shelf. In how many ways can eight of these books be selected so that there are at least two unselected books between any two selected books?

(b) How many nonnegative integer solutions are there for the following equality?

$$x_1 + x_2 + \cdots + x_m = k. \tag{3.3}$$

(c) How many nonnegative integer solutions are there for the following inequality?

$$x_1 + x_2 + \cdots + x_m \leq k. \tag{3.4}$$

(d) How many length m weakly increasing sequences of nonnegative integers $\leq k$ are there?

Exercise 3.5.8 This problem is about binary relations on the set of integers in the interval $[1..n]$ and digraphs whose vertex set is $[1..n]$.

(a) How many digraphs are there?

(b) How many simple graphs are there?

(c) How many asymmetric binary relations are there?

(d) How many linear strict partial orders are there?

Exercise 3.5.9 Answer the following questions with a number or a simple formula involving factorials and binomial coefficients. Briefly explain your answers.

(a) How many ways are there to order the 26 letters of the alphabet so that no two of the vowels `a`, `e`, `i`, `o`, `u` appear consecutively and the last letter in the ordering is not a vowel?
Hint: Every vowel appears to the left of a consonant.

(b) How many ways are there to order the 26 letters of the alphabet so that there are *at least two* consonants immediately following each vowel?

(c) In how many different ways can $2n$ students be paired up?

(d) Two n-digit sequences of digits 0,1,...,9 are said to be of the *same type* if the digits of one are a permutation of the digits of the other. For $n = 8$, for example, the sequences `03088929` and `00238899` are the same type. How many types of n-digit sequences are there?

Exercise 3.5.10 Suppose that two identical 52-card decks are mixed together. Write a simple formula for the number of distinct permutations of the 104 cards.

3.6 Sequences with Repetitions

3.6.1 Sequences of Subsets

Choosing a k-element subset of an n-element set is the same as splitting the set into a pair of subsets: the first subset of size k and the second subset consisting of the remaining $n - k$ elements. So, the Subset Rule can be understood as a rule for counting the number of such splits into pairs of subsets.

We can generalize this to a way to count splits into more than two subsets. Let A be an n-element set and $k_1, k_2, \ldots, k_m$ be nonnegative integers whose sum is n. A $(k_1, k_2, \ldots, k_m)$*-split of* A is a sequence

$$(A_1, A_2, \ldots, A_m)$$

where the A_i are disjoint subsets of A and $|A_i| = k_i$ for $i = 1, \ldots, m$.

To count the number of splits we take the same approach as for the Subset Rule. Namely, we map any permutation $a_1 a_2 \ldots a_n$ of an n-element set A into a $(k_1, k_2, \ldots, k_m)$-split by letting the 1st subset in the split be the first k_1 elements of the permutation, the 2nd subset of the split be the next k_2 elements, ..., and the mth subset of the split be the final k_m elements of the permutation. This map is a $k_1!\ k_2!\ \cdots\ k_m!$-to-1 function from the $n!$ permutations to the $(k_1, k_2, \ldots, k_m)$-splits of A, so from the Division Rule we conclude the *Subset Split Rule*:

Definition 33. For $n, k_1, \ldots, k_m \in \mathbb{N}$, such that $k_1 + k_2 + \cdots + k_m = n$, define the *multinomial coefficient*

$$\binom{n}{k_1, k_2, \ldots, k_m} \triangleq \frac{n!}{k_1!\, k_2!\, \ldots k_m!}.$$

Rule 34 (Subset Split Rule). *The number of $(k_1, k_2, \ldots, k_m)$-splits of an n-element set is*

$$\binom{n}{k_1, \ldots, k_m}.$$

3.6.2 The Bookkeeper Rule

We can also generalize our count of n-bit sequences with k ones to counting sequences of n letters over an alphabet with more than two letters. For example, how many sequences can be formed by permuting the letters in the 10-letter word BOOKKEEPER?

Notice that there are 1 B, 2 O's, 2 K's, 3 E's, 1 P, and 1 R in BOOKKEEPER. This leads to a straightforward bijection between permutations of BOOKKEEPER and (1,2,2,3,1,1)-splits of $\{1, 2, \ldots, 10\}$. Namely, map a permutation to the sequence of sets of positions where each of the different letters occur.

For example, in the permutation BOOKKEEPER itself, the B is in the 1st position, the O's occur in the 2nd and 3rd positions, K's in 4th and 5th, the E's in the 6th, 7th and 9th, P in the 8th, and R is in the 10th position. So BOOKKEEPER maps to

$$(\{1\}, \{2, 3\}, \{4, 5\}, \{6, 7, 9\}, \{8\}, \{10\}).$$

From this bijection and the Subset Split Rule, we conclude that the number of ways to rearrange the letters in the word BOOKKEEPER is:

$$\frac{\overbrace{10!}^{\text{total letters}}}{\underbrace{1!}_{\text{B's}}\underbrace{2!}_{\text{O's}}\underbrace{2!}_{\text{K's}}\underbrace{3!}_{\text{E's}}\underbrace{1!}_{\text{P's}}\underbrace{1!}_{\text{R's}}}$$

This example generalizes directly to an exceptionally useful counting principle which we will call the

Rule 35 (Bookkeeper Rule). *Let $l_1, \ldots, l_m$ be distinct elements. The number of sequences with k_1 occurrences of l_1, and k_2 occurrences of l_2, ..., and k_m occurrences of l_m is*

$$\binom{k_1 + k_2 + \cdots + k_m}{k_1, \ldots, k_m}.$$

For example, suppose you are planning a 20-mile walk, which should include 5 northward miles, 5 eastward miles, 5 southward miles, and 5 westward miles. How many different walks are possible?

There is a bijection between such walks and sequences with 5 N's, 5 E's, 5 S's, and 5 W's. By the Bookkeeper Rule, the number of such sequences is:

$$\frac{20!}{(5!)^4}.$$

A Word about Words

Someday you might refer to the Subset Split Rule or the Bookkeeper Rule in front of a roomful of colleagues and discover that they're all staring back at you blankly. This is not because they're dumb, but rather because we made up the name "Bookkeeper Rule." However, the rule is excellent and the name is apt, so we suggest that you play through: "You know? The Bookkeeper Rule? Don't you guys know *anything?*"

The Bookkeeper Rule is sometimes called the "formula for permutations with indistinguishable objects." The size k subsets of an n-element set are sometimes called *k-combinations.* Other similar-sounding descriptions are "combinations with repetition, permutations with repetition, r-permutations, permutations with indistinguishable objects," and so on. However, the counting rules we've taught you are sufficient to solve all these sorts of problems without knowing this jargon, so we won't burden you with it.

Practice Problem

Exercise 3.6.1 The Tao of BOOKKEEPER: we seek enlightenment through contemplation of the word $BOOKKEEPER$.

(a) In how many ways can you arrange the letters in the word $POKE$?

(b) In how many ways can you arrange the letters in the word BO_1O_2K? Observe that we have subscripted the O's to make them distinct symbols.

(c) Suppose we map arrangements of the letters in BO_1O_2K to arrangements of the letters in $BOOK$ by erasing the subscripts. Indicate with arrows how the arrangements on the left are mapped to the arrangements on the right.

O_2BO_1K	
KO_2BO_1	$BOOK$
O_1BO_2K	$OBOK$
KO_1BO_2	$KOBO$
BO_1O_2K	...
BO_2O_1K	
...	

(d) This is a k-to-1 mapping, young grasshopper? What is k?

(e) In light of the Division Rule, how many arrangements are there of $BOOK$?

(f) Very good, young master! How many arrangements are there of the letters in $KE_1E_2PE_3R$?

(g) Suppose we map each arrangement of $KE_1E_2PE_3R$ to an arrangement of $KEEPER$ by erasing subscripts. List all the different arrangements of $KE_1E_2PE_3R$ that are mapped to $REPEEK$ in this way.

(h) What kind of mapping is this?

(i) So how many arrangements are there of the letters in $KEEPER$?

Now you are ready to face the BOOKKEEPER!

(j) How many arrangements of $BO_1O_2K_1K_2E_1E_2PE_3R$ are there?

(k) How many arrangements of $BOOK_1K_2E_1E_2PE_3R$ are there?

(l) How many arrangements of $BOOKKE_1E_2PE_3R$ are there?

(m) How many arrangements of $BOOKKEEPER$ are there?

Remember well what you have learned: subscripts on, subscripts off.
This is the Tao of Bookkeeper.

(n) How many arrangements of $VOODOODOLL$ are there?

(o) How many length 52 sequences of digits contain exactly 17 two's, 23 fives, and 12 nines?

3.6.3 The Binomial Theorem

Counting gives insight into one of the basic theorems of algebra. A *binomial* is a sum of two terms, such as $a + b$. Now consider its fourth power $(a + b)^4$.

By repeatedly using distributivity of products over sums to multiply out this 4th power expression completely, we get

$$\begin{array}{rcccccccc} (a+b)^4 & = & & aaaa & + & aaab & + & aaba & + & aabb \\ & & + & abaa & + & abab & + & abba & + & abbb \\ & & + & baaa & + & baab & + & baba & + & babb \\ & & + & bbaa & + & bbab & + & bbba & + & bbbb \end{array}$$

Notice that there is one term for every sequence of a's and b's. So there are 2^4 terms, and the number of terms with k copies of b and $n-k$ copies of a is:

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

by the Bookkeeper Rule. Hence, the coefficient of $a^{n-k}b^k$ is $\binom{n}{k}$. So for $n=4$, this means:

$$(a+b)^4 = \binom{4}{0}\cdot a^4b^0 + \binom{4}{1}\cdot a^3b^1 + \binom{4}{2}\cdot a^2b^2 + \binom{4}{3}\cdot a^1b^3 + \binom{4}{4}\cdot a^0b^4$$

In general, this reasoning gives the Binomial Theorem:

Theorem 36 (*Binomial Theorem*)**.** *For all $n \in \mathbb{N}$ and $a, b \in \mathbb{R}$:*

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

The Binomial Theorem explains why the n choose k number is called a*binomial coefficient*.

This reasoning about binomials extends nicely to *multinomials*, which are sums of two or more terms. For example, suppose we wanted the coefficient of

$$bo^2k^2e^3pr$$

in the expansion of $(b+o+k+e+p+r)^{10}$. Each term in this expansion is a product of 10 variables where each variable is one of b, o, k, e, p or r. Now, the coefficient of $bo^2k^2e^3pr$ is the number of those terms with exactly 1 b, 2 o's, 2 k's, 3 e's, 1 p and 1 r. And the number of such terms is precisely the number of rearrangements of the word BOOKKEEPER:

$$\binom{10}{1,2,2,3,1,1} = \frac{10!}{1!\;2!\;2!\;3!\;1!\;1!}.$$

This reasoning extends to a general theorem:

Theorem 37 (Multinomial Theorem)**.** *For all $n \in \mathbb{N}$,*

$$(z_1 + z_2 + \cdots + z_m)^n = \sum_{\substack{k_1,\ldots,k_m \in \mathbb{N} \\ k_1+\cdots+k_m = n}} \binom{n}{k_1, k_2, \ldots, k_m} z_1^{k_1} z_2^{k_2} \cdots z_m^{k_m}.$$

But you'll be better off remembering the reasoning behind the Multinomial Theorem rather than this cumbersome formal statement.

Practice Problems

Exercise 3.6.2 How many different permutations are there of the sequence of letters in "MISSISSIPPI"?

Exercise 3.6.3 Find the coefficients mentioned in the given expressions.

(a) Find the coefficients of $x^{10}y^5$ in $(19x + 4y)^{15}$

(b) Find the coefficient of x^4 in $(x + 1)^9$.

(c) Find the coefficient of x^4 in $(3x + 2)^6$.

Exercise 3.6.4 Find the coefficients of

(a) x^5 in $(1 + x)^{11}$

(b) x^8y^9 in $(3x + 2y)^{17}$

(c) a^6b^6 in $(a^2 + b^3)^5$

Exercise 3.6.5 Let G be a simple graph with 6 vertices and an edge between every pair of vertices (that is, G is a *complete* graph). A length-3 cycle in G is called a *triangle*.

A set of two edges that share a vertex is called an *incident pair* (i.p.); the shared vertex is called the *center* of the i.p. That is, an i.p. is a set, $\{\langle u—v\rangle, \langle v—w\rangle\}$, where u, v and w are distinct vertices, and its center is v.

(a) How many triangles are there?

(b) How many incident pairs are there?

Now suppose that every edge in G is colored either red or blue. A triangle or i.p. is called *multicolored* when its edges are not all the same color.

(c) Map the i.p. $\{\langle u—v\rangle, \langle v—w\rangle\}$ to the triangle $\{\langle u—v\rangle, \langle v—w\rangle, \langle u—w\rangle\}$.

Notice that multicolored i.p.'s map to multicolored triangles. Explain why this mapping is 2-to-1 on these multicolored objects.

(d) Show that at most six multicolored i.p.'s can have the same center. Conclude that there are at most 36 possible multicolored i.p.'s.

Hint: A vertex incident to r red edges and b blue edges is the center of $r \cdot b$ different multicolored i.p.'s.

(e) If two people are not friends, they are called *strangers*. If every pair of people in a group are friends, or if every pair are strangers, the group is called *uniform*.

Explain why parts (3.6.5), (3.6.5), and (3.6.5) imply that *every set of six people includes* two *uniform three-person groups*.

Exercise 3.6.6 There is a robot that steps between integer positions in 3-dimensional space. Each step of the robot increments one coordinate and leaves the other two unchanged.

(a) How many paths can the robot follow going from the origin $(0, 0, 0)$ to $(3, 4, 5)$?

(b) How many paths can the robot follow going from the origin (i, j, k) to (m, n, p)?

3.7 Counting Practice: Poker Hands

Five-Card Draw is a card game in which each player is initially dealt a *hand* consisting of 5 cards from a deck of 52 cards.[6] The number of different hands in Five-Card Draw is the number of 5-element

[6] There are 52 cards in a standard deck. Each card has a *suit* and a *rank*. There are four suits:

$$\spadesuit \text{ (spades)} \quad \heartsuit \text{ (hearts)} \quad \clubsuit \text{ (clubs)} \quad \diamondsuit \text{ (diamonds)}$$

And there are 13 ranks, listed here from lowest to highest:

$$\overset{\text{Ace}}{A}, 2, 3, 4, 5, 6, 7, 8, 9, \overset{\text{Jack}}{J}, \overset{\text{Queen}}{Q}, \overset{\text{King}}{K}.$$

subsets of a 52-element set, which is

$$\binom{52}{5} = 2,598,960.$$

Let's get some counting practice by working out the number of hands with various special properties.

3.7.1 Hands with a Four-of-a-Kind

A *Four-of-a-Kind* is a set of four cards with the same rank. How many different hands contain a Four-of-a-Kind? Here are a couple examples:

$$\{8\spadesuit,\ 8\diamondsuit,\ Q\heartsuit,\ 8\heartsuit,\ 8\clubsuit\}$$
$$\{A\clubsuit,\ 2\clubsuit,\ 2\heartsuit,\ 2\diamondsuit,\ 2\spadesuit\}$$

As usual, the first step is to map this question to a sequence-counting problem. A hand with a Four-of-a-Kind is completely described by a sequence specifying:

1. The rank of the four cards.
2. The rank of the extra card.
3. The suit of the extra card.

Thus, there is a bijection between hands with a Four-of-a-Kind and sequences consisting of two distinct ranks followed by a suit. For example, the three hands above are associated with the following sequences:

$$(8, Q, \heartsuit) \leftrightarrow \{\,8\spadesuit,\ 8\diamondsuit,\ 8\heartsuit,\ 8\clubsuit,\ Q\heartsuit\}$$
$$(2, A, \clubsuit) \leftrightarrow \{2\clubsuit,\ 2\heartsuit,\ 2\diamondsuit,\ 2\spadesuit,\ A\clubsuit\}$$

Now we need only count the sequences. There are 13 ways to choose the first rank, 12 ways to choose the second rank, and 4 ways to choose the suit. Thus, by the Generalized Product Rule, there are $13 \cdot 12 \cdot 4 = 624$ hands with a Four-of-a-Kind. This means that only 1 hand in about 4165 has a Four-of-a-Kind. Not surprisingly, Four-of-a-Kind is considered to be a very good poker hand!

3.7.2 Hands with a Full House

A *Full House* is a hand with three cards of one rank and two cards of another rank. Here are some examples:

$$\{2\spadesuit,\ 2\clubsuit,\ 2\diamondsuit,\ J\clubsuit,\ J\diamondsuit\}$$
$$\{5\diamondsuit,\ 5\clubsuit,\ 5\heartsuit,\ 7\heartsuit,\ 7\clubsuit\}$$

Again, we shift to a problem about sequences. There is a bijection between Full Houses and sequences specifying:

1. The rank of the triple, which can be chosen in 13 ways.
2. The suits of the triple, which can be selected in $\binom{4}{3}$ ways.
3. The rank of the pair, which can be chosen in 12 ways.
4. The suits of the pair, which can be selected in $\binom{4}{2}$ ways.

Thus, for example, $8\heartsuit$ is the 8 of hearts and $A\spadesuit$ is the ace of spades.

The example hands correspond to sequences as shown below:

$$(2, \{\spadesuit, \clubsuit, \diamondsuit\}, J, \{\clubsuit, \diamondsuit\}) \leftrightarrow \{2\spadesuit,\ 2\clubsuit,\ 2\diamondsuit,\ J\clubsuit,\ J\diamondsuit\}$$
$$(5, \{\diamondsuit, \clubsuit, \heartsuit\}, 7, \{\heartsuit, \clubsuit\}) \leftrightarrow \{5\diamondsuit,\ 5\clubsuit,\ 5\heartsuit,\ 7\heartsuit,\ 7\clubsuit\}$$

By the Generalized Product Rule, the number of Full Houses is:

$$13 \cdot \binom{4}{3} \cdot 12 \cdot \binom{4}{2}.$$

We're on a roll—but we're about to hit a speed bump.

3.7.3 Hands with Two Pairs

How many hands have *Two Pairs*; that is, two cards of one rank, two cards of another rank, and one card of a third rank? Here are examples:

$$\{3\diamondsuit,\ 3\spadesuit,\ Q\diamondsuit,\ Q\heartsuit,\ A\clubsuit\}$$
$$\{9\heartsuit,\ 9\diamondsuit,\ 5\heartsuit,\ 5\clubsuit,\ K\spadesuit\}$$

Each hand with Two Pairs is described by a sequence consisting of:

1. The rank of the first pair, which can be chosen in 13 ways.
2. The suits of the first pair, which can be selected $\binom{4}{2}$ ways.
3. The rank of the second pair, which can be chosen in 12 ways.
4. The suits of the second pair, which can be selected in $\binom{4}{2}$ ways.
5. The rank of the extra card, which can be chosen in 11 ways.
6. The suit of the extra card, which can be selected in $\binom{4}{1} = 4$ ways.

Thus, it might appear that the number of hands with Two Pairs is:

$$13 \cdot \binom{4}{2} \cdot 12 \cdot \binom{4}{2} \cdot 11 \cdot 4.$$

Wrong answer! The problem is that there is *not* a bijection from such sequences to hands with Two Pairs. This is actually a 2-to-1 mapping. For example, here are the pairs of sequences that map to the hands given above:

$$\begin{array}{ll}
(3, \{\diamondsuit, \spadesuit\}, Q, \{\diamondsuit, \heartsuit\}, A, \clubsuit) \searrow & \\
 & \{3\diamondsuit,\ 3\spadesuit,\ Q\diamondsuit,\ Q\heartsuit,\ A\clubsuit\} \\
(Q, \{\diamondsuit, \heartsuit\}, 3, \{\diamondsuit, \spadesuit\}, A, \clubsuit) \nearrow & \\
 & \\
(9, \{\heartsuit, \diamondsuit\}, 5, \{\heartsuit, \clubsuit\}, K, \spadesuit) \searrow & \\
 & \{9\heartsuit,\ 9\diamondsuit,\ 5\heartsuit,\ 5\clubsuit,\ K\spadesuit\} \\
(5, \{\heartsuit, \clubsuit\}, 9, \{\heartsuit, \diamondsuit\}, K, \spadesuit) \nearrow &
\end{array}$$

The problem is that nothing distinguishes the first pair from the second. A pair of 5's and a pair of 9's is the same as a pair of 9's and a pair of 5's. We avoided this difficulty in counting Full Houses because, for example, a pair of 6's and a triple of kings is different from a pair of kings and a triple of 6's.

We ran into precisely this difficulty last time, when we went from counting arrangements of *different* pieces on a chessboard to counting arrangements of two *identical* rooks. The solution then was to apply the Division Rule, and we can do the same here. In this case, the Division rule says there are twice as many sequences as hands, so the number of hands with Two Pairs is actually:

$$\frac{13 \cdot \binom{4}{2} \cdot 12 \cdot \binom{4}{2} \cdot 11 \cdot 4}{2}.$$

Another Approach

The preceding example was disturbing! One could easily overlook the fact that the mapping was 2-to-1 on an exam, fail the course, and turn to a life of crime. You can make the world a safer place in two ways:

1. Whenever you use a mapping $f : A \to B$ to translate one counting problem to another, check that the same number of elements in A are mapped to each element in B. If k elements of A map to each of element of B, then apply the Division Rule using the constant k.

2. As an extra check, try solving the same problem in a different way. Multiple approaches are often available—and all had better give the same answer! (Sometimes different approaches give answers that *look* different, but turn out to be the same after some algebra.)

We already used the first method; let's try the second. There is a bijection between hands with two pairs and sequences that specify:

1. The ranks of the two pairs, which can be chosen in $\binom{13}{2}$ ways.

2. The suits of the lower-rank pair, which can be selected in $\binom{4}{2}$ ways.

3. The suits of the higher-rank pair, which can be selected in $\binom{4}{2}$ ways.

4. The rank of the extra card, which can be chosen in 11 ways.

5. The suit of the extra card, which can be selected in $\binom{4}{1} = 4$ ways.

For example, the following sequences and hands correspond:

$$(\{3, Q\}, \{\diamondsuit, \spadesuit\}, \{\diamondsuit, \heartsuit\}, A, \clubsuit) \leftrightarrow \{3\diamondsuit,\ 3\spadesuit,\ Q\diamondsuit,\ Q\heartsuit,\ A\clubsuit\}$$
$$(\{9, 5\}, \{\heartsuit, \clubsuit\}, \{\heartsuit, \diamondsuit\}, K, \spadesuit) \leftrightarrow \{9\heartsuit,\ 9\diamondsuit,\ 5\heartsuit,\ 5\clubsuit,\ K\spadesuit\}$$

Thus, the number of hands with two pairs is:

$$\binom{13}{2} \cdot \binom{4}{2} \cdot \binom{4}{2} \cdot 11 \cdot 4.$$

This is the same answer we got before, though in a slightly different form.

3.7.4 Hands with Every Suit

How many hands contain at least one card from every suit? Here is an example of such a hand:

$$\{7\diamondsuit,\ K\clubsuit,\ 3\diamondsuit,\ A\heartsuit,\ 2\spadesuit\}$$

Each such hand is described by a sequence that specifies:

1. The ranks of the diamond, the club, the heart, and the spade, which can be selected in $13 \cdot 13 \cdot 13 \cdot 13 = 13^4$ ways.

2. The suit of the extra card, which can be selected in 4 ways.

3. The rank of the extra card, which can be selected in 12 ways.

For example, the hand above is described by the sequence:

$$(7, K, A, 2, \diamondsuit, 3) \leftrightarrow \{7\diamondsuit,\ K\clubsuit,\ A\heartsuit,\ 2\spadesuit,\ 3\diamondsuit\}.$$

Are there other sequences that correspond to the same hand? There is one more! We could equally well regard either the $3\diamondsuit$ or the $7\diamondsuit$ as the extra card, so this is actually a 2-to-1 mapping. Here are the two sequences corresponding to the example hand:

$$\begin{array}{ll} (7, K, A, 2, \diamondsuit, 3) & \searrow \\ & \qquad \{7\diamondsuit,\ K\clubsuit,\ A\heartsuit,\ 2\spadesuit,\ 3\diamondsuit\} \\ (3, K, A, 2, \diamondsuit, 7) & \nearrow \end{array}$$

Therefore, the number of hands with every suit is:

$$\frac{13^4 \cdot 4 \cdot 12}{2}.$$

Practice Problems

Exercise 3.7.1 Indicate how many 5-card hands there are of each of the following kinds.

(a) A **Sequence** is a hand consisting of five consecutive cards of any suit, such as

$$5\heartsuit - 6\heartsuit - 7\spadesuit - 8\blacklozenge - 9\clubsuit.$$

Note that an ace may either be high (as in 10-J-Q-K-A), or low (as in A-2-3-4-5), but can't go "around the corner" (that is, Q-K-A-2-3 is *not* a sequence).

How many different **Sequence** hands are possible?

(b) A **Matching Suit** is a hand consisting of cards that are all of the same suit in any order. How many different **Matching Suit** hands are possible?

(c) A **Straight Flush** is a hand that is both a *Sequence* and a *Matching Suit*. How many different **Straight Flush** hands are possible?

(d) A **Straight** is a hand that is a *Sequence* but not a *Matching Suit*. How many possible **Straights** are there?

(e) A **Flush** is a hand that is a *Matching Suit* but not a *Sequence*. How many possible **Flushes** are there?

3.8 The Pigeonhole Principle

Here is an old puzzle:

> A drawer in a dark room contains red socks, green socks, and blue socks. How many socks must you withdraw to be sure that you have a matching pair?

For example, picking out three socks is not enough; you might end up with one red, one green, and one blue. The solution relies on the

Pigeonhole Principle

> *If there are more pigeons than holes they occupy, then at least two pigeons must be in the same hole.*

A rigorous statement of the Principle goes this way:

Rule 38 (Pigeonhole Principle)**.** *If $|A| > |B|$, then for every total function $f : A \to B$, there exist two different elements of A that are mapped by f to the same element of B.*

Stating the Principle this way may be less intuitive, but it should now sound familiar: it is simply the contrapositive of the *Mapping Rules* injective case.Here, the pigeons form set A, the pigeonholes are the set B, and f describes which hole each pigeon occupies.

Mathematicians have come up with many ingenious applications for the pigeonhole principle. If there were a cookbook procedure for generating such arguments, we'd give it to you. Unfortunately, there isn't one. One helpful tip, though: when you try to solve a problem with the pigeonhole principle, the key is to clearly identify three things:

1. The set A (the pigeons).
2. The set B (the pigeonholes).
3. The function f (the rule for assigning pigeons to pigeonholes).

The Pope's Pigeonholes

The town of Orvieto in Umbria, Italy, offered a refuge for medieval popes who might be forced to flee from Rome. It lies on top of a high plateau whose steep cliffs protected against attackers. Over centuries the townspeople excavated around 1200 underground rooms where vast flocks of pigeons were kept as a self-renewing food source to enable the town to withstand long sieges. The figure on the right shows a typical cave wall in which dozens pigeonholes have been carved.

3.8.1 Hairs on Heads

There are a number of generalizations of the pigeonhole principle. For example:

Rule 39 (*Generalized Pigeonhole Principle*)**.** *If $|A| > k \cdot |B|$, then every total function $f : A \to B$ maps at least $k+1$ different elements of A to the same element of B.*

For example, if you pick two people at random, surely they are extremely unlikely to have *exactly* the same number of hairs on their heads. However, in the remarkable city of Boston, Massachusetts, there is a group of *three* people who have exactly the same number of hairs! Of course, there are many completely bald people in Boston, and they all have zero hairs. But we're talking about non-bald people; say a person is non-bald if they have at least ten thousand hairs on their head.

Boston has about 500,000 non-bald people, and the number of hairs on a person's head is at most 200,000.
Let A be the set of non-bald people in Boston.
Let $B = \{10,000, 10,001, \ldots, 200,000\}$, and let f map a person to the number of hairs on his or her head. Since $|A| > 2|B|$, the Generalized Pigeonhole Principle implies that at least three people have exactly the same number of hairs. We don't know who they are, but we know they exist!

3.8.2 Subsets with the Same Sum

For your reading pleasure, we have displayed ninety 25-digit numbers in Figure 3.3. Are there two different subsets of these 25-digit numbers that have the same sum? For example, maybe the sum of the last ten numbers in the first column is equal to the sum of the first eleven numbers in the second column?

Finding two subsets with the same sum may seem like a silly puzzle, but solving these sorts of problems turns out to be useful in diverse applications such as finding good ways to fit packages into shipping containers and decoding secret messages.

It turns out that it is hard to find different subsets with the same sum, which is why this problem arises in cryptography. But it is easy to prove that two such subsets *exist*. That's where the Pigeonhole Principle comes in.

Let A be the collection of all subsets of the 90 numbers in the list. Now the sum of any subset of numbers is at most $90 \cdot 10^{25}$, since there are only 90 numbers and every 25-digit number is less than 10^{25}. So let B be the the integer interval $[0..90 \cdot 10^{25}]$, and let f map each subset of numbers (in A) to its sum (in B).

We proved that an n-element set has 2^n different subsets in Section 3.2. Therefore:

$$|A| = 2^{90} \geq 1.237 \times 10^{27}$$

On the other hand:

$$|B| = 90 \cdot 10^{25} + 1 \leq 0.901 \times 10^{27}.$$

Both quantities are enormous, but $|A|$ is a bit greater than $|B|$. This means that f maps at least two elements of A to the same element of B. In other words, by the Pigeonhole Principle, two different subsets must have the same sum!

0020480135385502964448038	3171004832173501394113017
5763257331083479647409398	8247331000042995311646021
0489445991866915676240992	3208234421597368647019265
5800949123548989122628663	8496243997123475922766310
1082662032430379651370981	3437254656355157864869113
6042900801199280218026001	8518399140676002660747477
1178480894769706178994993	3574883393058653923711365
6116171789137737896701405	8543691283470191452333763
1253127351683239693851327	3644909946040480189969149
6144868973001582369723512	8675309258374137092461352
1301505129234077811069011	3790044132737084094417246
6247314593851169234746152	8694321112363996867296665
1311567111143866433882194	3870332127437971355322815
6814428944266874963488274	8772321203608477245851154
1470029452721203587686214	4080505804577801451363100
6870852945543886849147881	8791422161722582546341091
1578271047286257499433886	4167283461025702348124920
6914955508120950093732397	9062628024592126283973285
1638243921852176243192354	4235996831123777788211249
6949632451365987152423541	9137845566925526349897794
1763580219131985963102365	4670939445749439042111220
7128211143613619828415650	9153762966803189291934419
1826227795601842231029694	4815379351865384279613427
7173920083651862307925394	9270880194077636406984249
1843971862675102037201420	4837052948212922604442190
7215654874211755676220587	9324301480722103490379204
2396951193722134526177237	5106389423855018550671530
7256932847164391040233050	9436090832146695147140581
2781394568268599801096354	5142368192004769218069910
7332822657075235431620317	9475308159734538249013238
2796605196713610405408019	5181234096130144084041856
7426441829541573444964139	9492376623917486974923202
2931016394761975263190347	5198267398125617994391348
7632198126531809327186321	9511972558779880288252979
2933458058294405155197296	5317592940316231219758372
7712154432211912882310511	9602413424619187112552264
3075514410490975920315348	5384358126771794128356947
7858918664240262356610010	9631217114906129219461111
8149436716871371161932035	3157693105325111284321993
3111474985252793452860017	5439211712248901995423441
7898156786763212963178679	9908189853102753335981319
3145621587936120118438701	5610379826092838192760458
8147591017037573337848616	9913237476341764299813987
3148901255628881103198549	5632317555465228677676044
5692168374637019617423712	8176063831682536571306791

Figure 3.3: Ninety 25-digit numbers. Can you find two different subsets of these numbers that have the same sum?

Notice that this proof gives no indication *which* two sets of numbers have the same sum. This frustrating variety of argument is called a *nonconstructive proof*.

The \$500 Prize for Sets with Distinct Subset Sums

How can we construct a set of n positive integers such that all its subsets have *distinct* sums? One way is to use powers of two:

$$\{1, 2, 4, 8, 16\}$$

This approach is so natural that one suspects all other such sets must involve larger numbers. (For example, we could safely replace 16 by 17, but not by 15.) Remarkably, there are examples involving *smaller* numbers. Here is one:

$$\{6, 9, 11, 12, 13\}$$

One of the top mathematicians of the Twentieth Century, Paul Erdős, conjectured in 1931 that there are no such sets involving *significantly* smaller numbers. More precisely, he conjectured that the largest number in such a set must be greater than $c2^n$ for some constant $c > 0$. He offered \$500 to anyone who could prove or disprove his conjecture, but the problem remains unsolved.

3.8.3 A Magic Trick

A Magician sends an Assistant into the audience with a deck of 52 cards while the Magician looks away.

Five audience members each select one card from the deck. The Assistant then gathers up the five cards and holds up four of them so the Magician can see them. The Magician concentrates for a short time and then correctly names the secret, fifth card!

Since we don't really believe the Magician can read minds, we know the Assistant has somehow communicated the secret card to the Magician. Real Magicians and Assistants are not to be trusted, so we expect that the Assistant would secretly signal the Magician with coded phrases or body language, but for this trick they don't have to cheat. In fact, the Magician and Assistant could be kept out of sight of each other while some audience member holds up the 4 cards designated by the Assistant for the Magician to see.

Of course, without cheating, there is still an obvious way the Assistant can communicate to the Magician: he can choose any of the $4! = 24$ permutations of the 4 cards as the order in which to hold up the cards. However, this alone won't quite work: there are 48 cards remaining in the deck, so the Assistant doesn't have enough choices of orders to indicate exactly what the secret card is (though he could narrow it down to two cards).

3.8.4 The Secret

The method the Assistant can use to communicate the fifth card exactly is a nice application of what we know about counting and matching.

The Assistant has a second legitimate way to communicate: he can choose *which of the five cards to keep hidden*. Of course, it's not clear how the Magician could determine which of these five possibilities the Assistant selected by looking at the four visible cards, but there is a way, as we'll now explain.

The problem facing the Magician and Assistant is actually a bipartite matching problem. Each vertex on the left will correspond to the information available to the Assistant, namely, a *set* of 5 cards. So the set X of left-hand vertices will have $\binom{52}{5}$ elements.

Each vertex on the right will correspond to the information available to the Magician, namely, a *sequence* of 4 distinct cards. So the set Y of right-hand vertices will have $52 \cdot 51 \cdot 50 \cdot 49$ elements. When the audience selects a set of 5 cards, then the Assistant must reveal a sequence of 4 cards from that hand. This constraint is represented by having an edge between a set of 5 cards on the left and

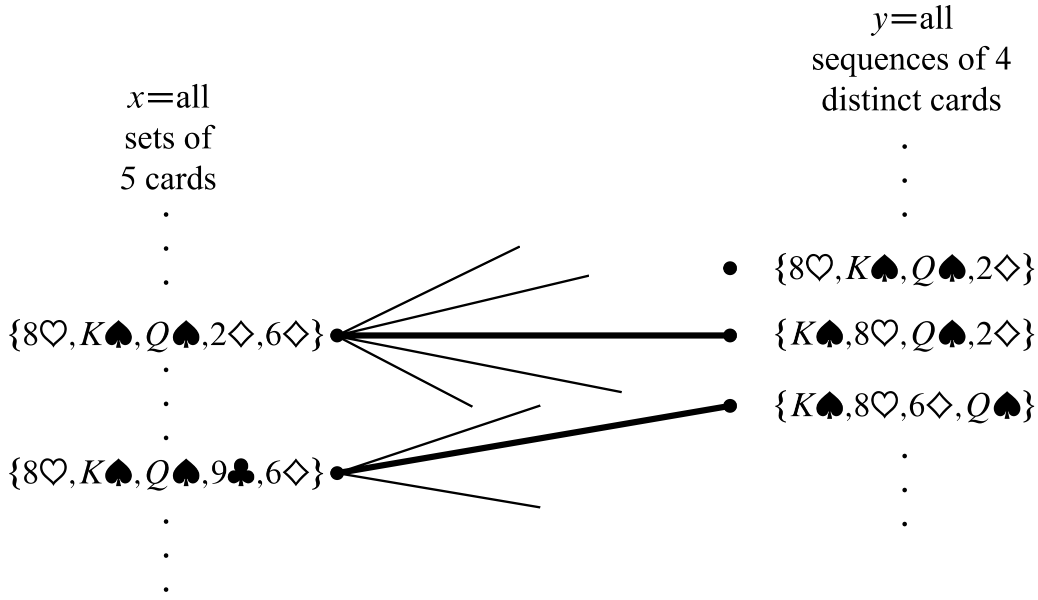


Figure 3.4: The bipartite graph where the nodes on the left correspond to *sets* of 5 cards and the nodes on the right correspond to *sequences* of 4 cards. There is an edge between a set and a sequence whenever all the cards in the sequence are contained in the set.

a sequence of 4 cards on the right precisely when every card in the sequence is also in the set. This specifies the bipartite graph. Some edges are shown in the diagram in Figure 3.4.

For example,

$$\{8\heartsuit, K\spadesuit, Q\spadesuit, 2\diamondsuit, 6\diamondsuit\} \tag{3.5}$$

is an element of X on the left. If the audience selects this set of 5 cards, then there are many different 4-card sequences on the right in set Y that the Assistant could choose to reveal, including $(8\heartsuit, K\spadesuit, Q\spadesuit, 2\diamondsuit)$, $(K\spadesuit, 8\heartsuit, Q\spadesuit, 2\diamondsuit)$ and $(K\spadesuit, 8\heartsuit, 6\diamondsuit, Q\spadesuit)$.

What the Magician and his Assistant need to perform the trick is a *matching* for the X vertices. If they agree in advance on some matching, then when the audience selects a set of 5 cards, the Assistant reveals the matching sequence of 4 cards. The Magician uses the matching to find the audience's chosen set of 5 cards, and so he can name the one not already revealed.

For example, suppose the Assistant and Magician agree on a matching containing the two bold edges in Figure 3.4. If the audience selects the set

$$\{8\heartsuit, K\spadesuit, Q\spadesuit, 9\clubsuit, 6\diamondsuit\}, \tag{3.6}$$

then the Assistant reveals the corresponding sequence

$$(K\spadesuit, 8\heartsuit, 6\diamondsuit, Q\spadesuit). \tag{3.7}$$

Using the matching, the Magician sees that the hand (3.6) is matched to the sequence (3.7), so he can name the one card in the corresponding set not already revealed, namely, the $9\clubsuit$. Notice that the fact that the sets are *matched*, that is, that different sets are paired with *distinct* sequences, is essential. For example, if the audience picked the previous hand (3.5), it would be possible for the Assistant to reveal the same sequence (3.7), but he better not do that; if he did, then the Magician would have no way to tell if the remaining card was the $9\clubsuit$ or the $2\diamondsuit$.

So how can we be sure the needed matching can be found? The answer is that each vertex on the left has degree $5 \cdot 4! = 120$, since there are five ways to select the card kept secret and there are $4!$ permutations of the remaining 4 cards. In addition, each vertex on the right has degree 48, since there are 48 possibilities for the fifth card. Therefore this graph is *degree-constrained*, and so has a

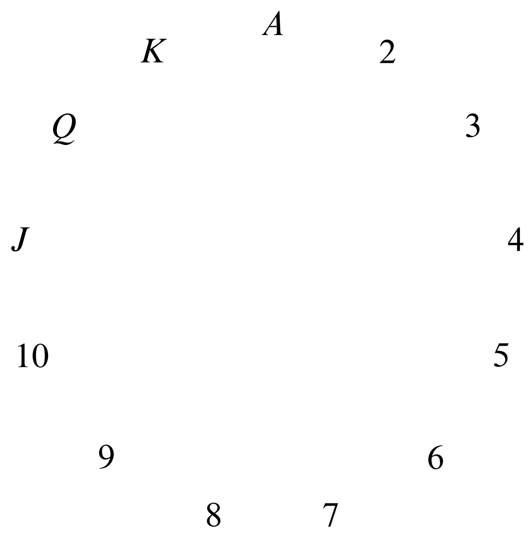


Figure 3.5: The 13 card ranks arranged in cyclic order.

matching[7].

In fact, this reasoning shows that the Magician could still pull off the trick if 120 cards were left instead of 48, that is, the trick would work with a deck as large as 124 different cards—without any magic!

3.8.5 The Real Secret

But wait a minute! It's all very well in principle to have the Magician and his Assistant agree on a matching, but how are they supposed to remember a matching with $\binom{52}{5} = 2,598,960$ edges? For the trick to work in practice, there has to be a way to match hands and card sequences mentally and on the fly.

We'll describe one approach. As a running example, suppose that the audience selects:

$$10\heartsuit \quad 9\diamondsuit \quad 3\heartsuit \quad Q\spadesuit \quad J\diamondsuit.$$

- The Assistant picks out two cards of the same suit. In the example, the assistant might choose the $3\heartsuit$ and $10\heartsuit$. This is always possible because of the Pigeonhole Principle—there are five cards and 4 suits so two cards must be in the same suit.

- The Assistant locates the ranks of these two cards on the cycle shown in Figure 3.5. For any two distinct ranks on this cycle, one is always between 1 and 6 hops clockwise from the other. For example, the $3\heartsuit$ is 6 hops clockwise from the $10\heartsuit$.

- The more counterclockwise of these two cards is revealed first, and the other becomes the secret card. Thus, in our example, the $10\heartsuit$ would be revealed, and the $3\heartsuit$ would be the secret card. Therefore:

  - The suit of the secret card is the same as the suit of the first card revealed.
  - The rank of the secret card is between 1 and 6 hops clockwise from the rank of the first card revealed.

- All that remains is to communicate a number between 1 and 6. The Magician and Assistant agree beforehand on an ordering of all the cards in the deck from smallest to largest such as:

$$A\clubsuit \; A\diamondsuit \; A\heartsuit \; A\spadesuit \; 2\clubsuit \; 2\diamondsuit \; 2\heartsuit \; 2\spadesuit \; \ldots \; K\heartsuit \; K\spadesuit$$

[7] A bipartite graph G is degree-constrained when there is an integer $d \geq 1$ such that $\deg(l) \geq d \geq \deg(r)$ for every $l \in L(G)$ and $r \in R(G)$ (the left and the right partitions of the vertices, respectively). It can be proven that *if G is a degree-constrained bipartite graph, then there is a matching that covers $L(G)$.*

The order in which the last three cards are revealed communicates the number according to the following scheme:

$$\begin{array}{rl}
(\text{ small, medium, large }) & = 1 \\
(\text{ small, large, medium }) & = 2 \\
(\text{ medium, small, large }) & = 3 \\
(\text{ medium, large, small }) & = 4 \\
(\text{ large, small, medium }) & = 5 \\
(\text{ large, medium, small }) & = 6
\end{array}$$

In the example, the Assistant wants to send 6 and so reveals the remaining three cards in large, medium, small order. Here is the complete sequence that the Magician sees:

$$10\heartsuit \quad Q\spadesuit \quad J\diamondsuit \quad 9\diamondsuit$$

- The Magician starts with the first card $10\heartsuit$ and hops 6 ranks clockwise to reach $3\heartsuit$, which is the secret card!

So that's how the trick can work with a standard deck of 52 cards. On the other hand, Hall's Theorem implies that the Magician and Assistant can *in principle* perform the trick with a deck of up to 124 cards. It turns out that there is a method which they could actually learn to use with a reasonable amount of practice for a 124-card deck, but we won't explain it here.[8]

3.8.6 The Same Trick with Four Cards?

Suppose that the audience selects only *four* cards and the Assistant reveals a sequence of *three* to the Magician. Can the Magician determine the fourth card?

Let X be all the sets of four cards that the audience might select, and let Y be all the sequences of three cards that the Assistant might reveal. Now, on one hand, we have

$$|X| = \binom{52}{4} = 270,725$$

by the Subset Rule. On the other hand, we have

$$|Y| = 52 \cdot 51 \cdot 50 = 132,600$$

by the Generalized Product Rule. Thus, by the Pigeonhole Principle, the Assistant must reveal the *same* sequence of three cards for at least

$$\left\lceil \frac{270,725}{132,600} \right\rceil = 3$$

different four-card hands. This is bad news for the Magician: if he sees that sequence of three, then there are at least three possibilities for the fourth card which he cannot distinguish. So there is no legitimate way for the Assistant to communicate exactly what the fourth card is!

Practice Problems

Exercise 3.8.1 Below is a list of properties that a group of people might possess.

For each property, either give the minimum number of people that must be in a group to ensure that the property holds, or else indicate that the property need not hold even for arbitrarily large groups of people.

(Assume that every year has exactly 365 days; ignore leap years.)

[8] See *The Best Card Trick* by Michael Kleber for more information.

(a) At least 2 people were born on the same day of the year (ignore year of birth).

(b) At least 2 people were born on January 1.

(c) At least 3 people were born on the same day of the week.

(d) At least 4 people were born in the same month.

(e) At least 2 people were born exactly one week apart.

Exercise 3.8.2 Solve the following problems using the pigeonhole principle. For each problem, try to identify the *pigeons*, the *pigeonholes*, and a *rule* assigning each pigeon to a pigeonhole.

(a) In a certain Institute of Technology, every ID number starts with a 9. Suppose that each of the 75 students in a class sums the nine digits of their ID number. Explain why two people must arrive at the same sum.

(b) In every set of 100 integers, there exist two whose difference is a multiple of 37.

(c) For any five points inside a unit square (not on the boundary), there are two points at distance *less than* $1/\sqrt{2}$.

(d) Show that if $n+1$ numbers are selected from $\{1, 2, 3, \ldots, 2n\}$, two must be consecutive, that is, equal to k and $k+1$ for some k.

Exercise 3.8.3 The aim of this problem is to prove that there exist a natural number n such that 3^n has at least 2013 consecutive zeros in its decimal expansion.

(a) Prove that there exist a nonnegative integer n such that $3^n \equiv 1 \pmod{10^{2014}}$.

Hint: Use pigeonhole principle or Euler's theorem.

(b) Conclude that there exist a natural number n such that 3^n has at least 2013 consecutive zeros.

Exercise 3.8.4 Suppose $2n+1$ numbers are selected from $\{1, 2, 3, \ldots, 4n\}$. Using the Pigeonhole Principle, show that there must be two selected numbers whose difference is 2. Clearly indicate what are the pigeons, holes, and rules for assigning a pigeon to a hole.

Exercise 3.8.5

(a) Show that any odd integer x in the range $10^9 < x < 2 \cdot 10^9$ containing all ten digits $0, 1, \ldots, 9$ must have consecutive even digits. *Hint:* What can you conclude about the parities of the first and last digit?

(b) Show that there are 2 vertices of equal degree in any finite undirected graph with $n \geq 2$ vertices. *Hint:* Cases conditioned upon the existence of a degree zero vertex.

Exercise 3.8.6 Suppose $n+1$ numbers are selected from $\{1, 2, 3, \ldots, 2n\}$. Using the Pigeonhole Principle, show that there must be two selected numbers whose quotient is a power of two. Clearly indicate what are the pigeons, holes, and rules for assigning a pigeon to a hole.

Hint: Factor each number into the product of an odd number and a power of 2.

Exercise 3.8.7

(a) Let R be an 82×4 rectangular matrix each of whose entries are colored red, white or blue. Explain why at least two of the 82 rows in R must have identical color patterns.

(b) Conclude that R contains four points with the same color that form the corners of a rectangle.

(c) Now show that the conclusion from part (3.8.7) holds even when R has only 19 rows.

Hint: How many ways are there to pick two positions in a row of length four and color them the same?

Exercise 3.8.8 Use the Pigeonhole Principle to determine the smallest nonnegative integer n such that every set of n integers is guaranteed to contain three integers that are congruent mod 211. Clearly indicate what are the pigeons, holes, and rules for assigning a pigeon to a hole, and give the value of n.

3.9 Inclusion-Exclusion

How big is a union of sets? For example, suppose there are 60 math majors, 200 CSE majors, and 40 physics majors. How many students are there in these three departments? Let M be the set of math majors, E be the set of CSE majors, and P be the set of physics majors. In these terms, we're asking for $|M \cup E \cup P|$.

The Sum Rule says that if M, E and P are disjoint, then the sum of their sizes is

$$|M \cup E \cup P| = |M| + |E| + |P|.$$

However, the sets M, E and P might *not* be disjoint. For example, there might be a student majoring in both math and physics. Such a student would be counted twice on the right side of this equation, once as an element of M and once as an element of P. Worse, there might be a triple-major counted *three* times on the right side!

Our most-complicated counting rule determines the size of a union of sets that are not necessarily disjoint. Before we state the rule, let's build some intuition by considering some easier special cases: unions of just two or three sets.

3.9.1 Union of Two Sets

For two sets, S_1 and S_2, the *Inclusion-Exclusion Rule* is that the size of their union is:

$$|S_1 \cup S_2| = |S_1| + |S_2| - |S_1 \cap S_2| \tag{3.8}$$

Intuitively, each element of S_1 is accounted for in the first term, and each element of S_2 is accounted for in the second term. Elements in *both* S_1 and S_2 are counted *twice*—once in the first term and once in the second. This double-counting is corrected by the final term.

3.9.2 Union of Three Sets

So how many students are there in the math, CSE, and physics departments? In other words, what is $|M \cup E \cup P|$ if:

$$\begin{aligned} |M| &= 60 \\ |E| &= 200 \\ |P| &= 40. \end{aligned}$$

The size of a union of three sets is given by a more complicated *Inclusion-Exclusion* formula:

$$\begin{aligned} |S_1 \cup S_2 \cup S_3| = {} & |S_1| + |S_2| + |S_3| \\ & - |S_1 \cap S_2| - |S_1 \cap S_3| - |S_2 \cap S_3| \\ & + |S_1 \cap S_2 \cap S_3|. \end{aligned}$$

Remarkably, the expression on the right accounts for each element in the union of S_1, S_2 and S_3 exactly once. For example, suppose that x is an element of all three sets. Then x is counted three times (by the $|S_1|$, $|S_2|$ and $|S_3|$ terms), subtracted off three times (by the $|S_1 \cap S_2|$, $|S_1 \cap S_3|$ and $|S_2 \cap S_3|$ terms), and then counted once more (by the $|S_1 \cap S_2 \cap S_3|$ term). The net effect is that x is counted just once.

If x is in two sets (say, S_1 and S_2), then x is counted twice (by the $|S_1|$ and $|S_2|$ terms) and subtracted once (by the $|S_1 \cap S_2|$ term). In this case, x does not contribute to any of the other terms, since $x \notin S_3$.

So we can't answer the original question without knowing the sizes of the various intersections. Let's suppose that there are:

4	math - CSE double majors
3	math - physics double majors
11	CSE - physics double majors
2	triple majors

Then $|M \cap E| = 4 + 2$, $|M \cap P| = 3 + 2$, $|E \cap P| = 11 + 2$, and $|M \cap E \cap P| = 2$. Plugging all this into the formula gives:

$$\begin{aligned}
|M \cup E \cup P| &= |M| + |E| + |P| - |M \cap E| - |M \cap P| - |E \cap P| \\
&\quad + |M \cap E \cap P| \\
&= 60 + 200 + 40 - 6 - 5 - 13 + 2 \\
&= 278
\end{aligned}$$

3.9.3 Sequences with 42, 04, or 60

In how many permutations of the set $\{0, 1, 2, \ldots, 9\}$ do either 4 and 2, 0 and 4, or 6 and 0 appear consecutively? For example, none of these pairs appears in:

$$(7, 2, 9, 5, 4, 1, 3, 8, 0, 6).$$

The 06 at the end doesn't count; we need 60. On the other hand, both 04 and 60 appear consecutively in this permutation:

$$(7, 2, 5, \underline{6}, \underline{0}, \underline{4}, 3, 8, 1, 9).$$

Let P_{42} be the set of all permutations in which 42 appears. Define P_{60} and P_{04} similarly. Thus, for example, the permutation above is contained in both P_{60} and P_{04}, but not P_{42}. In these terms, we're looking for the size of the set $P_{42} \cup P_{04} \cup P_{60}$.

First, we must determine the sizes of the individual sets, such as P_{60}. We can use a trick: group the 6 and 0 together as a single symbol. Then there is an immediate bijection between permutations of $\{0, 1, 2, \ldots 9\}$ containing 6 and 0 consecutively and permutations of:

$$\{60, 1, 2, 3, 4, 5, 7, 8, 9\}.$$

For example, the following two sequences correspond:

$$(7, 2, 5, \underline{6}, \underline{0}, 4, 3, 8, 1, 9) \longleftrightarrow (7, 2, 5, \underline{60}, 4, 3, 8, 1, 9).$$

There are 9! permutations of the set containing 60, so $|P_{60}| = 9!$ by the Bijection Rule. Similarly, $|P_{04}| = |P_{42}| = 9!$ as well.

Next, we must determine the sizes of the two-way intersections, such as $P_{42} \cap P_{60}$. Using the grouping trick again, there is a bijection with permutations of the set:

$$\{42, 60, 1, 3, 5, 7, 8, 9\}.$$

Thus, $|P_{42} \cap P_{60}| = 8!$. Similarly, $|P_{60} \cap P_{04}| = 8!$ by a bijection with the set:

$$\{604, 1, 2, 3, 5, 7, 8, 9\}.$$

And $|P_{42} \cap P_{04}| = 8!$ as well by a similar argument. Finally, note that $|P_{60} \cap P_{04} \cap P_{42}| = 7!$ by a bijection with the set:

$$\{6042, 1, 3, 5, 7, 8, 9\}.$$

Plugging all this into the formula gives:

$$|P_{42} \cup P_{04} \cup P_{60}| = 9! + 9! + 9! - 8! - 8! - 8! + 7!.$$

3.9.4 Union of n Sets

The size of a union of n sets is given by the following rule.

Rule 40 (*Inclusion-Exclusion*).

$$|S_1 \cup S_2 \cup \cdots \cup S_n| =$$

	the sum of the sizes of the individual sets
minus	*the sizes of all two-way intersections*
plus	*the sizes of all three-way intersections*
minus	*the sizes of all four-way intersections*
plus	*the sizes of all five-way intersections, etc.*

The formulas for unions of two and three sets are special cases of this general rule.

This way of expressing Inclusion-Exclusion is easy to understand and nearly as precise as expressing it in mathematical symbols, but we'll need the symbolic version below, so let's work on deciphering it now.

We already have a concise notation for the sum of sizes of the individual sets, namely,

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

A "two-way intersection" is a set of the form $S_i \cap S_j$ for $i \neq j$. We regard $S_j \cap S_i$ as the same two-way intersection as $S_i \cap S_j$, so we can assume that $i < j$. Now we can express the sum of the sizes of the two-way intersections as

$$\sum_{1 \le i < j \le n} |S_i \cap S_j|.$$

Similarly, the sum of the sizes of the three-way intersections is

$$\sum_{1 \le i < j < k \le n} |S_i \cap S_j \cap S_k|.$$

These sums have alternating signs in the Inclusion-Exclusion formula, with the sum of the k-way intersections getting the sign $(-1)^{k-1}$. This finally leads to a symbolic version of the rule:

Rule (Inclusion-Exclusion).

$$\begin{aligned} \left| \bigcup_{i=1}^{n} S_i \right| &= \sum_{i=1}^{n} |S_i| - \sum_{1 \le i < j \le n} |S_i \cap S_j| \\ &+ \sum_{1 \le i < j < k \le n} |S_i \cap S_j \cap S_k| + \cdots \\ &+ (-1)^{n-1} \left| \bigcap_{i=1}^{n} S_i \right|. \end{aligned}$$

While it's often handy express the rule in this way as a sum of sums, it is not necessary to group the terms by how many sets are in the intersections. So another way to state the rule is:

Rule (Inclusion-Exclusion-II).

$$\left| \bigcup_{i=1}^{n} S_i \right| = \sum_{\emptyset \ne I \subseteq \{1,\ldots,n\}} (-1)^{|I|+1} \left| \bigcap_{i \in I} S_i \right| \tag{3.9}$$

A proof of these rules using just highschool algebra is given in Problem 3.9.4.

Practice Problems

Exercise 3.9.1 Let A_1, A_2, A_3 be sets with $|A_1| = 100$, $|A_2| = 1,000$, and $|A_3| = 10,000$.

Determine $|A_1 \cup A_2 \cup A_3|$ in each of the following cases, or give an example showing that the value cannot be determined.

(a) $A_1 \subset A_2 \subset A_3$.

(b) The sets are pairwise disjoint.

(c) For any two of the sets, there is exactly one element in both.

(d) There are two elements common to each pair of sets and one element in all three sets.

Exercise 3.9.2 The working days in the next year can be numbered 1, 2, 3, ..., 300. I'd like to avoid as many as possible.

- On even-numbered days, I'll say I'm sick.
- On days that are a multiple of 3, I'll say I was stuck in traffic.
- On days that are a multiple of 5, I'll refuse to come out from under the blankets.

In total, how many work days will I *avoid* in the coming year?

Exercise 3.9.3 To ensure password security, a company requires their employees to choose a password. A length 10 word containing each of the characters: `a, d, e, f, i, l, o, p, r, s`, is called a *cword*. A password can be a cword which does not contain any of the subwords "fails", "failed", or "drop."

For example, the following two cwords are passwords: `adefiloprs`, `srpolifeda`, but the following three cwords are not: a**drop**eflis, **failedrop**s, **drop**e**fails**.

(a) How many cwords contain the subword "drop"?

(b) How many cwords contain both "drop" and "fails"?

(c) Use the Inclusion-Exclusion Principle to find a simple arithmetic formula involving factorials for the number of passwords.

Exercise 3.9.4 Let's develop a proof of the Inclusion-Exclusion formula using high school algebra.

(a) Most high school students will get freaked by the following formula, even though they actually know the rule it expresses. How would you explain it to them?

$$\prod_{i=1}^{n}(1 - x_i) = \sum_{I \subseteq \{1,\ldots,n\}} (-1)^{|I|} \prod_{j \in I} x_j. \tag{3.10}$$

Hint: Show them an example.

Now to start proving (3.10), let M_S be the *membership* function for any set S:

$$M_S(x) = \begin{cases} 1 & \text{if } x \in S, \\ 0 & \text{if } x \notin S. \end{cases}$$

Let $S_1, \ldots, S_n$ be a sequence of finite sets, and abbreviate M_{S_i} as M_i. Let the *domain of discourse* D be the union of the S_i's. That is, we let $D \triangleq \bigcup_{i=1}^{n} S_i$, and take complements with respect to D, that is, $\overline{T} \triangleq D - T$, for $T \subseteq D$.

(b) Verify that for $T \subseteq D$ and $I \subseteq [1..n]$

$$M_{\overline{T}} = 1 - M_T, \tag{3.11}$$

$$M_{(\bigcap_{i \in I} S_i)} = \prod_{i \in I} M_i, \tag{3.12}$$

$$M_{(\bigcup_{i \in I} S_i)} = 1 - \prod_{i \in I} (1 - M_i). \tag{3.13}$$

(Note that (3.12) holds when I is empty because, by convention, an empty product equals 1, and an empty intersection equals the domain of discourse D.)

(c) Use (3.10) and (3.13) to prove

$$M_D = \sum_{\emptyset \neq I \subseteq \{1,\ldots,n\}} (-1)^{|I|+1} \prod_{j \in I} M_j. \tag{3.14}$$

(d) Prove that

$$|T| = \sum_{u \in D} M_T(u). \tag{3.15}$$

(e) Now use the previous parts to prove

$$|D| = \sum_{\emptyset \neq I \subseteq \{1,\ldots,n\}} (-1)^{|I|+1} \left| \bigcap_{i \in I} S_i \right| \tag{3.16}$$

(f) Finally, explain why (3.16) immediately implies the usual form of the Inclusion-Exclusion Principle:

$$|D| = \sum_{i=1}^{n} (-1)^{i+1} \sum_{\substack{I \subseteq [1..n] \\ |I| = i}} \left| \bigcap_{j \in I} S_j \right|. \tag{3.17}$$

Exercise 3.9.5 We want to count step-by-step paths between points with integer coordinates in three dimensions. A step may move a unit distance in the positive x, y or z direction. For example, a step from point $(2, 3, 7)$ in the y direction leads to $(2, 4, 7)$.

For points $\mathbf{p}$ and $\mathbf{q}$ we write $\mathbf{p} \leq \mathbf{q}$ to mean that $\mathbf{p}$ is coordinatewise less than or equal to $\mathbf{q}$. That is, if $\mathbf{p} = (p_x, p_y, p_z)$ and $\mathbf{q} = (q_x, q_y, q_z)$, then

$$\mathbf{p} \leq \mathbf{q} \quad \triangleq \quad [p_x \leq q_x \text{ AND } p_y \leq q_y \text{ AND } p_z \leq q_z].$$

So there is a path from $\mathbf{p}$ to $\mathbf{q}$ iff $\mathbf{p} \leq \mathbf{q}$.

(a) Let $P_{\{\mathbf{p},\mathbf{q}\}}$ be the set of paths from $\mathbf{p}$ to $\mathbf{q}$. Suppose that $\mathbf{p} \leq \mathbf{q}$, and let $d_x \triangleq q_x - p_x$, and likewise for d_y and d_z. Express the number of paths $|P_{\{\mathbf{p},\mathbf{q}\}}|$ as a multinomial coefficient involving the preceding quantities.

More generally, for any set S of points, let

$$P_S \triangleq \text{the paths that go through all the points in } S.$$

(b) Suppose $\mathbf{a} \leq \mathbf{b} \leq \mathbf{c} \leq \mathbf{d}$. Express $|P_{\{\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}\}}|$ in terms of $|P_{\{\mathbf{p},\mathbf{q}\}}|$ for various $\mathbf{p}, \mathbf{q} \in \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$.

(c) Let

$$\begin{aligned}
&\mathbf{o} \triangleq (0, 0, 0), \\
&\mathbf{a} \triangleq (3, 7, 11), \quad \mathbf{b} \triangleq (11, 6, 3), \quad \mathbf{c} \triangleq (10, 5, 40), \\
&\mathbf{d} \triangleq (12, 13, 14), \quad \mathbf{e} \triangleq (12, 6, 45), \\
&\mathbf{f} \triangleq (50, 50, 50).
\end{aligned}$$

Let N be the paths in $P_{\mathbf{o},\mathbf{f}}$ that do *not* go through any of $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}$. Express $|N|$ as an arithmetic combination of $|P_S|$ for various $S \subseteq \{\mathbf{o}, \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{f}\}$. Do not include any terms $|P_S|$

that equal zero.

Exercise 3.9.6 How many paths are there from point $(0,0)$ to $(50,50)$ if each step along a path increments one coordinate and leaves the other unchanged? How many are there when there are impassable boulders sitting at points $(10,11)$ and $(21,20)$? (You do not have to calculate the number explicitly; your answer may be an expression involving binomial coefficients.)

Hint: Inclusion-Exclusion.

Exercise 3.9.7 A *derangement* is a permutation $(x_1, x_2, \ldots, x_n)$ of the set $\{1, 2, \ldots, n\}$ such that $x_i \neq i$ for all i. For example, $(2,3,4,5,1)$ is a derangement, but $(2,1,3,5,4)$ is not because 3 appears in the third position. The objective of this problem is to count derangements.

It turns out to be easier to start by counting the permutations that are *not* derangements. Let S_i be the set of all permutations $(x_1, x_2, \ldots, x_n)$ that are not derangements because $x_i = i$. So the set of non-derangements is

$$\bigcup_{i=1}^{n} S_i.$$

(a) What is $|S_i|$?

(b) What is $|S_i \cap S_j|$ where $i \neq j$?

(c) What is $|S_{i_1} \cap S_{i_2} \cap \cdots \cap S_{i_k}|$ where $i_1, i_2, \ldots, i_k$ are all distinct?

(d) Use the inclusion-exclusion formula to express the number of non-derangements in terms of sizes of possible intersections of the sets $S_1, \ldots, S_n$.

(e) How many terms in the expression in part (3.9.7) have the form $|S_{i_1} \cap S_{i_2} \cap \cdots \cap S_{i_k}|$?

(f) Combine your answers to the preceding parts to prove the number of non-derangements is:

$$n!\left(\frac{1}{1!} - \frac{1}{2!} + \frac{1}{3!} - \cdots \pm \frac{1}{n!}\right).$$

Conclude that the number of derangements is $n!\left(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \cdots \pm \frac{1}{n!}\right)$.

(g) As n goes to infinity, the number of derangements approaches a constant fraction of all permutations. What is that constant? *Hint:* $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$.

Exercise 3.9.8 There are 10 students $A, B, \ldots, J$ who will be lined up left to right according to the some rules below.

Rule I: Student A must not be rightmost.

Rule II: Student B must be adjacent to C (directly to the left or right of C).

Rule III: Student D is always second.

You may answer the following questions with a numerical formula that may involve factorials.

(a) How many possible lineups are there that satisfy all three of these rules?

(b) How many possible lineups are there that satisfy at least one of these rules?

Exercise 3.9.9 A robot on a point in the 3-D integer lattice can move a unit distance in one positive direction at a time. That is, from position (x,y,z), it can move to either $(x+1,y,z)$, $(x,y+1,z)$ or $(x,y,z+1)$. For any two points P and Q in space, let $n(P,Q)$ denote the number of distinct paths the spacecraft can follow to go from P to Q.

Let $A = (0,10,20), B = (30,50,70), C = (80,90,100), D = (200,300,400)$.

(a) Express $n(A, B)$ as a *single multinomial coefficient.*

Answer the following questions with arithmetic expressions involving terms $n(P, Q)$ for $P, Q \in \{A, B, C, D\}$. Do not use numbers.

(b) How many paths from A to C go through B?

(c) How many paths from B to D do *not* go through C?

(d) How many paths from A to D go through *neither* B *nor* C?

3.10 Combinatorial Proofs

Suppose you have n different T-shirts, but only want to keep k. You could equally well select the k shirts you want to keep or select the complementary set of $n - k$ shirts you want to throw out. Thus, the number of ways to select k shirts from among n must be equal to the number of ways to select $n - k$ shirts from among n. Therefore:

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

This is easy to prove algebraically, since both sides are equal to:

$$\frac{n!}{k!\,(n-k)!}.$$

But we didn't really have to resort to algebra; we just used counting principles.

Hmmm....

3.10.1 Pascal's Triangle Identity

Bob, famed Math for Computer Science Teaching Assistant, has decided to try out for the US Olympic boxing team. After all, he's watched all of the *Rocky* movies and spent hours in front of a mirror sneering, "Yo, you wanna piece a' *me*?!" Bob figures that n people (including himself) are competing for spots on the team and only k will be selected. As part of maneuvering for a spot on the team, he needs to work out how many different teams are possible. There are two cases to consider:

- Bob *is* selected for the team, and his $k - 1$ teammates are selected from among the other $n - 1$ competitors. The number of different teams that can be formed in this way is:

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

- Bob is *not* selected for the team, and all k team members are selected from among the other $n - 1$ competitors. The number of teams that can be formed this way is:

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

All teams of the first type contain Bob, and no team of the second type does; therefore, the two sets of teams are disjoint. Thus, by the Sum Rule, the total number of possible Olympic boxing teams is:

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

Ted, equally-famed Teaching Assistant, thinks Bob isn't so tough and so he might as well also try out. He reasons that n people (including himself) are trying out for k spots. Thus, the number of ways to select the team is simply:

$$\binom{n}{k}.$$

Ted and Bob each correctly counted the number of possible boxing teams. Thus, their answers must be equal. So we know:

Lemma 41 (Pascal's *Triangle Identity*)**.**

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}. \tag{3.18}$$

We proved *Pascal's Triangle Identity without any algebra!* Instead, we relied purely on counting techniques.

3.10.2 Giving a Combinatorial Proof

A *combinatorial proof* is an argument that establishes an algebraic fact by relying on counting principles. Many such proofs follow the same basic outline:

1. Define a set S.
2. Show that $|S| = n$ by counting one way.
3. Show that $|S| = m$ by counting another way.
4. Conclude that $n = m$.

In the preceding example, S was the set of all possible Olympic boxing teams. Bob computed

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

by counting one way, and Ted computed

$$|S| = \binom{n}{k}$$

by counting another way. Equating these two expressions gave Pascal's Identity.

Checking a Combinatorial Proof

Combinatorial proofs are based on counting the same thing in different ways. This is fine when you've become practiced at different counting methods, but when in doubt, you can fall back on bijections and sequence counting to check such proofs.

For example, let's take a closer look at the combinatorial proof of Pascal's Identity (3.18). In this case, the set S of things to be counted is the collection of all size-k subsets of integers in the interval $[1..n]$.

Now we've already counted S one way, via the Bookkeeper Rule, and found $|S| = \binom{n}{k}$. The other "way" corresponds to defining a bijection between S and the disjoint union of two sets A and B where,

$$A \triangleq \{(1, X) \mid X \subseteq [2..n] \text{ AND } |X| = k-1\}$$
$$B \triangleq \{(0, Y) \mid Y \subseteq [2..n] \text{ AND } |Y| = k\}.$$

Clearly A and B are disjoint since the pairs in the two sets have different first coordinates, so $|A \cup B| = |A| + |B|$. Also,

$$|A| = \#\text{ specified sets } X = \binom{n-1}{k-1},$$
$$|B| = \#\text{ specified sets } Y = \binom{n-1}{k}.$$

Now finding a bijection $f : (A \cup B) \to S$ will prove the identity (3.18). In particular, we can define

$$f(c) \triangleq \begin{cases} X \cup \{1\} & \text{if } c = (1, X), \\ Y & \text{if } c = (0, Y). \end{cases}$$

It should be obvious that f is a bijection.

3.10.3 A Colorful Combinatorial Proof

The set that gets counted in a combinatorial proof in different ways is usually defined in terms of simple sequences or sets rather than an elaborate story about Teaching Assistants. Here is another colorful example of a combinatorial argument.

Theorem 42.

$$\sum_{r=0}^{n} \binom{n}{r}\binom{2n}{n-r} = \binom{3n}{n}$$

Proof. We give a combinatorial proof. Let S be all n-card hands that can be dealt from a deck containing n different red cards and $2n$ different black cards. First, note that every $3n$-element set has

$$|S| = \binom{3n}{n}$$

n-element subsets.

From another perspective, the number of hands with exactly r red cards is

$$\binom{n}{r}\binom{2n}{n-r}$$

since there are $\binom{n}{r}$ ways to choose the r red cards and $\binom{2n}{n-r}$ ways to choose the $n - r$ black cards. Since the number of red cards can be anywhere from 0 to n, the total number of n-card hands is:

$$|S| = \sum_{r=0}^{n} \binom{n}{r}\binom{2n}{n-r}.$$

Equating these two expressions for $|S|$ proves the theorem. □

Finding a Combinatorial Proof

Combinatorial proofs are almost magical. Theorem 42 looks pretty scary, but we proved it without any algebraic manipulations at all. The key to constructing a combinatorial proof is choosing the set S properly, which can be tricky. Generally, the simpler side of the equation should provide some guidance. For example, the right side of Theorem 42 is $\binom{3n}{n}$, which suggests that it will be helpful to choose S to be all n-element subsets of some $3n$-element set.

Practice Problems

Exercise 3.10.1 Prove the following identity by algebraic manipulation and by giving a combinatorial argument:

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

Exercise 3.10.2 Conside the following identity

$$n2^{n-1} = \sum_{k=1}^{n} k\binom{n}{k} \tag{3.19}$$

(a) Give a combinatorial proof of the identity (3.19) by letting S be the set of all length-n sequences of letters a, b and a single c and counting $|S|$ in two different ways.
(b) Now prove (3.19) algebraically by applying the Binomial Theorem to $(1+x)^n$ and taking derivatives.

Exercise 3.10.3 What do the following expressions equal? Give both algebraic and combinatorial proofs for your answers.
(a) $\sum_{i=0}^{n}\binom{n}{i}$
(b) $\sum_{i=0}^{n}\binom{n}{i}(-1)^i$
Hint: Consider the bit strings with an even number of ones and an odd number of ones.

Exercise 3.10.4 When an integer k occurs as the kth element of a sequence, we'll say it is "in place" in the sequence. For example, in the sequence

$$12453678$$

precisely the integers $1, 2, 6, 7$ and 8 occur in place. We're going to classify the sequences of distinct integers from 1 to n, that is the permutations of $[1..n]$, according to which integers do not occur "in place." Then we'll use this classification to prove the combinatorial identity[9]

$$n! = 1 + \sum_{k=1}^{n}(k-1)\cdot(k-1)!\,. \tag{3.20}$$

If π is a permutation of $[1..n]$, let $\mathsf{mnp}\{\pi\}$ be the *maximum* integer in $[1..n]$ that does not occur in place in π. For example, for $n = 8$,

$$\begin{aligned}\mathsf{mnp}\{12345687\} &= 8,\\ \mathsf{mnp}\{21345678\} &= 2,\\ \mathsf{mnp}\{23145678\} &= 3.\end{aligned}$$

(a) For how many permutations of $[1..n]$ is every element in place?

(b) How many permutations π of $[1..n]$ have $\mathsf{mnp}\{\pi\} = 1$?

(c) How many permutations of $[1..n]$ have $\mathsf{mnp}\{\pi\} = k$?

(d) Conclude the equation (3.20).

[9]This problem is based on "Use of everywhere divergent generating function," **mathoverflow**, response 8,147 by Aaron Meyerowitz, Nov. 12, 2010.

Exercise 3.10.5 Each day, an MIT student selects a breakfast from among b possibilities, lunch from among l possibilities, and dinner from among d possibilities. In each case one of the possibilities is Doritos. However, a legimate daily menu may include Doritos for at most one meal. Give a combinatorial (not algebraic) proof based on the number of legimate daily menus that

$$\begin{aligned} bld - [(b-1) + (l-1) + (d-1) + 1] \\ = b(l-1)(d-1) + (b-1)l(d-1) + (b-1)(l-1)d \\ -3(b-1)(l-1)(d-1) + (b-1)(l-1)(d-1) \end{aligned}$$

Hint: Let M_b be the number of menus where, if Doritos appear at all, they only appear at *b*reakfast; likewise, for M_l, M_d.

Exercise 3.10.6 The following problems have to do with combinatorial proof.

(a) Find a combinatorial (*not* algebraic) proof that $\sum_{i=0}^{n} \binom{n}{i} = 2^n$.

(b) Below is a combinatorial proof of an equation. What is the equation?

Proof. Stinky Peterson owns n newts, t toads, and s slugs. Conveniently, he lives in a dorm with $n+t+s$ other students. (The students are distinguishable, but creatures of the same variety are not distinguishable.) Stinky wants to put one creature in each neighbor's bed. Let W be the set of all ways in which this can be done.

On one hand, he could first determine who gets the slugs. Then, he could decide who among his remaining neighbors has earned a toad. Therefore, $|W|$ is equal to the expression on the left.

On the other hand, Stinky could first decide which people deserve newts and slugs and then, from among those, determine who truly merits a newt. This shows that $|W|$ is equal to the expression on the right.

Since both expressions are equal to $|W|$, they must be equal to each other. □

(Combinatorial proofs are real proofs. They are not only rigorous, but also convey an intuitive understanding that a purely algebraic argument might not reveal. However, combinatorial proofs are usually less colorful than this one.)

Exercise 3.10.7 You want to choose a team of m people for your startup company from a pool of n applicants, and from these m people you want to choose k to be the team managers. You took a Math for Computer Science subject, so you know you can do this in $\binom{n}{m}\binom{m}{k}$ ways. But your CFO, who went to Harvard Business School, comes up with the formula $\binom{n}{k}\binom{n-k}{m-k}$. Before doing the reasonable thing—dump on your CFO or Harvard Business School—you decide to check his answer against yours.

(a) Give a combinatorial proof that your CFO's formula agrees with yours.

(b) Verify this combinatorial proof by giving an *algebraic* proof of this same fact.

Exercise 3.10.8 Give a combinatorial proof of

$$1 \cdot 2 + 2 \cdot 3 + 3 \cdot 4 + \cdots + (n-1) \cdot n = 2\binom{n+1}{3}$$

Hint: Classify sets of three numbers from the integer interval $[0..n]$ by their maximum element.

4 Recurrences

A recurrence describes a sequence of numbers. Early terms are specified explicitly and later terms are expressed as a function of their predecessors. As a trivial example, this recurrence describes the sequence 1, 2, 3, etc.:

$$T_1 = 1$$
$$T_n = T_{n-1} + 1 \qquad \text{(for } n \geq 2\text{)}.$$

Here, the first term is defined to be 1 and each subsequent term is one more than its predecessor.

Recurrences turn out to be a powerful tool. In this chapter, we'll emphasize using recurrences to analyze the performance of recursive algorithms. However, recurrences have other applications in computer science as well, such as enumeration of structures and analysis of random processes. And, as we saw in the *book stacking problem* in Section 2.1.1, they also arise in the analysis of problems in the physical sciences.

A recurrence in isolation is not a very useful description of a sequence. One can not easily answer simple questions such as, "What is the hundredth term?" or "What is the asymptotic growth rate?" So one typically wants to *solve* a recurrence; that is, to find a closed-form expression for the nth term.

We'll first introduce two general solving techniques: guess-and-verify and plug-and-chug. These methods are applicable to every recurrence, but their success requires a flash of insight—sometimes an unrealistically brilliant flash. So we'll also introduce two big classes of recurrences, linear and divide-and-conquer, that often come up in computer science. Essentially all recurrences in these two classes are solvable using cookbook techniques; you follow the recipe and get the answer. A drawback is that calculation replaces insight. The "Aha!" moment that is essential in the guess-and-verify and plug-and-chug methods is replaced by a "Huh" at the end of a cookbook procedure.

At the end of the chapter, we'll develop rules of thumb to help you assess many recurrences without any calculation. These rules can help you distinguish promising approaches from bad ideas early in the process of designing an algorithm.

Recurrences are one aspect of a broad theme in computer science: reducing a big problem to progressively smaller problems until easy base cases are reached. This same idea underlies both induction proofs and recursive algorithms. As we'll see, all three ideas snap together nicely. For example, one might describe the running time of a recursive algorithm with a recurrence and use induction to verify the solution.

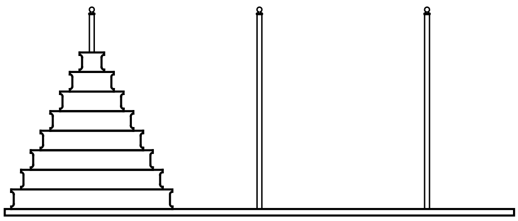


Figure 4.1: The initial configuration of the disks in the Towers of Hanoi problem.

4.1 The Towers of Hanoi

According to legend, there is a temple in Hanoi with three posts and 64 gold disks of different sizes. Each disk has a hole through the center so that it fits on a post. In the misty past, all the disks were on the first post, with the largest on the bottom and the smallest on top, as shown in Figure 4.1.

Monks in the temple have labored through the years since to move all the disks to one of the other two posts according to the following rules:

- The only permitted action is removing the top disk from one post and dropping it onto another post.

- A larger disk can never lie above a smaller disk on any post.

So, for example, picking up the whole stack of disks at once and dropping them on another post is illegal. That's good, because the legend says that when the monks complete the puzzle, the world will end!

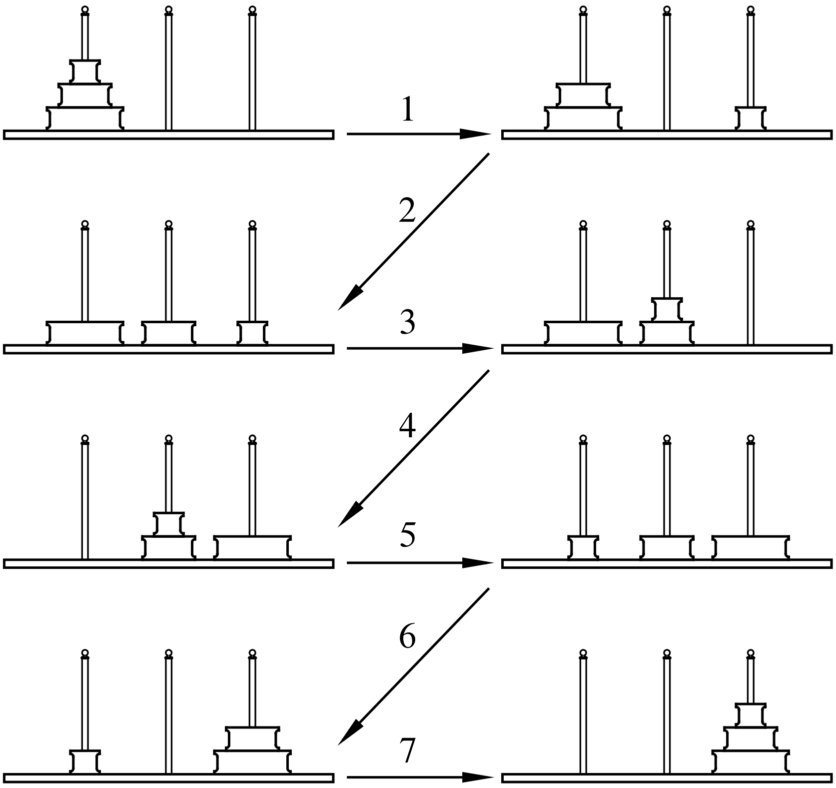


Figure 4.2: The 7-step solution to the Towers of Hanoi problem when there are $n = 3$ disks.

To clarify the problem, suppose there were only 3 gold disks instead of 64. Then the puzzle could be solved in 7 steps as shown in Figure 4.2.

The questions we must answer are, "Given sufficient time, can the monks succeed?" If so, "How long until the world ends?" And, most importantly, "Will this happen before the final exam?"

4.1.1 A Recursive Solution

The Towers of Hanoi problem can be solved recursively. As we describe the procedure, we'll also analyze the running time. To that end, let T_n be the minimum number of steps required to solve the n-disk problem. For example, some experimentation shows that $T_1 = 1$ and $T_2 = 3$. The procedure illustrated above shows that T_3 is at most 7, though there might be a solution with fewer steps.

The recursive solution has three stages, which are described below and illustrated in Figure 4.3. For clarity, the largest disk is shaded in the figures.

Stage 1. Move the top $n - 1$ disks from the first post to the second using the solution for $n - 1$ disks. This can be done in T_{n-1} steps.

Stage 2. Move the largest disk from the first post to the third post. This takes just 1 step.

Stage 3. Move the $n - 1$ disks from the second post to the third post, again using the solution for $n - 1$ disks. This can also be done in T_{n-1} steps.

This algorithm shows that T_n, the minimum number of steps required to move n disks to a different post, is at most $T_{n-1} + 1 + T_{n-1} = 2T_{n-1} + 1$.

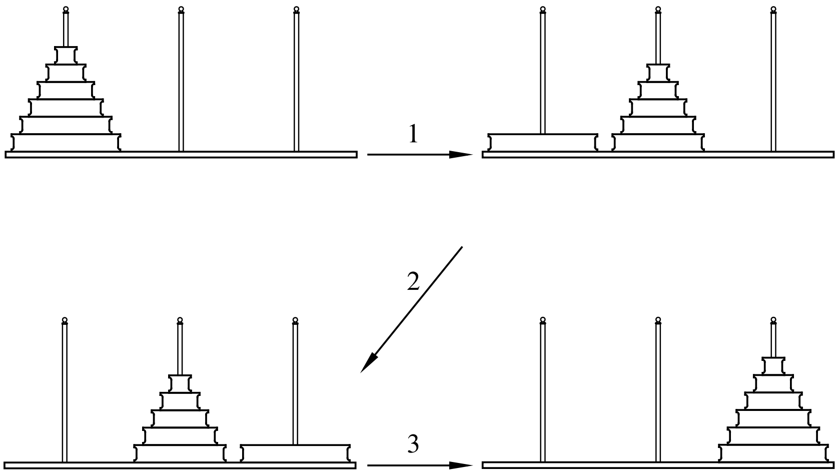


Figure 4.3: A recursive solution to the Towers of Hanoi problem.

We can use this fact to upper bound the number of operations required to move towers of various heights:

$$T_3 \leq 2 \cdot T_2 + 1 = 7$$
$$T_4 \leq 2 \cdot T_3 + 1 \leq 15$$

Continuing in this way, we could eventually compute an upper bound on the number T_{64} of steps required to move 64 disks. So this algorithm answers our first question: given sufficient time, the monks can finish their task and end the world. This is a shame. After all that effort, they'd probably want to smack a few high-fives and go out for burgers and ice cream, but nope—world's over.

4.1.2 Finding a Recurrence

We cannot yet compute the exact number of steps that the monks need to move the 64 disks, only an upper bound. Perhaps, having pondered the problem since the beginning of time, the monks have devised a better algorithm.

In fact, there is no better algorithm, and here is why. At some step, the monks must move the largest disk from the first post to a different post. For this to happen, the $n-1$ smaller disks must all be stacked out of the way on the only remaining post. Arranging the $n-1$ smaller disks this way requires at least T_{n-1} moves. After the largest disk is moved, at least another T_{n-1} moves are required to pile the $n-1$ smaller disks on top.

This argument shows that the number of steps required is at least $2T_{n-1}+1$. Since we gave an algorithm using exactly that number of steps, we can now write an expression for the number T_n of moves required to complete the Towers of Hanoi problem with n disks:

$$\begin{aligned} T_1 &= 1 \\ T_n &= 2T_{n-1} + 1 \qquad \text{(for } n \geq 2\text{)}. \end{aligned}$$

This is a typical recurrence. These two lines define a sequence of values, $T_1, T_2, T_3, \ldots$. The first line says that the first number T_1 in the sequence is equal to 1. The second line defines every other number in the sequence in terms of its predecessor. So we can use this recurrence to compute any number of terms in the sequence:

$$\begin{aligned} T_1 &= 1 & T_4 &= 2 \cdot T_3 + 1 = 15 \\ T_2 &= 2 \cdot T_1 + 1 = 3 & T_5 &= 2 \cdot T_4 + 1 = 31 \\ T_3 &= 2 \cdot T_2 + 1 = 7 & T_6 &= 2 \cdot T_5 + 1 = 63. \end{aligned}$$

4.1.3 Solving the Recurrence

We could determine the number of steps to move a 64-disk tower by computing T_7, T_8, and so on up to T_{64}. But that would take a lot of work. It would be nice to have a closed-form expression for T_n so that we could quickly find the number of steps required for any given number of disks. (For example, we might want to know how much sooner the world would end if the monks melted down one disk to purchase burgers and ice cream *before* the end of the world.)

There are several methods for solving recurrence equations. The simplest is to *guess* the solution and then *verify* that the guess is correct with an induction proof.

As a basis for a good guess, let's look for a pattern in the values of T_n computed above: 1, 3, 7, 15, 31, 63. A natural guess is $T_n = 2^n - 1$. But whenever you guess a solution to a recurrence, you should always verify it with a proof, typically by induction. After all, your guess might be wrong. (But why bother to verify in this case? After all, if we're wrong, its not the end of the... no, let's check.)

Claim 43. $T_n = 2^n - 1$ *satisfies the recurrence:*

$$\begin{aligned} T_1 &= 1 \\ T_n &= 2T_{n-1} + 1 \qquad \textit{(for } n \geq 2\textit{)}. \end{aligned}$$

Proof. The proof is by induction on n. The induction hypothesis is that $T_n = 2^n - 1$. This is true for $n = 1$ because $T_1 = 1 = 2^1 - 1$. Now assume that $T_{n-1} = 2^{n-1} - 1$ in order to prove that $T_n = 2^n - 1$, where $n \geq 2$:

$$\begin{aligned} T_n &= 2T_{n-1} + 1 \\ &= 2(2^{n-1} - 1) + 1 \\ &= 2^n - 1. \end{aligned}$$

The first equality is the recurrence equation, the second follows from the induction assumption, and the last step is simplification. □

Such verification proofs are especially tidy because recurrence equations and induction proofs have analogous structures. In particular, the base case relies on the first line of the recurrence, which defines T_1. And the inductive step uses the second line of the recurrence, which defines T_n as a function of preceding terms.

Our guess is verified. So we can now resolve our remaining questions about the 64-disk puzzle. Since $T_{64} = 2^{64} - 1$, the monks must complete more than 18 billion billion steps before the world ends. Better study for the final.

4.1.4 The Upper Bound Trap

When the solution to a recurrence is complicated, one might try to prove that some simpler expression is an upper bound on the solution. For example, the exact solution to the Towers of Hanoi recurrence is $T_n = 2^n - 1$. Let's try to prove the "nicer" upper bound $T_n \leq 2^n$, proceeding exactly as before.

Proof. (Failed attempt.) The proof is by induction on n. The induction hypothesis is that $T_n \leq 2^n$. This is true for $n = 1$ because $T_1 = 1 \leq 2^1$. Now assume that $T_{n-1} \leq 2^{n-1}$ in order to prove that $T_n \leq 2^n$, where $n \geq 2$:

$$\begin{aligned} T_n &= 2T_{n-1} + 1 \\ &\leq 2(2^{n-1}) + 1 \\ &\not\leq 2^n \qquad \text{IMPLIES Uh-oh!} \end{aligned}$$

The first equality is the recurrence relation, the second follows from the induction hypothesis, and the third step is a flaming train wreck. □

The proof doesn't work! As is so often the case with induction proofs, the argument only goes through with a *stronger* hypothesis. This isn't to say that upper bounding the solution to a recurrence is hopeless, but this is a situation where induction and recurrences do not mix well.

4.1.5 Plug and Chug

Guess-and-verify is a simple and general way to solve recurrence equations. But there is one big drawback: you have to *guess right*. That was not hard for the Towers of Hanoi example. But sometimes the solution to a recurrence has a strange form that is quite difficult to guess. Practice helps, of course, but so can some other methods.

Plug-and-chug is another way to solve recurrences. This is also sometimes called "expansion" or "iteration." As in guess-and-verify, the key step is identifying a pattern. But instead of looking at a sequence of *numbers*, you have to spot a pattern in a sequence of *expressions*, which is sometimes easier. The method consists of three steps, which are described below and illustrated with the Towers of Hanoi example.

Step 1: Plug and Chug Until a Pattern Appears

The first step is to expand the recurrence equation by alternately "plugging" (applying the recurrence) and "chugging" (simplifying the result) until a pattern appears. Be careful: too much simplification can make a pattern harder to spot. The rule to remember—indeed, a rule applicable to the whole of college life—is *chug in moderation*.

$$\begin{aligned}
T_n &= 2T_{n-1} + 1 & \\
&= 2(2T_{n-2} + 1) + 1 & \text{plug} \\
&= 4T_{n-2} + 2 + 1 & \text{chug} \\
&= 4(2T_{n-3} + 1) + 2 + 1 & \text{plug} \\
&= 8T_{n-3} + 4 + 2 + 1 & \text{chug} \\
&= 8(2T_{n-4} + 1) + 4 + 2 + 1 & \text{plug} \\
&= 16T_{n-4} + 8 + 4 + 2 + 1 & \text{chug}
\end{aligned}$$

Above, we started with the recurrence equation. Then we replaced T_{n-1} with $2T_{n-2} + 1$, since the recurrence says the two are equivalent. In the third step, we simplified a little—but not too much! After several similar rounds of plugging and chugging, a pattern is apparent. The following formula seems to hold:

$$\begin{aligned}
T_n &= 2^k T_{n-k} + 2^{k-1} + 2^{k-2} + \cdots + 2^2 + 2^1 + 2^0 \\
&= 2^k T_{n-k} + 2^k - 1
\end{aligned}$$

Once the pattern is clear, simplifying is safe and convenient. In particular, we've collapsed the geometric sum to a closed form on the second line.

Step 2: Verify the Pattern

The next step is to verify the general formula with one more round of plug-and-chug.

$$\begin{aligned}
T_n &= 2^k T_{n-k} + 2^k - 1 & \\
&= 2^k (2T_{n-(k+1)} + 1) + 2^k - 1 & \text{plug} \\
&= 2^{k+1} T_{n-(k+1)} + 2^{k+1} - 1 & \text{chug}
\end{aligned}$$

The final expression on the right is the same as the expression on the first line, except that k is replaced by $k + 1$. Surprisingly, this effectively *proves* that the formula is correct for all k. Here is why: we know the formula holds for $k = 1$, because that's the original recurrence equation. And we've just shown that if the formula holds for some $k \geq 1$, then it also holds for $k + 1$. So the formula holds for all $k \geq 1$ by induction.

Step 3: Write T_n Using Early Terms with Known Values

The last step is to express T_n as a function of early terms whose values are known. Here, choosing $k = n - 1$ expresses T_n in terms of T_1, which is equal to 1. Simplifying gives a closed-form expression for T_n:

$$\begin{aligned} T_n &= 2^{n-1}T_1 + 2^{n-1} - 1 \\ &= 2^{n-1} \cdot 1 + 2^{n-1} - 1 \\ &= 2^n - 1. \end{aligned}$$

We're done! This is the same answer we got from guess-and-verify.

Let's compare guess-and-verify with plug-and-chug. In the guess-and-verify method, we computed several terms at the beginning of the sequence T_1, T_2, T_3, etc., until a pattern appeared. We generalized to a formula for the nth term T_n. In contrast, plug-and-chug works backward from the nth term. Specifically, we started with an expression for T_n involving the preceding term T_{n-1}, and rewrote this using progressively earlier terms T_{n-2}, T_{n-3}, etc. Eventually, we noticed a pattern, which allowed us to express T_n using the very first term T_1 whose value we knew. Substituting this value gave a closed-form expression for T_n. So guess-and-verify and plug-and-chug tackle the problem from opposite directions.

Practice Problems

Exercise 4.1.1 Find the shortest sequence of moves that transfers a tower of n disks from the left peg A to the right peg B, if direct moves between A and B are disallowed. (Each move must be to or from the middle peg. As usual, a larger disk must never appear above a smaller one.)

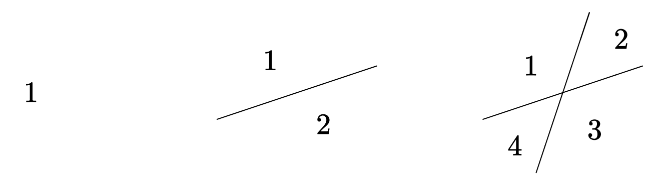
Exercise 4.1.2 Show that, in the process of transferring a tower under the restrictions of the preceding exercise, we will actually encounter every properly stacked arrangement of n disks on three pegs.

4.2 Lines in the Plane

Let us now dive into a problem of geometric flavor: How many slices of "pizza" can a person obtain by making n straight cuts with a pizza knife? Or, more academically: What is the maximum number L_n of "regions" defined by n "lines in the plane"? This problem was first solved in 1826, by the Swiss mathematician Jacob Steiner.

4.2.1 Finding a Recurrence

We start by looking at "small cases", remembering to begin with the smallest of all. The plane with no lines has one region; with one line it has two regions; and with two lines it has four regions.

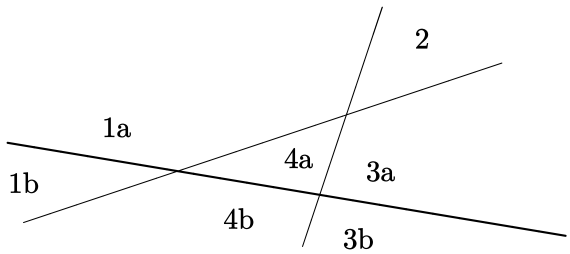


$L_0 = 1$ $L_1 = 2$ $L_2 = 4$

Each line extends infinitely in both directions.

Sure, we think, $L_n = 2^n$; of course! Adding a new line simply doubles the number of regions. Unfortunately this is wrong. We could achieve the doubling if the nth line would split each old region in two; certainly it can split an old region in at most two pieces, since each old region is "convex".[1] (A straight line can split a convex region into at most two new regions, which will also be convex.)

But when we add the third line – the thick one in the diagram – we soon find that it can split at most three of the old regions, no matter how we've placed the first two lines.
Thus $L_3 = 4 + 3 = 7$ is the best we can do.



And after some thought we realize the appropriate generalization. The nth line (for $n > 0$) increases the number of regions by k if and only if it splits k of the old regions, and it splits k old regions if and only if it hits the previous lines in $k-1$ different places. Two lines can intersect in at most one point. Therefore the new line can intersect the $n-1$ old lines in at most $n-1$ different points, and we must have $k \le n$. We have established the upper bound

$$L_n \le L_{n-1} + n\,, \qquad \text{for } n > 0.$$

Furthermore it's easy to show by induction that we can achieve equality in this formula. We simply place the nth line in such a way that it's not parallel to any of the others (hence it intersects them all), and such that it doesn't go through any of the existing intersection points (hence it intersects them all in different places). The recurrence is therefore

$$L_0 = 1\,; \tag{4.1}$$
$$L_n = L_{n-1} + n\,, \qquad \text{for } n > 0. \tag{4.2}$$

The known values of L_1, L_2, and L_3 check perfectly here, so we'll buy this.

4.2.2 Solving the Recurrence

Now we need a closed-form solution. We could try to make a good guess, but 1, 2, 4, 7, 11, 16, ... doesn't look familiar; so let's try another tack. We can often understand a recurrence by "unfolding" or "unwinding" it all the way to the end, as follows:

$$\begin{aligned} L_n &= L_{n-1} + n \\ &= L_{n-2} + (n-1) + n \\ &= L_{n-3} + (n-2) + (n-1) + n \\ &\ \ \vdots \\ &= L_0 + 1 + 2 + \cdots + (n-2) + (n-1) + n \\ &= 1 \ + \ S_n\,, \qquad \text{where } S_n = 1 + 2 + 3 + \cdots + (n-1) + n. \end{aligned}$$

In other words, L_n is one more than the sum S_n of the first n positive integers.

As we covered it elsewhere, employing a trick produced by C.F. Gauss in 1786, when he was nine years old, S_n can be evaluated as follows

$$S_n = \frac{n(n+1)}{2}\,, \qquad \text{for } n \ge 0. \tag{4.3}$$

OK, we have our solution:

$$L_n = \frac{n(n+1)}{2} + 1\,, \qquad \text{for } n \ge 0. \tag{4.4}$$

[1]A region is convex if it includes all line segments between any two of its points. (That may not be what your dictionary says, but it's what mathematicians believe!)

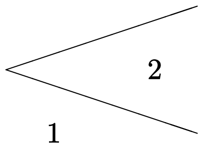
We might be satisfied with this derivation and consider it a proof, even though we waved our hands a bit when doing the unfolding and reflecting. But mathematically sophisticated students should be able to meet stricter standards; so it's a good idea to construct a rigorous "proof" by "induction". The key induction step is

$$L_n = L_{n-1} + n = \left(\tfrac{1}{2}(n-1)n + 1\right) + n = \tfrac{1}{2}n(n+1) + 1\,.$$

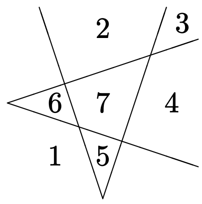
Now there can be no doubt about the closed form (4.4).

And now, briefly, a variation of the lines-in-the-plane problem: Suppose that instead of straight lines we use bent lines, each containing one “zig". What is the maximum number Z_n of regions determined by n such bent lines in the plane?

We might expect Z_n to be about twice as big as L_n, or maybe three times as big. Let's see:



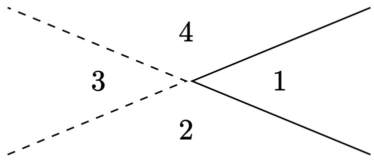
$Z_1 = 2$



$Z_2 = 7$

From these small cases, and after a little thought (and maybe a bit of *afterthought*) we realize that a bent line is like two straight lines except that regions merge when the “two” lines don't extend past their intersection point.

Regions 2, 3, and 4, which would be distinct with two lines, become a single region when there's a bent line; we lose two regions.



However, if we arrange things properly — the zig point must lie “beyond” the intersections with the other lines – that's all we lose; that is, we lose only two regions per line. Thus

$$\begin{aligned} Z_n = L_{2n} - 2n &= 2n(2n+1)/2 + 1 - 2n \\ &= 2n^2 - n + 1\,, \qquad \text{for } n \geq 0. \end{aligned} \tag{4.5}$$

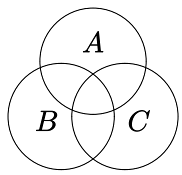
Comparing the closed forms (4.4) and (4.5), we find that for large n,

$$\begin{aligned} L_n &\sim \tfrac{1}{2}n^2\,, \\ Z_n &\sim 2n^2\,; \end{aligned}$$

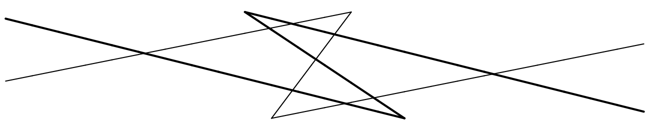
so we get about four times as many regions with bent lines as with straight lines. The '$\sim$' symbol denotes *asymptotic equality*, and is formally defined in Section 2.1.4.

Practice Problems

Exercise 4.2.1 A “Venn diagram” with three overlapping circles is often used to illustrate the eight possible subsets associated with three given sets. Can the sixteen possibilities that arise with four given sets be illustrated by four overlapping circles?



Exercise 4.2.2 What's the maximum number of regions definable by n "zig-zag" lines, each of which consists of two parallel infinite half-lines joined by a straight segment?



$ZZ_2 = 12$

4.3 Merge Sort

Let us bring into picture an instance of a somewhat different kind of recurrence that shows up in the analysis of *Merge Sort*, a sorting algorithm you are presumably familiar with.

4.3.1 Finding a Recurrence

A traditional question about sorting algorithms is, "What is the maximum number of comparisons used in sorting n items?" This is taken as an estimate of the running time. In the case of Merge Sort, we can express this quantity with a recurrence. Let T_n be the maximum number of comparisons used while Merge Sorting a list of n numbers. For now, assume that n is a power of 2. This ensures that the input can be divided in half at every stage of the recursion.

- If there is only one number in the list, then no comparisons are required, so $T_1 = 0$.
- Otherwise, T_n includes comparisons used in sorting the first half (at most $T_{n/2}$), in sorting the second half (also at most $T_{n/2}$), and in merging the two halves. The number of comparisons in the merging step is at most $n-1$. This is because at least one number is emitted after each comparison and one more number is emitted at the end when one list becomes empty. Since n items are emitted in all, there can be at most $n-1$ comparisons.

Therefore, the maximum number of comparisons needed to Merge Sort n items is given by this recurrence:

$$\begin{aligned} T_1 &= 0 \\ T_n &= 2T_{n/2} + n - 1 && \text{(for } n \geq 2 \text{ and a power of 2).} \end{aligned}$$

This fully describes the number of comparisons, but not in a very useful way; a closed-form expression would be much more helpful. To get that, we have to solve the recurrence.

4.3.2 Solving the Recurrence

Let's first try to solve the Merge Sort recurrence with the guess-and-verify technique. Here are the first few values:

$$\begin{aligned} T_1 &= 0 \\ T_2 &= 2T_1 + 2 - 1 = 1 \\ T_4 &= 2T_2 + 4 - 1 = 5 \\ T_8 &= 2T_4 + 8 - 1 = 17 \\ T_{16} &= 2T_8 + 16 - 1 = 49. \end{aligned}$$

We're in trouble! Guessing the solution to this recurrence is hard because there is no obvious pattern. So let's try the plug-and-chug method instead.

Step 1: Plug and Chug Until a Pattern Appears

First, we expand the recurrence equation by alternately plugging and chugging until a pattern appears.

$$\begin{aligned} T_n &= 2T_{n/2} + n - 1 \\ &= 2(2T_{n/4} + n/2 - 1) + (n-1) && \text{plug} \\ &= 4T_{n/4} + (n-2) + (n-1) && \text{chug} \\ &= 4(2T_{n/8} + n/4 - 1) + (n-2) + (n-1) && \text{plug} \\ &= 8T_{n/8} + (n-4) + (n-2) + (n-1) && \text{chug} \\ &= 8(2T_{n/16} + n/8 - 1) + (n-4) + (n-2) + (n-1) && \text{plug} \end{aligned}$$

$$= 16T_{n/16} + (n-8) + (n-4) + (n-2) + (n-1) \qquad \text{chug}$$

A pattern is emerging. In particular, this formula seems holds:

$$\begin{aligned}
T_n &= 2^k T_{n/2^k} + (n - 2^{k-1}) + (n - 2^{k-2}) + \cdots + (n - 2^0) \\
&= 2^k T_{n/2^k} + kn - 2^{k-1} - 2^{k-2} \cdots - 2^0 \\
&= 2^k T_{n/2^k} + kn - 2^k + 1.
\end{aligned}$$

On the second line, we grouped the n terms and powers of 2. On the third, we collapsed the geometric sum.

Step 2: Verify the Pattern

Next, we verify the pattern with one additional round of plug-and-chug. If we guessed the wrong pattern, then this is where we'll discover the mistake.

$$\begin{aligned}
T_n &= 2^k T_{n/2^k} + kn - 2^k + 1 \\
&= 2^k (2T_{n/2^{k+1}} + n/2^k - 1) + kn - 2^k + 1 && \text{plug} \\
&= 2^{k+1} T_{n/2^{k+1}} + (k+1)n - 2^{k+1} + 1 && \text{chug}
\end{aligned}$$

The formula is unchanged except that k is replaced by $k+1$. This amounts to the induction step in a proof that the formula holds for all $k \geq 1$.

Step 3: Write T_n Using Early Terms with Known Values

Finally, we express T_n using early terms whose values are known. Specifically, if we let $k = \log n$, then $T_{n/2^k} = T_1$, which we know is 0:

$$\begin{aligned}
T_n &= 2^k T_{n/2^k} + kn - 2^k + 1 \\
&= 2^{\log n} T_{n/2^{\log n}} + n \log n - 2^{\log n} + 1 \\
&= nT_1 + n \log n - n + 1 \\
&= n \log n - n + 1.
\end{aligned}$$

We're done! We have a closed-form expression for the maximum number of comparisons used in Merge Sorting a list of n numbers. In retrospect, it is easy to see why guess-and-verify failed: this formula is fairly complicated.

As a check, we can confirm that this formula gives the same values that we computed earlier:

n	T_n	$n \log n - n + 1$
1	0	$1 \log 1 - 1 + 1 = 0$
2	1	$2 \log 2 - 2 + 1 = 1$
4	5	$4 \log 4 - 4 + 1 = 5$
8	17	$8 \log 8 - 8 + 1 = 17$
16	49	$16 \log 16 - 16 + 1 = 49$

As a double-check, we could write out an explicit induction proof. This would be straightforward, because we already worked out the guts of the proof in step 2 of the plug-and-chug procedure.

4.4 Linear Recurrences

So far we've solved recurrences with two techniques: guess-and-verify and plug-and-chug. These methods require spotting a pattern in a sequence of numbers or expressions. In this section and the next, we'll give cookbook solutions for two large classes of recurrences. These methods require no flash of insight; you just follow the recipe and get the answer.

4.4.1 Solving the Fibonacci Recurrence

We ended the episode on the Fibonacci story earlier in Section 2.2 at a sort of a cliffhanger without revealing the answer to a key question: "Does it have a closed form solution?" That part is settled here with a rather pleasant ending.

The Fibonacci recurrence belongs to the class of linear recurrences, which are essentially all solvable with a technique that you can learn in an hour. This is somewhat amazing, since the Fibonacci recurrence remained unsolved for almost six centuries!

In general, a *homogeneous linear recurrence* has the form

$$f(n) = a_1 f(n-1) + a_2 f(n-2) + \cdots + a_d f(n-d)$$

where $a_1, a_2, \ldots, a_d$ and d are constants.

The *order* of the recurrence is d. Commonly, the value of the function f is also specified at a few points; these are called *boundary conditions*.
For example, the Fibonacci recurrence has order $d = 2$ with coefficients $a_1 = a_2 = 1$ and $g(n) = 0$. The boundary conditions are $f(0) = 1$ and $f(1) = 1$.

Here we've switched from subscript notation to functional notation, from T_n (and F_n) to $f(n)$.
The change is cosmetic, but the expressiveness of functions will be useful later.

The word "homogeneous"sounds scary, but effectively means "the simpler kind." We'll consider linear recurrences with a more complicated form later.

Let's try to solve the Fibonacci recurrence with the benefit centuries of hindsight. In general, linear recurrences tend to have exponential solutions. So let's guess that

$$f(n) = x^n$$

where x is a parameter introduced to improve our odds of making a correct guess. We'll figure out the best value for x later. To further improve our odds, let's neglect the boundary conditions $f(0) = 0$ and $f(1) = 1$ for now. Plugging this guess into the recurrence $f(n) = f(n-1) + f(n-2)$ gives

$$x^n = x^{n-1} + x^{n-2}.$$

Dividing both sides by x^{n-2} leaves a quadratic equation:

$$x^2 = x + 1.$$

Solving this equation gives *two* plausible values for the parameter x:

$$x = \frac{1 \pm \sqrt{5}}{2}.$$

This suggests that there are at least two different solutions to the recurrence, neglecting the boundary conditions.

$$f(n) = \left(\frac{1+\sqrt{5}}{2}\right)^n \quad \text{or} \quad f(n) = \left(\frac{1-\sqrt{5}}{2}\right)^n$$

A charming features of homogeneous linear recurrences is that any linear combination of solutions is another solution.

Theorem 44. *If $f(n)$ and $g(n)$ are both solutions to a homogeneous linear recurrence, then $h(n) = sf(n) + tg(n)$ is also a solution for all $s, t \in \mathbb{R}$.*

Proof.

$$\begin{aligned} h(n) &= sf(n) + tg(n) \\ &= s\left(a_1 f(n-1) + \cdots + a_d f(n-d)\right) + t\left(a_1 g(n-1) + \cdots + a_d g(n-d)\right) \end{aligned}$$

$$\begin{aligned}&= a_1(sf(n-1)+tg(n-1)) + \cdots + a_d(sf(n-d)+tg(n-d))\\&= a_1h(n-1)+\cdots+a_dh(n-d)\end{aligned}$$

The first step uses the definition of the function h, and the second uses the fact that f and g are solutions to the recurrence. In the last two steps, we rearrange terms and use the definition of h again. Since the first expression is equal to the last, h is also a solution to the recurrence. □

The phenomenon described in this theorem—a linear combination of solutions is another solution—also holds for many differential equations and physical systems. In fact, linear recurrences are so similar to linear differential equations that you can safely snooze through that topic in some future math class.

Returning to the Fibonacci recurrence, this theorem implies that

$$f(n) = s\left(\frac{1+\sqrt{5}}{2}\right)^n + t\left(\frac{1-\sqrt{5}}{2}\right)^n$$

is a solution for all real numbers s and t. The theorem expanded two solutions to a whole spectrum of possibilities! Now, given all these options to choose from, we can find one solution that satisfies the boundary conditions, $f(0)=1$ and $f(1)=1$. Each boundary condition puts some constraints on the parameters s and t. In particular, the first boundary condition implies that

$$f(0) = s\left(\frac{1+\sqrt{5}}{2}\right)^0 + t\left(\frac{1-\sqrt{5}}{2}\right)^0 = s+t = 1.$$

Similarly, the second boundary condition implies that

$$f(1) = s\left(\frac{1+\sqrt{5}}{2}\right)^1 + t\left(\frac{1-\sqrt{5}}{2}\right)^1 = 1.$$

Now we have two linear equations in two unknowns. The system is not degenerate, so there is a unique solution:

$$s = \frac{1}{\sqrt{5}}\cdot\frac{1+\sqrt{5}}{2} \qquad t = -\frac{1}{\sqrt{5}}\cdot\frac{1-\sqrt{5}}{2}.$$

These values of s and t identify a solution to the Fibonacci recurrence that also satisfies the boundary conditions:

$$\begin{aligned}f(n) &= \frac{1}{\sqrt{5}}\cdot\frac{1+\sqrt{5}}{2}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\cdot\frac{1-\sqrt{5}}{2}\left(\frac{1-\sqrt{5}}{2}\right)^n\\&= \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n+1}.\end{aligned}$$

It is easy to see why no one stumbled across this solution for almost six centuries. All Fibonacci numbers are integers, but this expression is full of square roots of five! Amazingly, the square roots always cancel out. This expression really does give the Fibonacci numbers if we plug in $n = 0, 1, 2$, etc.

This closed form for Fibonacci numbers is known as *Binet's formula* and has some interesting corollaries. The first term tends to infinity because the base of the exponential, $(1+\sqrt{5})/2 = 1.618\ldots$ is greater than one. This value is often denoted ϕ and called the "golden ratio." The second term tends to zero, because $(1-\sqrt{5})/2 = -0.618033988\ldots$ has absolute value less than 1. This implies that the nth Fibonacci number is:

$$f(n) = \frac{\phi^{n+1}}{\sqrt{5}} + o(1).$$

Remarkably, this expression involving irrational numbers is actually very close to an integer for all large n—namely, a Fibonacci number! For example:

$$\frac{\phi^{20}}{\sqrt{5}} = 6765.000029\cdots \approx f(19).$$

This also implies that the ratio of consecutive Fibonacci numbers rapidly approaches the golden ratio. For example:

$$\frac{f(20)}{f(19)} = \frac{10946}{6765} = 1.618033998\ldots.$$

4.4.2 Solving Homogeneous Linear Recurrences

The method we used to solve the Fibonacci recurrence can be extended to solve any homogeneous linear recurrence; that is, a recurrence of the form

$$f(n) = a_1 f(n-1) + a_2 f(n-2) + \cdots + a_d f(n-d)$$

where $a_1, a_2, \ldots, a_d$ and d are constants. Substituting the guess $f(n) = x^n$, as with the Fibonacci recurrence, gives

$$x^n = a_1 x^{n-1} + a_2 x^{n-2} + \cdots + a_d x^{n-d}.$$

Dividing by x^{n-d} gives

$$x^d = a_1 x^{d-1} + a_2 x^{d-2} + \cdots + a_{d-1} x + a_d.$$

This is called the *characteristic equation* of the recurrence. The characteristic equation can be read off quickly since the coefficients of the equation are the same as the coefficients of the recurrence.

The solutions to a linear recurrence are defined by the roots of the characteristic equation. Neglecting boundary conditions for the moment:

- If r is a nonrepeated root of the characteristic equation, then r^n is a solution to the recurrence.
- If r is a repeated root with multiplicity k then r^n, nr^n, $n^2 r^n$, ..., $n^{k-1} r^n$ are all solutions to the recurrence.

Theorem 44 implies that every linear combination of these solutions is also a solution.

For example, suppose that the characteristic equation of a recurrence has roots s, t and u twice. These four roots imply four distinct solutions:

$$f(n) = s^n \qquad f(n) = t^n \qquad f(n) = u^n \qquad f(n) = nu^n.$$

Furthermore, every linear combination

$$f(n) = a \cdot s^n + b \cdot t^n + c \cdot u^n + d \cdot nu^n \tag{4.6}$$

is also a solution.

All that remains is to select a solution consistent with the boundary conditions by choosing the constants appropriately. Each boundary condition implies a linear equation involving these constants. So we can determine the constants by solving a system of linear equations. For example, suppose our boundary conditions were $f(0) = 0$, $f(1) = 1$, $f(2) = 4$ and $f(3) = 9$. Then we would obtain four equations in four unknowns:

$$\begin{array}{lll} f(0) = 0 & \text{implies} & a \cdot s^0 + b \cdot t^0 + c \cdot u^0 + d \cdot 0u^0 = 0 \\ f(1) = 1 & \text{implies} & a \cdot s^1 + b \cdot t^1 + c \cdot u^1 + d \cdot 1u^1 = 1 \\ f(2) = 4 & \text{implies} & a \cdot s^2 + b \cdot t^2 + c \cdot u^2 + d \cdot 2u^2 = 4 \\ f(3) = 9 & \text{implies} & a \cdot s^3 + b \cdot t^3 + c \cdot u^3 + d \cdot 3u^3 = 9 \end{array}$$

This looks nasty, but remember that s, t and u are just constants. Solving this system gives values for a, b, c and d that define a solution to the recurrence consistent with the boundary conditions.

4.4.3 Solving General Linear Recurrences

We can now solve all linear homogeneous recurrences, which have the form

$$f(n) = a_1 f(n-1) + a_2 f(n-2) + \cdots + a_d f(n-d).$$

Many recurrences that arise in practice do not quite fit this mold. For example, the Towers of Hanoi problem led to this recurrence:

$$\begin{aligned} f(1) &= 1 \\ f(n) &= 2f(n-1) + 1 \qquad &\text{(for } n \geq 2\text{)}. \end{aligned}$$

The problem is the extra $+1$; that is not allowed in a homogeneous linear recurrence. In general, adding an extra function $g(n)$ to the right side of a linear recurrence gives an *inhomogeneous linear recurrence*:

$$f(n) = a_1 f(n-1) + a_2 f(n-2) + \cdots + a_d f(n-d) + g(n).$$

Solving inhomogeneous linear recurrences is neither very different nor very difficult. We can divide the whole job into five steps:

1. Replace $g(n)$ by 0, leaving a homogeneous recurrence. As before, find roots of the characteristic equation.

2. Write down the solution to the homogeneous recurrence, but do not yet use the boundary conditions to determine coefficients. This is called the *homogeneous solution.*

3. Now restore $g(n)$ and find a single solution to the recurrence, ignoring boundary conditions. This is called a *particular solution.* We'll explain how to find a particular solution shortly.

4. Add the homogeneous and particular solutions together to obtain the *general solution.*

5. Now use the boundary conditions to determine constants by the usual method of generating and solving a system of linear equations.

As an example, let's consider a variation of the Towers of Hanoi problem. Suppose that moving a disk takes time proportional to its size. Specifically, moving the smallest disk takes 1 second, the next-smallest takes 2 seconds, and moving the nth disk then requires n seconds instead of 1. So, in this variation, the time to complete the job is given by a recurrence with a $+n$ term instead of a $+1$:

$$\begin{aligned} f(1) &= 1 \\ f(n) &= 2f(n-1) + n \qquad &\text{for } n \geq 2. \end{aligned}$$

Clearly, this will take longer, but how much longer? Let's solve the recurrence with the method described above.

In Steps 1 and 2, dropping the $+n$ leaves the homogeneous recurrence $f(n) = 2f(n-1)$. The characteristic equation is $x = 2$. So the homogeneous solution is $f(n) = c2^n$.

In Step 3, we must find a solution to the full recurrence $f(n) = 2f(n-1) + n$, without regard to the boundary condition. Let's guess that there is a solution of the form $f(n) = an + b$ for some constants a and b. Substituting this guess into the recurrence gives

$$\begin{aligned} an + b &= 2(a(n-1) + b) + n \\ 0 &= (a+1)n + (b - 2a). \end{aligned}$$

The second equation is a simplification of the first. The second equation holds for all n if both $a+1 = 0$ (which implies $a = -1$) and $b - 2a = 0$ (which implies that $b = -2$). So $f(n) = an + b = -n - 2$ is a particular solution.

In the Step 4, we add the homogeneous and particular solutions to obtain the general solution

$$f(n) = c2^n - n - 2.$$

Finally, in step 5, we use the boundary condition $f(1) = 1$ to determine the value of the constant c:

$$\begin{aligned} f(1) = 1 \quad &\text{IMPLIES} \quad c2^1 - 1 - 2 = 1 \\ &\text{IMPLIES} \quad c = 2. \end{aligned}$$

Therefore, the function $f(n) = 2 \cdot 2^n - n - 2$ solves this variant of the Towers of Hanoi recurrence. For comparison, the solution to the original Towers of Hanoi problem was $2^n - 1$. So if moving disks takes time proportional to their size, then the monks will need about twice as much time to solve the whole puzzle.

4.4.4 How to Guess a Particular Solution

Finding a particular solution can be the hardest part of solving inhomogeneous recurrences. This involves guessing, and you might guess wrong.[2] However, some rules of thumb make this job fairly easy most of the time.

- Generally, look for a particular solution with the same form as the inhomogeneous term $g(n)$.
- If $g(n)$ is a constant, then guess a particular solution $f(n) = c$. If this doesn't work, try polynomials of progressively higher degree: $f(n) = bn + c$, then $f(n) = an^2 + bn + c$, etc.
- More generally, if $g(n)$ is a polynomial, try a polynomial of the same degree, then a polynomial of degree one higher, then two higher, etc. For example, if $g(n) = 6n + 5$, then try $f(n) = bn + c$ and then $f(n) = an^2 + bn + c$.
- If $g(n)$ is an exponential, such as 3^n, then first guess that $f(n) = c3^n$. Failing that, try $f(n) = bn3^n + c3^n$ and then $an^2 3^n + bn3^n + c3^n$, etc.

The entire process is summarized on the following page.

Short Guide to Solving Linear Recurrences

A linear recurrence is an equation

$$f(n) = \underbrace{a_1 f(n-1) + a_2 f(n-2) + \cdots + a_d f(n-d)}_{\text{homogeneous part}} \quad \underbrace{+\, g(n)}_{\text{inhomogeneous part}}$$

together with boundary conditions such as $f(0) = b_0$, $f(1) = b_1$, etc. Linear recurrences are solved as follows:

1. Find the roots of the characteristic equation

$$x^n = a_1 x^{n-1} + a_2 x^{n-2} + \cdots + a_{k-1} x + a_k.$$

2. Write down the homogeneous solution. Each root generates one term and the homogeneous solution is their sum. A nonrepeated root r generates the term cr^n, where c is a constant to be determined later. A root r with multiplicity k generates the terms

$$d_1 r^n \qquad d_2 n r^n \qquad d_3 n^2 r^n \qquad \ldots \qquad d_k n^{k-1} r^n$$

where $d_1, \ldots d_k$ are constants to be determined later.

[2] The chapter on *generating functions* explains how to solve linear recurrences with generating functions—it's a little more complicated, but it does not require guessing.

3. Find a particular solution. This is a solution to the full recurrence that need not be consistent with the boundary conditions. Use guess-and-verify. If $g(n)$ is a constant or a polynomial, try a polynomial of the same degree, then of one higher degree, then two higher. For example, if $g(n) = n$, then try $f(n) = bn + c$ and then $an^2 + bn + c$. If $g(n)$ is an exponential, such as 3^n, then first guess $f(n) = c3^n$. Failing that, try $f(n) = (bn+c)3^n$ and then $(an^2 + bn + c)3^n$, etc.

4. Form the general solution, which is the sum of the homogeneous solution and the particular solution. Here is a typical general solution:

$$f(n) = \underbrace{c2^n + d(-1)^n}_{\text{homogeneous solution}} + \underbrace{3n + 1}_{\text{inhomogeneous solution}}.$$

5. Substitute the boundary conditions into the general solution. Each boundary condition gives a linear equation in the unknown constants. For example, substituting $f(1) = 2$ into the general solution above gives

$$\begin{aligned} 2 &= c \cdot 2^1 + d \cdot (-1)^1 + 3 \cdot 1 + 1 \\ \text{IMPLIES} \quad -2 &= 2c - d. \end{aligned}$$

Determine the values of these constants by solving the resulting system of linear equations.

4.5 Divide-and-Conquer Recurrences

We now have a recipe for solving general linear recurrences. But the Merge Sort recurrence, which we encountered earlier, is not linear:

$$\begin{aligned} T(1) &= 0 \\ T(n) &= 2T(n/2) + n - 1 \qquad (\text{for } n \geq 2). \end{aligned}$$

In particular, $T(n)$ is not a linear combination of a fixed number of immediately preceding terms; rather, $T(n)$ is a function of $T(n/2)$, a term halfway back in the sequence.

Merge Sort is an example of a divide-and-conquer algorithm: it divides the input, "conquers" the pieces, and combines the results. Analysis of such algorithms commonly leads to *divide-and-conquer* recurrences, which have this form:

$$T(n) = \sum_{i=1}^{k} a_i T(b_i n) + g(n)$$

Here $a_1, \ldots a_k$ are positive constants, $b_1, \ldots, b_k$ are constants between 0 and 1, and $g(n)$ is a nonnegative function. For example, setting $a_1 = 2$, $b_1 = 1/2$ and $g(n) = n - 1$ gives the Merge Sort recurrence.

4.5.1 The Akra-Bazzi Formula

The solution to virtually all divide and conquer solutions is given by the amazing *Akra-Bazzi formula.* Quite simply, the asymptotic solution to the general divide-and-conquer recurrence

$$T(n) = \sum_{i=1}^{k} a_i T(b_i n) + g(n)$$

is

$$T(n) = \Theta\left(n^p \left(1 + \int_1^n \frac{g(u)}{u^{p+1}}\, du\right)\right) \tag{4.7}$$

where p satisfies

$$\sum_{i=1}^{k} a_i b_i^p = 1. \tag{4.8}$$

A rarely-troublesome requirement is that the function $g(n)$ must not grow or oscillate too quickly. Specifically, $|g'(n)|$ must be bounded by some polynomial. So, for example, the Akra-Bazzi formula is valid when $g(n) = x^2 \log n$, but not when $g(n) = 2^n$.

Let's solve the Merge Sort recurrence again, using the Akra-Bazzi formula instead of plug-and-chug. First, we find the value p that satisfies

$$2 \cdot (1/2)^p = 1.$$

Looks like $p = 1$ does the job. Then we compute the integral:

$$\begin{aligned}
T(n) &= \Theta\left(n\left(1 + \int_1^n \frac{u-1}{u^2}\,du\right)\right) \\
&= \Theta\left(n\left(1 + \left[\log u + \frac{1}{u}\right]_1^n\right)\right) \\
&= \Theta\left(n\left(\log n + \frac{1}{n}\right)\right) \\
&= \Theta(n \log n).
\end{aligned}$$

The first step is integration and the second is simplification. We can drop the $1/n$ term in the last step, because the $\log n$ term dominates. We're done!

Let's try a scary-looking recurrence:

$$T(n) = 2T(n/2) + (8/9)T(3n/4) + n^2.$$

Here, $a_1 = 2$, $b_1 = 1/2$, $a_2 = 8/9$ and $b_2 = 3/4$. So we find the value p that satisfies

$$2 \cdot (1/2)^p + (8/9)(3/4)^p = 1.$$

Equations of this form don't always have closed-form solutions, so you may need to approximate p numerically sometimes. But in this case the solution is simple: $p = 2$. Then we integrate:

$$\begin{aligned}
T(n) &= \Theta\left(n^2\left(1 + \int_1^n \frac{u^2}{u^3}\,du\right)\right) \\
&= \Theta\left(n^2(1 + \log n)\right) \\
&= \Theta\left(n^2 \log n\right).
\end{aligned}$$

That was easy!

4.5.2 Two Technical Issues

Until now, we've swept a couple issues related to divide-and-conquer recurrences under the rug. Let's address those issues now.

First, the Akra-Bazzi formula makes no use of boundary conditions. To see why, let's go back to Merge Sort. During the plug-and-chug analysis, we found that

$$T_n = nT_1 + n \log n - n + 1.$$

This expresses the nth term as a function of the first term, whose value is specified in a boundary condition. But notice that $T_n = \Theta(n \log n)$ for *every* value of T_1. The boundary condition doesn't matter!

This is the typical situation: *the asymptotic solution to a divide-and-conquer recurrence is independent of the boundary conditions.* Intuitively, if the bottom-level operation in a recursive algorithm takes, say, twice as long, then the overall running time will at most double. This matters in practice, but the factor of 2 is concealed by asymptotic notation. There are corner-case exceptions. For example, the solution to $T(n) = 2T(n/2)$ is either $\Theta(n)$ or zero, depending on whether $T(1)$ is zero. These cases are of little practical interest, so we won't consider them further.

There is a second nagging issue with divide-and-conquer recurrences that does not arise with linear recurrences. Specifically, dividing a problem of size n may create subproblems of non-integer size. For example, the Merge Sort recurrence contains the term $T(n/2)$. So what if n is 15? How long does it take to sort seven-and-a-half items? Previously, we dodged this issue by analyzing Merge Sort only when the size of the input was a power of 2. But then we don't know what happens for an input of size, say, 100.

Of course, a practical implementation of Merge Sort would split the input *approximately* in half, sort the halves recursively, and merge the results. For example, a list of 15 numbers would be split into lists of 7 and 8. More generally, a list of n numbers would be split into approximate halves of size $\lceil n/2 \rceil$ and $\lceil n/2 \rceil$. So the maximum number of comparisons is actually given by this recurrence:

$$
\begin{aligned}
T(1) &= 0 \\
T(n) &= T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + n - 1 \qquad\qquad \text{(for } n \geq 2\text{).}
\end{aligned}
$$

This may be rigorously correct, but the ceiling and floor operations make the recurrence hard to solve exactly.

Fortunately, *the asymptotic solution to a divide and conquer recurrence is unaffected by floors and ceilings.* More precisely, the solution is not changed by replacing a term $T(b_i n)$ with either $T(\lceil b_i n \rceil)$ or $T(\lfloor b_i n \rfloor)$. So leaving floors and ceilings out of divide-and-conquer recurrences makes sense in many contexts; those are complications that make no difference.

4.5.3 The Akra-Bazzi Theorem

The Akra-Bazzi formula together with our assertions about boundary conditions and integrality all follow from the *Akra-Bazzi Theorem*, which is stated below.

Theorem 45 (Akra-Bazzi). *Suppose that the function $T : \mathbb{R} \to \mathbb{R}$ is nonnegative and bounded for $0 \leq x \leq x_0$ and satisfies the recurrence*

$$
T(x) = \sum_{i=1}^{k} a_i T(b_i x + h_i(x)) + g(x) \qquad \textit{for } x > x_0, \tag{4.9}
$$

where:

1. *x_0 is large enough so that T is well-defined,*

2. *$a_1, \ldots, a_k$ are positive constants,*

3. *$b_1, \ldots, b_k$ are constants between 0 and 1,*

4. *$g(x)$ is a nonnegative function such that $|g'(x)|$ is bounded by a polynomial,*

5. *$|h_i(x)| = O(x/\log^2 x)$.*

Then

$$
T(x) = \Theta\left(x^p \left(1 + \int_1^x \frac{g(u)}{u^{p+1}}\, du\right)\right)
$$

where p satisfies

$$
\sum_{i=1}^{k} a_i b_i^p = 1.
$$

The Akra-Bazzi theorem can be proved using a complicated induction argument, though we won't do that here. But let's at least go over the statement of the theorem.

All the recurrences we've considered were defined over the integers, and that is the common case. But the Akra-Bazzi theorem applies more generally to functions defined over the real numbers.

The Akra-Bazzi formula is lifted directed from the theorem statement, except that the recurrence in the theorem includes extra functions, h_i. These functions extend the theorem to address floors, ceilings, and other small adjustments to the sizes of subproblems. The trick is illustrated by this combination of parameters

$$
\begin{aligned}
a_1 &= 1 & b_1 &= 1/2 & h_1(x) &= \left\lceil \frac{x}{2} \right\rceil - \frac{x}{2} \\
a_2 &= 1 & b_2 &= 1/2 & h_2(x) &= \left\lfloor \frac{x}{2} \right\rfloor - \frac{x}{2}
\end{aligned}
$$
$$g(x) = x - 1$$

which corresponds the recurrence

$$
\begin{aligned}
T(x) &= 1 \cdot T\left(\frac{x}{2} + \left(\left\lceil \frac{x}{2} \right\rceil - \frac{x}{2}\right)\right) + \cdot T\left(\frac{x}{2} + \left(\left\lfloor \frac{x}{2} \right\rfloor - \frac{x}{2}\right)\right) + x - 1 \\
&= T\left(\left\lceil \frac{x}{2} \right\rceil\right) + T\left(\left\lfloor \frac{x}{2} \right\rfloor\right) + x - 1.
\end{aligned}
$$

This is the rigorously correct Merge Sort recurrence valid for all input sizes, complete with floor and ceiling operators. In this case, the functions $h_1(x)$ and $h_2(x)$ are both at most 1, which is easily $O(x/\log^2 x)$ as required by the theorem statement. These functions h_i do not affect—or even appear in—the asymptotic solution to the recurrence. This justifies our earlier claim that applying floor and ceiling operators to the size of a subproblem does not alter the asymptotic solution to a divide-and-conquer recurrence.

4.5.4 The Master Theorem

There is a special case of the Akra-Bazzi formula known as the Master Theorem that handles some of the recurrences that commonly arise in computer science. It is called the *Master* Theorem because it was proved long before Akra and Bazzi arrived on the scene and, for many years, it was the final word on solving divide-and-conquer recurrences. We include the Master Theorem here because it is still widely referenced in algorithms courses and you can use it without having to know anything about integration.

Theorem 46 (Master Theorem)**.** *Let T be a recurrence of the form*

$$T(n) = aT\left(\frac{n}{b}\right) + g(n).$$

Case 1: *If $g(n) = O\left(n^{\log_b(a)-\epsilon}\right)$ for some constant $\epsilon > 0$, then*

$$T(n) = \Theta\left(n^{\log_b(a)}\right).$$

Case 2: *If $g(n) = \Theta\left(n^{\log_b(a)} \log^k(n)\right)$ for some constant $k \geq 0$, then*

$$T(n) = \Theta\left(n^{\log_b(a)} \log^{k+1}(n)\right).$$

Case 3: *If $g(n) = \Omega\left(n^{\log_b(a)+\epsilon}\right)$ for some constant $\epsilon > 0$ and $ag(n/b) < cg(n)$ for some constant $c < 1$ and sufficiently large n, then*

$$T(n) = \Theta(g(n)).$$

The Master Theorem can be proved by induction on n or, more easily, as a corollary of Theorem 45. We will not include the details here.

Practice Problems

Exercise 4.5.1 The running time of an algorithm A is described by the recurrence $T(n) = 7T(n/2) + n^2$. A competing algorithm A' has a running time of $T'(n) = aT'(n/4) + n^2$. For what values of a is A' asymptotically faster than A?

Exercise 4.5.2 Use the Akra-Bazzi formula to find $\Theta()$ asymptotic bounds for the following divide-and-conquer recurrences. For each recurrence, $T(1) = 1$ and $T(n) = \Theta(1)$ for all constant n. State the value of p you get for each recurrence (which can be left in the form of logs). Also, state the values of the $a_i, b_i,$ and $h_i(n)$ for each recurrence.

1. $T(n) = 3T(\lfloor n/3 \rfloor) + n$.
2. $T(n) = 4T(\lfloor n/3 \rfloor) + n^2$.
3. $T(n) = 3T(\lfloor n/4 \rfloor) + n$.
4. $T(n) = T(\lfloor n/4 \rfloor) + T(\lfloor n/3 \rfloor) + n$.
5. $T(n) = T(\lceil n/4 \rceil) + T(\lfloor 3n/4 \rfloor) + n$.
6. $T(n) = 2T(\lfloor n/4 \rfloor) + \sqrt{n}$.
7. $T(n) = 2T(\lfloor n/4 \rfloor + 1) + \sqrt{n}$.
8. $T(n) = 2T(\lfloor n/4 + \sqrt{n} \rfloor) + 1$.
9. $T(n) = 3T(\lceil n^{1/3} \rceil) + \log_3 n$. (For this problem, $T(2) = 1$.)
10. $T(n) = \sqrt{e}T(\lfloor n^{1/e} \rfloor) + \ln n$.

Exercise 4.5.3 We have devised an error-tolerant version of **MergeSort**. We call our exciting new algorithm **OverSort**.

Here is how the new algorithm works. The input is a list of n distinct numbers. If the list contains a single number, then there is nothing to do. If the list contains two numbers, then we sort them with a single comparison. If the list contains more than two numbers, then we perform the following sequence of steps.

- We make a list containing the first $\frac{2}{3}n$ numbers and sort it recursively.
- We make a list containing the last $\frac{2}{3}n$ numbers and sort it recursively.
- We make a list containing the first $\frac{1}{3}n$ numbers and the last $\frac{1}{3}n$ numbers and sort it recursively.
- We merge the first and second lists, throwing out duplicates.
- We merge this combined list with the third list, again throwing out duplicates.

The final, merged list is the output. What's great is that because multiple copies of each number are maintained, even if the sorter occasionally forgets about a number, **OverSort** can still output a complete, sorted list.

(a) Let $T(n)$ be the maximum number of comparisons that **OverSort** could use to sort a list of n distinct numbers, assuming the sorter never forgets a number and n is a power of 3. What is $T(3)$? Write a recurrence relation for $T(n)$. (*Hint:* Merging a list of j distinct numbers and a list of k distinct numbers, and throwing out duplicates of numbers that appear in both lists, requires $j + k - d$ comparisons, when $d > 0$ is the number of duplicates.)

(b) Now we're going to apply the *Akra-Bazzi Theorem* to find a Θ bound on $T(n)$. Begin by identifying the following constants and functions in the Akra-Bazzi recurrence (4.9):

- The constant k.
- The constants a_i.
- The constants b_i.
- The functions h_i.
- The function g.
- The constant p. You can leave p in terms of logarithms, but you'll need a rough estimate of its value later on.

(c) Does the condition $|g'(x)| = O(x^c)$ for some $c \in \mathbb{N}$ hold?

(d) Does the condition $|h_i(x)| = O(x/\log^2 x)$ hold?

(e) Determine a Θ bound on $T(n)$ by integration.

Exercise 4.5.4 Use the Akra-Bazzi formula to find $\Theta()$ asymptotic bounds for the following recurrences. For each recurrence $T(0) = 1$ and $n \in \mathbb{N}$.

(a) $T(n) = 2T(\lfloor n/4 \rfloor) + T(\lfloor n/3 \rfloor) + n$

(b) $T(n) = 4T(\lfloor n/2 + \sqrt{n} \rfloor) + n^2$

(c) A society of devil-worshipers meets every week in a catacomb to initiate new members. Members who have been in the society for two or more weeks initiate four new members each and members who have been in the society for only one week initiate one new member each. On week 0 there is one devil-worshiper. There are two devil-worshipers on week 1.

Write a recurrence relation for the number of members $D(n)$ in the society on the nth week.

4.6 A Feel for Recurrences

We've guessed and verified, plugged and chugged, found roots, computed integrals, and solved linear systems and exponential equations. Now let's step back and look for some rules of thumb. What kinds of recurrences have what sorts of solutions?

Here are some recurrences we solved earlier:

	Recurrence	Solution
Towers of Hanoi	$T_n = 2T_{n-1} + 1$	$T_n \sim 2^n$
Merge Sort	$T_n = 2T_{n/2} + n - 1$	$T_n \sim n \log n$
Hanoi variation	$T_n = 2T_{n-1} + n$	$T_n \sim 2 \cdot 2^n$
Fibonacci	$T_n = T_{n-1} + T_{n-2}$	$T_n \sim (1.618\ldots)^{n+1}/\sqrt{5}$

Notice that the recurrence equations for Towers of Hanoi and Merge Sort are somewhat similar, but the solutions are radically different. Merge Sorting $n = 64$ items takes a few hundred comparisons, while moving $n = 64$ disks takes more than 10^{19} steps!

Each recurrence has one strength and one weakness. In the Towers of Hanoi, we broke a problem of size n into two subproblem of size $n - 1$ (which is large), but needed only 1 additional step (which is small). In Merge Sort, we divided the problem of size n into two subproblems of size $n/2$ (which is small), but needed $(n - 1)$ additional steps (which is large). Yet, Merge Sort is faster by a mile!

This suggests that *generating smaller subproblems is far more important to algorithmic speed than reducing the additional steps per recursive call.* For example, shifting to the variation of Towers of Hanoi increased the last term from $+1$ to $+n$, but the solution only doubled. And one of the two subproblems in the Fibonacci recurrence is just *slightly* smaller than in Towers of Hanoi (size $n-2$ instead of $n-1$). Yet the solution is exponentially smaller! More generally, linear recurrences (which have big subproblems) typically have exponential solutions, while divide-and-conquer recurrences (which have small subproblems) usually have solutions bounded above by a polynomial.

All the examples listed above break a problem of size n into two smaller problems. How does the number of subproblems affect the solution? For example, suppose we increased the number of subproblems in Towers of Hanoi from 2 to 3, giving this recurrence:

$$T_n = 3T_{n-1} + 1$$

This increases the root of the characteristic equation from 2 to 3, which raises the solution exponentially, from $\Theta(2^n)$ to $\Theta(3^n)$.

Divide-and-conquer recurrences are also sensitive to the number of subproblems. For example, for this generalization of the Merge Sort recurrence:

$$T_1 = 0$$

$$T_n = aT_{n/2} + n - 1.$$

the Akra-Bazzi formula gives:

$$T_n = \begin{cases} \Theta(n) & \text{for } a < 2 \\ \Theta(n \log n) & \text{for } a = 2 \\ \Theta(n^{\log a}) & \text{for } a > 2. \end{cases}$$

So the solution takes on three completely different forms as a goes from 1.99 to 2.01!

How do boundary conditions affect the solution to a recurrence? We've seen that they are almost irrelevant for divide-and-conquer recurrences. For linear recurrences, the solution is usually dominated by an exponential whose base is determined by the number and size of subproblems. Boundary conditions matter greatly only when they give the dominant term a zero coefficient, which changes the asymptotic solution.

So now we have a rule of thumb! The performance of a recursive procedure is usually dictated by the size and number of subproblems, rather than the amount of work per recursive call or time spent at the base of the recursion. In particular, if subproblems are smaller than the original by an additive factor, the solution is most often exponential. But if the subproblems are only a fraction the size of the original, then the solution is typically bounded by a polynomial.

5 | Generating Functions

Generating Functions are one of the most surprising and useful inventions in Discrete Math. Roughly speaking, generating functions transform problems about *sequences* into problems about *functions*. This is great because we've got piles of mathematical machinery for manipulating functions. Thanks to generating functions, we can then apply all that machinery to problems about sequences. In this way, we can use generating functions to solve all sorts of counting problems. They can also be used to find closed-form expressions for sums and to solve recurrences. In fact, many of the problems we addressed while studying *sums and asymptotics* and *counting* can be formulated and solved using generating functions.

5.1 Definitions and Examples

The *ordinary generating function* for the sequence[1] $\langle g_0, g_1, g_2, g_3 \ldots \rangle$ is the power series:

$$G(x) = g_0 + g_1 x + g_2 x^2 + g_3 x^3 + \cdots .$$

There are a few other kinds of generating functions in common use, but ordinary generating functions are enough to illustrate the power of the idea, so we'll stick to them and from now on, *generating function* will mean the ordinary kind.

A generating function is a "formal" power series in the sense that we usually regard x as a place-holder rather than a number. Only in rare cases will we actually evaluate a generating function by letting x take a real number value, so we generally ignore the issue of *convergence*.

Throughout this chapter, we'll indicate the correspondence between a sequence and its generating function with a double-sided arrow as follows:

$$\langle g_0, g_1, g_2, g_3, \ldots \rangle \longleftrightarrow g_0 + g_1 x + g_2 x^2 + g_3 x^3 + \cdots .$$

For example, here are some sequences and their generating functions:

$$\begin{aligned}
\langle 0, 0, 0, 0, \ldots \rangle &\longleftrightarrow 0 + 0x + 0x^2 + 0x^3 + \cdots = 0 \\
\langle 1, 0, 0, 0, \ldots \rangle &\longleftrightarrow 1 + 0x + 0x^2 + 0x^3 + \cdots = 1 \\
\langle 3, 2, 1, 0, \ldots \rangle &\longleftrightarrow 3 + 2x + 1x^2 + 0x^3 + \cdots = 3 + 2x + x^2
\end{aligned}$$

The pattern here is simple: the ith term in the sequence (indexing from 0) is the coefficient of x^i in the generating function.

Recall that the sum of an infinite *geometric series* is:

$$1 + z + z^2 + z^3 + \cdots = \frac{1}{1-z}.$$

This equation does not hold when $|z| \geq 1$, but as remarked, we won't worry about convergence issues for now. This formula gives *closed form* generating functions for a whole range of sequences. For example:

$$\langle 1, 1, 1, 1, \ldots \rangle \longleftrightarrow 1 + x + x^2 + x^3 + x^4 + \cdots \quad = \frac{1}{1-x}$$

[1] In this chapter, we'll put sequences in angle brackets to more clearly distinguish them from the many other mathematical expressions floating around.

$$\begin{aligned}
\langle 1,-1,1,-1,\ldots\rangle &\longleftrightarrow 1-x+x^2-x^3+x^4-\cdots &&= \frac{1}{1+x}\\
\left\langle 1,a,a^2,a^3,\ldots\right\rangle &\longleftrightarrow 1+ax+a^2x^2+a^3x^3+\cdots &&= \frac{1}{1-ax}\\
\langle 1,0,1,0,1,0,\ldots\rangle &\longleftrightarrow 1+x^2+x^4+x^6+x^8+\cdots &&= \frac{1}{1-x^2}
\end{aligned}$$

5.2 Operations on Generating Functions

The magic of generating functions is that we can carry out all sorts of manipulations on sequences by performing mathematical operations on their associated generating functions. Let's experiment with various operations and characterize their effects in terms of sequences.

5.2.1 Scaling

Multiplying a generating function by a constant scales every term in the associated sequence by the same constant. For example, we noted above that:

$$\langle 1,0,1,0,1,0,\ldots\rangle \longleftrightarrow 1+x^2+x^4+x^6+\cdots = \frac{1}{1-x^2}.$$

Multiplying the generating function by 2 gives

$$\frac{2}{1-x^2} = 2+2x^2+2x^4+2x^6+\cdots$$

which generates the sequence:

$$\langle 2,0,2,0,2,0,\ldots\rangle.$$

Rule 47 (*Scaling Rule*)**.** *If*

$$\langle f_0,f_1,f_2,\ldots\rangle \longleftrightarrow F(x),$$

then

$$\langle cf_0,\ cf_1,\ cf_2,\ \ldots\rangle \longleftrightarrow c\cdot F(x).$$

The idea behind this rule is that:

$$\begin{aligned}
\langle cf_0,cf_1,cf_2,\ldots\rangle &\longleftrightarrow cf_0+cf_1x+cf_2x^2+\cdots\\
&= c\cdot(f_0+f_1x+f_2x^2+\cdots)\\
&= cF(x).
\end{aligned}$$

5.2.2 Addition

Adding generating functions corresponds to adding the two sequences term by term. For example, adding two of our earlier examples gives:

$$\begin{array}{rcl}
\langle 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ \ldots\rangle & \longleftrightarrow & \dfrac{1}{1-x}\\
+\ \langle 1,-1,1,-1,1,-1,\ldots\rangle & \longleftrightarrow & \dfrac{1}{1+x}\\
\hline
\langle 2,\ 0,\ 2,\ 0,\ 2,\ 0,\ \ldots\rangle & \longleftrightarrow & \dfrac{1}{1-x}+\dfrac{1}{1+x}
\end{array}$$

We've now derived two different expressions that both generate the sequence $\langle 2,0,2,0,\ldots\rangle$. They are, of course, equal:

$$\frac{1}{1-x}+\frac{1}{1+x} = \frac{(1+x)+(1-x)}{(1-x)(1+x)} = \frac{2}{1-x^2}.$$

Rule 48 (Addition Rule)**.** *If*

$$\begin{aligned}\langle f_0, f_1, f_2, \ldots \rangle &\longleftrightarrow F(x) \qquad \text{and} \\ \langle g_0, g_1, g_2, \ldots \rangle &\longleftrightarrow G(x),\end{aligned}$$

then

$$\langle f_0 + g_0,\ f_1 + g_1,\ f_2 + g_2,\ \ldots \rangle \longleftrightarrow F(x) + G(x).$$

The idea behind this rule is that:

$$\begin{aligned}\langle f_0 + g_0,\ f_1 + g_1,\ f_2 + g_2,\ \ldots \rangle \longleftrightarrow &\sum_{n=0}^{\infty} (f_n + g_n) x^n \\ &= \left(\sum_{n=0}^{\infty} f_n x^n\right) + \left(\sum_{n=0}^{\infty} g_n x^n\right) \\ &= F(x) + G(x).\end{aligned}$$

5.2.3 Right Shifting

Let's start over again with a simple sequence and its generating function:

$$\langle 1, 1, 1, 1, \ldots \rangle \longleftrightarrow \frac{1}{1-x}.$$

Now let's *right-shift* the sequence by adding k leading zeros:

$$\begin{aligned}\langle \overbrace{0, 0, \ldots, 0}^{k \text{ zeroes}}, 1, 1, 1, \ldots \rangle \longleftrightarrow &\ x^k + x^{k+1} + x^{k+2} + x^{k+3} + \cdots \\ &= x^k \cdot (1 + x + x^2 + x^3 + \cdots) \\ &= \frac{x^k}{1-x}.\end{aligned}$$

Evidently, adding k leading zeros to the sequence corresponds to multiplying the generating function by x^k. This holds true in general.

Rule 49 (*Right-Shift Rule*)**.** *If* $\langle f_0, f_1, f_2, \ldots \rangle \longleftrightarrow F(x)$, *then:*

$$\langle \overbrace{0, 0, \ldots, 0}^{k \text{ zeroes}}, f_0, f_1, f_2, \ldots \rangle \longleftrightarrow x^k \cdot F(x).$$

The idea behind this rule is that:

$$\begin{aligned}\langle \overbrace{0, 0, \ldots, 0}^{k \text{ zeroes}}, f_0, f_1, f_2, \ldots \rangle \longleftrightarrow &\ f_0 x^k + f_1 x^{k+1} + f_2 x^{k+2} + \cdots \\ &= x^k \cdot (f_0 + f_1 x + f_2 x^2 + f_3 x^3 + \cdots) \\ &= x^k \cdot F(x).\end{aligned}$$

5.2.4 Differentiation

What happens if we take the *derivative* of a generating function? As an example, let's differentiate the now-familiar generating function for an infinite sequence of 1's:

$$\begin{aligned}1 + x + x^2 + x^3 + x^4 + \cdots &= \frac{1}{1-x} \\ \text{IMPLIES} \quad \frac{d}{dx}(1 + x + x^2 + x^3 + x^4 + \cdots) &= \frac{d}{dx}\left(\frac{1}{1-x}\right)\end{aligned}$$

$$\text{IMPLIES} \qquad 1 + 2x + 3x^2 + 4x^3 + \cdots = \frac{1}{(1-x)^2}$$

$$\text{IMPLIES} \qquad \langle 1, 2, 3, 4, \ldots \rangle \longleftrightarrow \frac{1}{(1-x)^2}. \tag{5.1}$$

We found a generating function for the sequence $\langle 1, 2, 3, 4, \ldots \rangle$ of positive integers!

In general, differentiating a generating function has two effects on the corresponding sequence: each term is multiplied by its index and the entire sequence is shifted left one place.

Rule 50 (*Derivative Rule*)**.** *If*

$$\langle f_0, f_1, f_2, f_3, \ldots \rangle \longleftrightarrow F(x),$$

then

$$\langle f_1, 2f_2, 3f_3, \ldots \rangle \longleftrightarrow F'(x).$$

The idea behind this rule is that:

$$\begin{aligned} \langle f_1, 2f_2, 3f_3, \ldots \rangle &\longleftrightarrow f_1 + 2f_2x + 3f_3x^2 + \cdots \\ &= \frac{d}{dx}\left(f_0 + f_1x + f_2x^2 + f_3x^3 + \cdots\right) \\ &= \frac{d}{dx} F(x). \end{aligned}$$

The Derivative Rule is very useful. In fact, there is frequent, independent need for each of differentiation's two effects, multiplying terms by their index and left-shifting one place. Typically, we want just one effect and must somehow cancel out the other. For example, let's try to find the generating function for the sequence of squares, $\langle 0, 1, 4, 9, 16, \ldots \rangle$. If we could start with the sequence $\langle 1, 1, 1, 1, \ldots \rangle$ and multiply each term by its index two times, then we'd have the desired result:

$$\langle 0 \cdot 0,\ 1 \cdot 1,\ 2 \cdot 2,\ 3 \cdot 3,\ \ldots \rangle = \langle 0, 1, 4, 9, \ldots \rangle.$$

A challenge is that differentiation not only multiplies each term by its index, but also shifts the whole sequence left one place. However, the Right-Shift Rule 49 tells how to cancel out this unwanted left-shift: multiply the generating function by x.

Our procedure, therefore, is to begin with the generating function for $\langle 1, 1, 1, 1, \ldots \rangle$, differentiate, multiply by x, and then differentiate and multiply by x once more. Then

$$\begin{array}{rlll} & \langle 1, 1, 1, 1, \ldots \rangle \longleftrightarrow & \dfrac{1}{1-x} & \\ \text{Derivative Rule:} & \langle 1, 2, 3, 4, \ldots \rangle \longleftrightarrow & \dfrac{d}{dx}\dfrac{1}{1-x} & = \dfrac{1}{(1-x)^2} \\ \text{Right-shift Rule:} & \langle 0, 1, 2, 3, \ldots \rangle \longleftrightarrow & x \cdot \dfrac{1}{(1-x)^2} & = \dfrac{x}{(1-x)^2} \\ \text{Derivative Rule:} & \langle 1, 4, 9, 16, \ldots \rangle \longleftrightarrow & \dfrac{d}{dx}\dfrac{x}{(1-x)^2} & = \dfrac{1+x}{(1-x)^3} \\ \text{Right-shift Rule:} & \langle 0, 1, 4, 9, \ldots \rangle \longleftrightarrow & x \cdot \dfrac{1+x}{(1-x)^3} & = \dfrac{x(1+x)}{(1-x)^3} \end{array}$$

Thus, the generating function for squares is:

$$\frac{x(1+x)}{(1-x)^3}. \tag{5.2}$$

5.2.5 Products

Rule 51 (Product Rule)**.** *If*

$$\langle a_0, a_1, a_2, \ldots \rangle \longleftrightarrow A(x), \quad \textit{and} \quad \langle b_0, b_1, b_2, \ldots \rangle \longleftrightarrow B(x),$$

then

$$\langle c_0,\ c_1,\ c_2,\ \ldots \rangle \longleftrightarrow A(x) \cdot B(x),$$

where

$$c_n \triangleq a_0 b_n + a_1 b_{n-1} + a_2 b_{n-2} + \cdots + a_n b_0.$$

To understand this rule, let

$$C(x) \triangleq A(x) \cdot B(x) = \sum_{n=0}^{\infty} c_n x^n.$$

We can evaluate the product $A(x) \cdot B(x)$ by using a table to identify all the cross-terms from the product of the sums:

	b_0x^0	b_1x^1	b_2x^2	b_3x^3	$\ldots$
a_0x^0	$a_0b_0x^0$	$a_0b_1x^1$	$a_0b_2x^2$	$a_0b_3x^3$	$\ldots$
a_1x^1	$a_1b_0x^1$	$a_1b_1x^2$	$a_1b_2x^3$	$\ldots$	
a_2x^2	$a_2b_0x^2$	$a_2b_1x^3$	$\ldots$		
a_3x^3	$a_3b_0x^3$	$\ldots$			
$\vdots$	$\ldots$				

Notice that all terms involving the same power of x lie on a diagonal. Collecting these terms together, we find that the coefficient of x^n in the product is the sum of all the terms on the $(n+1)$st diagonal, namely,

$$a_0 b_n + a_1 b_{n-1} + a_2 b_{n-2} + \cdots + a_n b_0. \tag{5.3}$$

This expression (5.3) may be familiar from a signal processing course; the sequence $\langle c_0, c_1, c_2, \ldots \rangle$ is called the *convolution* of sequences $\langle a_0, a_1, a_2, \ldots \rangle$ and $\langle b_0, b_1, b_2, \ldots \rangle$.

5.3 Evaluating Sums

The product rule looks complicated. But it is surprisingly useful. For example, suppose that we set

$$B(x) = \frac{1}{1-x}.$$

Then $b_i = 1$ for $i \geq 0$ and the nth coefficient of $A(x)B(x)$ is

$$a_0 \cdot 1 + a_1 \cdot 1 + a_2 \cdot 1 + \cdots + a_n \cdot 1 = \sum_{i=0}^{n} a_i.$$

In other words, given any sequence $\langle a_0, a_1, a_2, \ldots \rangle$, we can compute

$$s_n = \sum_{i=0}^{n} a_i$$

for all n by simply multiplying the sequence's generating function by $1/(1-x)$. This is the Summation Rule.

Rule 52 (Summation Rule)**.** *If*

$$\langle a_0, a_1, a_2, \ldots \rangle \longleftrightarrow A(x),$$

then

$$\langle s_0, s_1, s_2, \ldots \rangle \longleftrightarrow \frac{A(x)}{1-x}$$

where

$$s_n = \sum_{i=0}^{n} a_i \quad \textit{for } n \geq 0.$$

The Summation Rule sounds powerful, and it is! We already know that computing sums is often not easy. But multiplying by $1/(1-x)$ is about as easy as it gets.

For example, suppose that we want to compute the sum of the first n squares

$$s_n = \sum_{i=0}^{n} i^2$$

and we forgot the methods on sums we so far learned. All we need to do is compute the generating function for $\langle 0, 1, 4, 9, \ldots \rangle$ and multiply by $1/(1-x)$. We already computed the generating function for $\langle 0, 1, 4, 9, \ldots \rangle$ in Equation 5.2—it is

$$\frac{x(1+x)}{(1-x)^3}.$$

Hence, the generating function for $\langle s_0, s_1, s_2, \ldots \rangle$ is

$$\frac{x(1+x)}{(1-x)^4}.$$

This means that $\sum_{i=0}^{n} i^2$ is the coefficient of x^n in $x(1+x)/(1-x)^4$.

That was pretty easy, but there is one problem—we have no idea how to determine the coefficient of x^n in $x(1+x)/(1-x)^4$! And without that, this whole endeavor (while magical) would be useless. Fortunately, there is a straightforward way to produce the sequence of coefficients from a generating function.

5.4 Extracting Coefficients

5.4.1 Taylor Series

Given a sequence of coefficients $\langle f_0, f_1, f_2, \ldots \rangle$, computing the generating function $F(x)$ is easy since

$$F(x) = f_0 + f_1 x + f_2 x^2 + \cdots.$$

To compute the sequence of coefficients from the generating function, we need to compute the *Taylor Series* for the generating function.

Rule 53 (Taylor Series)**.** *Let $F(x)$ be the generating function for the sequence*

$$\langle f_0, f_1, f_2, \ldots \rangle.$$

Then

$$f_0 = F(0)$$

and

$$f_n = \frac{F^{(n)}(0)}{n!}$$

for $n \geq 1$, where $F^{(n)}(0)$ is the nth derivative of $F(x)$ evaluated at $x = 0$.

This is because if

$$F(x) = f_0 + f_1 x + f_2 x^2 + \cdots,$$

then

$$\begin{aligned} F(0) &= f_0 + f_1 \cdot 0 + f_2 \cdot 0^2 + \cdots \\ &= f_0. \end{aligned}$$

Also,

$$\begin{aligned} F'(x) &= \frac{d}{dx}(F(x)) \\ &= f_1 + 2f_2 x + 3f_3 x^2 + 4f_4 x^3 + \cdots \end{aligned}$$

and so

$$F'(0) = f_1,$$

as desired. Taking second derivatives, we find that

$$\begin{aligned} F''(x) &= \frac{d}{dx}(F'(x)) \\ &= 2f_2 + 3 \cdot 2f_3 x + 4 \cdot 3f_4 x^2 + \cdots \end{aligned}$$

and so

$$F''(0) = 2f_2,$$

which means that

$$f_2 = \frac{F''(0)}{2}.$$

In general,

$$\begin{aligned} F^{(n)} = n!f_n + (n+1)!f_{n+1}x + \frac{(n+2)!}{2} f_{n+2}x^2 + \cdots \\ + \frac{(n+k)!}{k!} f_{n+k}x^k + \cdots \end{aligned}$$

and so

$$F^{(n)}(0) = n!f_n$$

and

$$f_n = \frac{F^{(n)}(0)}{n!},$$

as claimed.

This means that

$$\left\langle F(0), F'(0), \frac{F''(0)}{2!}, \frac{F'''(0)}{3!}, \ldots, \frac{F^{(n)}(0)}{n!}, \ldots \right\rangle \longleftrightarrow F(x). \tag{5.4}$$

The sequence on the left-hand side of Equation 5.4 gives the well-known Taylor Series expansion for a function

$$F(x) = F(0) + F'(0)x + \frac{F''(0)}{2!}x^2 + \frac{F'''(0)}{3!}x^3 + \cdots + \frac{F^{(n)}(0)}{n!}x^n + \cdots.$$

5.4.2 Examples

Let's try this out on a familiar example:

$$F(x) = \frac{1}{1-x}.$$

Computing derivatives, we find that

$$\begin{aligned}
F'(x) &= \frac{1}{(1-x)^2}, \\
F''(x) &= \frac{2}{(1-x)^3}, \\
F'''(x) &= \frac{2 \cdot 3}{(1-x)^4}, \\
&\vdots \\
F^{(n)} &= \frac{n!}{(1-x)^{n+1}}.
\end{aligned}$$

This means that the coefficient of x^n in $1/(1-x)$ is

$$\frac{F^{(n)}(0)}{n!} = \frac{n!}{n!\,(1-0)^{n+1}} = 1.$$

In other words, we have reconfirmed what we already knew; namely, that

$$\frac{1}{1-x} = 1 + x + x^2 + \cdots.$$

Using a similar approach, we can establish some other well-known series:

$$\begin{aligned}
e^x &= 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + \cdots, \\
e^{ax} &= 1 + ax + \frac{a^2}{2!}x^2 + \frac{a^3}{3!}x^3 + \cdots + \frac{a^n}{n!}x^n + \cdots, \\
\ln(1-x) &= -ax - \frac{a^2}{2}x^2 - \frac{a^3}{3}x^3 - \cdots - \frac{a^n}{n}x^n - \cdots.
\end{aligned}$$

But what about the series for

$$F(x) = \frac{x(1+x)}{(1-x)^4}? \tag{5.5}$$

In particular, we need to know the coefficient of x^n in $F(x)$ to determine

$$s_n = \sum_{i=0}^{n} i^2.$$

While it is theoretically possible to compute the nth derivative of $F(x)$, the result is a bloody mess. Maybe these generating functions weren't such a great idea after all....

5.4.3 Massage Helps

In times of stress, a little massage can often help relieve the tension. The same is true for polynomials with painful derivatives. For example, let's take a closer look at Equation 5.5. If we massage it a little bit, we find that

$$F(x) = \frac{x + x^2}{(1-x)^4} = \frac{x}{(1-x)^4} + \frac{x^2}{(1-x)^4}. \tag{5.6}$$

The goal is to find the coefficient of x^n in $F(x)$. If you stare at Equation 5.6 long enough (or if you combine the Right-Shift Rule with the Addition Rule), you will notice that the coefficient of x^n in $F(x)$ is just the sum of

$$\text{the coefficient of } x^{n-1} \text{ in } \frac{1}{(1-x)^4} \text{ and}$$
$$\text{the coefficient of } x^{n-2} \text{ in } \frac{1}{(1-x)^4}.$$

Maybe there is some hope after all. Let's see if we can produce the coefficients for $1/(1-x)^4$. We'll start by looking at the derivatives:

$$\begin{aligned}
F'(x) &= \frac{4}{(1-x)^5},\\
F''(x) &= \frac{4\cdot 5}{(1-x)^6},\\
F'''(x) &= \frac{4\cdot 5\cdot 6}{(1-x)^7},\\
&\vdots\\
F^{(n)}(x) &= \frac{(n+3)!}{6(1-x)^{n+4}}.
\end{aligned}$$

This means that the nth coefficient of $1/(1-x)^4$ is

$$\frac{F^{(n)}(0)}{n!} = \frac{(n+3)!}{6n!} = \frac{(n+3)(n+2)(n+1)}{6}. \tag{5.7}$$

We are now almost done. Equation 5.7 means that the coefficient of x^{n-1} in $1/(1-x)^4$ is

$$\frac{(n+2)(n+1)n}{6} \tag{5.8}$$

and the coefficient[2] of x^{n-2} is

$$\frac{(n+1)n(n-1)}{6}. \tag{5.9}$$

Adding these values produces the desired sum

$$\begin{aligned}
\sum_{i=0}^{n} i^2 &= \frac{(n+2)(n+1)n}{6} + \frac{(n+1)n(n-1)}{6}\\
&= \frac{(2n+1)(n+1)n}{6}.
\end{aligned}$$

This matches the equation on summing the squares of the first n natural numbers we deduced in a prior chapter. Using generating functions to get the result may have seemed to be more complicated, but at least there was no need for guessing or solving a linear system of equations over 4 variables.

You might argue that the massage step was a little tricky. After all, how were you supposed to know that by converting $F(x)$ into the form shown in Equation 5.6, it would be sufficient to compute derivatives of $1/(1-x)^4$, which is easy, instead of derivatives of $x(1+x)/(1-x)^4$, which could be harder than solving a 64-disk Tower of Hanoi problem step-by-step?

The good news is that this sort of massage works for any generating function that is a ratio of polynomials. Even better, you probably already know how to do it from calculus—it's the method of *partial fractions*!

[2] To be precise, Equation 5.8 holds for $n \geq 1$ and Equation 5.9 holds for $n \geq 2$. But since Equation 5.8 is 0 for $n = 1$ and Equation 5.9 is 0 for $n = 1, 2$, both equations hold for all $n \geq 0$.

5.4.4 Partial Fractions

The idea behind partial fractions is to express a ratio of polynomials as a sum of a polynomial and terms of the form

$$\frac{cx^a}{(1-\alpha x)^b} \tag{5.10}$$

where a and b are integers and $b > a \geq 0$. That's because it is easy to compute derivatives of $1/(1-\alpha x)^b$ and thus it is easy to compute the coefficients of Equation 5.10. Let's see why.

Lemma 54. *If $b \in \mathbb{N}^+$, then the nth derivative of $1/(1-\alpha x)^b$ is*

$$\frac{(n+b-1)!\,\alpha^n}{(b-1)!\,(1-\alpha x)^{b+n}}.$$

Proof. The proof is by induction on n. The induction hypothesis $P(n)$ is the statement of the lemma.

Base case ($n = 1$): The first derivative is

$$\frac{b\alpha}{(1-\alpha x)^{b+1}}.$$

This matches

$$\frac{(1+b-1)!\,\alpha^1}{(b-1)!\,(1-\alpha x)^{b+1}} = \frac{b\alpha}{(1-\alpha x)^{b+1}},$$

and so $P(1)$ is true.

Induction step: We next assume $P(n)$ to prove $P(n+1)$ for $n \geq 1$. $P(n)$ implies that the nth derivative of $1/(1-\alpha x)^b$ is

$$\frac{(n+b-1)!\,\alpha^n}{(b-1)!\,(1-\alpha x)^{b+n}}.$$

Taking one more derivative reveals that the $(n+1)$st derivative is

$$\frac{(n+b-1)!\,(b+n)\alpha^{n+1}}{(b-1)!\,(1-\alpha x)^{b+n+1}} = \frac{(n+b)!\,\alpha^{n+1}}{(b-1)!\,(1-\alpha x)^{b+n+1}},$$

which means that $P(n+1)$ is true. Hence, the induction is complete. $\square$

Corollary 55. *If $a, b \in \mathbb{N}$ and $b > a \geq 0$, then for any $n \geq a$, the coefficient of x^n in*

$$\frac{cx^a}{(1-\alpha x)^b}$$

is

$$\frac{c(n-a+b-1)!\,\alpha^{n-a}}{(n-a)!\,(b-1)!}.$$

Proof. By the Taylor Series Rule, the nth coefficient of

$$\frac{1}{(1-\alpha x)^b}$$

is the nth derivative of this expression evaluated at $x = 0$ and then divided by $n!$. By Lemma 54, this is

$$\frac{(n+b-1)!\,\alpha^n}{n!\,(b-1)!\,(1-0)^{b+n}} = \frac{(n+b-1)!\,\alpha^n}{n!\,(b-1)!}.$$

By the Scaling Rule and the Right-Shift Rule, the coefficient of x^n in

$$\frac{cx^\alpha}{(1-\alpha x)^b}$$

is thus

$$\frac{c(n-a+b-1)!\,\alpha^{n-a}}{(n-a)!\,(b-1)!}.$$

as claimed. $\square$

Massaging a ratio of polynomials into a sum of a polynomial and terms of the form in Equation 5.10 takes a bit of work but is generally straightforward. We will show you the process by means of an example.

Suppose our generating function is the ratio

$$F(x) = \frac{4x^3 + 2x^2 + 3x + 6}{2x^3 - 3x^2 + 1}. \tag{5.11}$$

The first step in massaging $F(x)$ is to get the degree of the numerator to be less than the degree of the denominator. This can be accomplished by dividing the numerator by the denominator and taking the remainder, just as in the Fundamental Theorem of Arithmetic—only now we have polynomials instead of numbers. In this case we have

$$\frac{4x^3 + 2x^2 + 3x + 6}{2x^3 - 3x^2 + 1} = 2 + \frac{8x^2 + 3x + 4}{2x^3 - 3x^2 + 1}.$$

The next step is to factor the denominator. This will produce the values of α for Equation 5.10. In this case,

$$\begin{aligned} 2x^3 - 3x^2 + 1 &= (2x+1)(x^2 - 2x + 1) \\ &= (2x+1)(x-1)^2 \\ &= (1-x)^2(1+2x). \end{aligned}$$

We next find values c_1, c_2, c_3 so that

$$\frac{8x^2 + 3x + 4}{2x^3 - 3x^2 + 1} = \frac{c_1}{1+2x} + \frac{c_2}{(1-x)^2} + \frac{c_3 x}{(1-x)^2}. \tag{5.12}$$

This is done by cranking through the algebra:

$$\begin{aligned} \frac{c_1}{1+2x} + \frac{c_2}{(1-x)^2} + \frac{c_3 x}{(1-x)^2} &= \frac{c_1(1-x)^2 + c_2(1+2x) + c_3 x(1+2x)}{(1+2x)(1-x)^2} \\ &= \frac{c_1 - 2c_1 x + c_1 x^2 + c_2 + 2c_2 x + c_3 x + 2c_3 x^2}{2x^3 - 3x^2 + 1} \\ &= \frac{c_1 + c_2 + (-2c_1 + 2c_2 + c_3)x + (c_1 + 2c_3)x^2}{2x^3 - 3x^2 + 1}. \end{aligned}$$

For Equation 5.12 to hold, we need

$$\begin{aligned} 8 &= c_1 + 2c_3, \\ 3 &= -2c_1 + 2c_2 + c_3, \\ 4 &= c_1 + c_2. \end{aligned}$$

Solving these equations, we find that $c_1 = 2,\ c_2 = 2$, and $c_3 = 3$. Hence,

$$\begin{aligned} F(x) &= \frac{4x^3 + 2x^2 + 3x + 6}{2x^3 - 3x^2 + 1} \\ &= 2 + \frac{2}{1+2x} + \frac{2}{(1-x)^2} + \frac{3x}{(1-x)^2}. \end{aligned}$$

Our massage is done! We can now compute the coefficients of $F(x)$ using Corollary 55 and the Sum Rule. The result is

$$f_0 = 2 + 2 + 2 = 6$$

and

$$f_n = \frac{2(n-0+1-1)!\,(-2)^{n-0}}{(n-0)!\,(1-1)!} + \frac{2(n-0+2-1)!\,(1)^{n-0}}{(n-0)!\,(2-1)!} + \frac{3(n-1+2-1)!\,(1)^{n-1}}{(n-1)!\,(2-1)!}$$

$$
\begin{aligned}
&= (-1)^n 2^{n+1} + 2(n+1) + 3n \\
&= (-1)^n 2^{n+1} + 5n + 2
\end{aligned}
$$

for $n \geq 1$.

Aren't you glad that you know that? Actually, this method turns out to be useful in solving linear recurrences, as we'll see in the next section.

5.5 Solving Linear Recurrences

Generating functions can be used to find a solution to any linear recurrence. We'll show you how this is done by means of a familiar example, the Fibonacci recurrence, so that you can more easily understand the similarities and differences of this approach and the method we showed you in studying recurrences.

5.5.1 Finding the Generating Function

Let's begin by recalling the definition of the Fibonacci numbers:

$$
\begin{aligned}
f_0 &= 0 \\
f_1 &= 1 \\
f_n &= f_{n-1} + f_{n-2} \qquad \text{for } n \geq 2.
\end{aligned}
$$

We can expand the final clause into an infinite sequence of equations. Thus, the Fibonacci numbers are defined by:

$$
\begin{aligned}
f_0 =& 0 \\
f_1 =& 1 \\
f_2 =& f_1 + f_0 \\
f_3 =& f_2 + f_1 \\
f_4 =& f_3 + f_2 \\
&\vdots
\end{aligned}
$$

The overall plan is to *define* a function $F(x)$ that generates the sequence on the left side of the equality symbols, which are the Fibonacci numbers. Then we *derive* a function that generates the sequence on the right side. Finally, we equate the two and solve for $F(x)$. Let's try this. First, we define:

$$F(x) = f_0 + f_1 x + f_2 x^2 + f_3 x^3 + f_4 x^4 + \cdots .$$

Now we need to derive a generating function for the sequence:

$$\langle 0,\ 1,\ f_1 + f_0,\ f_2 + f_1,\ f_3 + f_2,\ \ldots \rangle .$$

One approach is to break this into a sum of three sequences for which we know generating functions and then apply the Addition Rule:

$$
\begin{array}{rllllll}
\langle 0, & 1, & 0, & 0, & 0, & \ldots \rangle & \longleftrightarrow\ x \\
\langle 0, & f_0, & f_1, & f_2, & f_3, & \ldots \rangle & \longleftrightarrow\ xF(x) \\
+\ \langle 0, & 0, & f_0, & f_1, & f_2, & \ldots \rangle & \longleftrightarrow\ x^2 F(x) \\
\hline
\multicolumn{6}{r}{\langle 0, 1 + f_0, f_1 + f_0, f_2 + f_1, f_3 + f_2, \ldots \rangle} & \longleftrightarrow\ x + xF(x) + x^2 F(x)
\end{array}
$$

This sequence is almost identical to the right sides of the Fibonacci equations. The one blemish is that the second term is $1 + f_0$ instead of simply 1. However, this amounts to nothing, since $f_0 = 0$ anyway.

If we equate $F(x)$ with the new function $x + xF(x) + x^2F(x)$, then we're implicitly writing down *all* of the equations that define the Fibonacci numbers in one fell swoop:

$$\begin{array}{ccccccccc} F(x) & = f_0 + & f_1 & x + & f_2 & x^2 + & f_3 & x^3 + \cdots \\ \shortparallel & \shortparallel & \shortparallel & & \shortparallel & & \shortparallel & \\ x + xF(x) + x^2F(x) = & 0 & +(1+f_0)x & + & (f_1+f_0)x^2 & + & (f_2+f_1)x^3 & + \cdots \end{array}$$

Solving for $F(x)$ gives the generating function for the Fibonacci sequence:

$$F(x) = x + xF(x) + x^2F(x)$$

so

$$F(x) = \frac{x}{1 - x - x^2}. \tag{5.13}$$

This is pretty cool. After all, who would have thought that the Fibonacci numbers are precisely the coefficients of such a simple function? Even better, this function is a ratio of polynomials and so we can use the method of partial fractions from Section 5.4.4 to find a closed-form expression for the nth Fibonacci number.

5.5.2 Extracting the Coefficients

Repeated differentiation of Equation 5.13 would be very painful. But it is easy to use the method of partial fractions to compute the coefficients. Since the degree of the numerator in Equation 5.13 is less than the degree of the denominator, the first step is to factor the denominator:

$$1 - x - x^2 = (1 - \alpha_1 x)(1 - \alpha_2 x)$$

where $\alpha_1 = (1 + \sqrt{5})/2$ and $\alpha_2 = (1 - \sqrt{5})/2$. These are the same as the roots of the characteristic equation for the Fibonacci recurrence that we found in studying recurrences.That is not a coincidence.

The next step is to find c_1 and c_2 that satisfy

$$\begin{aligned} \frac{x}{1 - x - x^2} &= \frac{c_1}{1 - \alpha_1 x} + \frac{c_2}{1 - \alpha_2 x} \\ &= \frac{c_1(1 - \alpha_2 x) + c_2(1 - \alpha_1 x)}{(1 - \alpha_1 x)(1 - \alpha_2 x)} \\ &= \frac{c_1 + c_2 - (c_1\alpha_2 + c_2\alpha_1)x}{1 - x - x^2}. \end{aligned}$$

Hence,

$$c_1 + c_2 = 0 \qquad \text{and} \qquad -(c_1\alpha_2 + c_2\alpha_1) = 1.$$

Solving these equations, we find that

$$\begin{aligned} c_1 &= \frac{1}{\alpha_1 - \alpha_2} = \frac{1}{\sqrt{5}} \\ c_2 &= \frac{-1}{\alpha_1 - \alpha_2} = \frac{-1}{\sqrt{5}}. \end{aligned}$$

We can now use Corollary 55 and the Sum Rule to conclude that

$$\begin{aligned} f_n &= \frac{\alpha_1^n}{\sqrt{5}} - \frac{\alpha_2^n}{\sqrt{5}} \\ &= \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right). \end{aligned}$$

This is exactly the same formula we derived for the nth Fibonacci number when we studies linear recurrences.

5.5.3 General Linear Recurrences

The method that we just used to solve the Fibonacci recurrence can also be used to solve general linear recurrences of the form

$$f_n = a_1 f_{n-1} + a_2 f_{n-2} + \cdots + a_d f_{n-d} + g_n$$

for $n \geq d$. The generating function for $\langle f_0, f_1, f_2, \dots \rangle$ is

$$F(x) = \frac{h(x) + G(x)}{1 - a_1 x - a_2 x^2 - \cdots - a_d x^d}$$

where $G(x)$ is the generating function for the sequence

$$\langle \overbrace{0, 0, \dots, 0}^{d}, g_d, g_{d+1}, g_{d+2}, \dots \rangle$$

and $h(x)$ is a polynomial of degree at most $d-1$ that is based on the values of f_0, f_1, $\dots$, f_{d-1}. In particular,

$$h(x) = \sum_{i=0}^{d-1} h_i x^i$$

where

$$h_i = f_0 - a_1 f_{i-1} - a_2 f_{i-2} - \cdots - a_i f_0$$

for $0 \leq i < d$.

To solve the recurrence, we use the method of partial fractions described in Section 5.4.4 to find a closed-form expression for $F(x)$. This can be easy or hard to do depending on $G(x)$.

Practice Problems

Exercise 5.5.1 The famous mathematician, Fibonacci, has decided to start a rabbit farm to fill up his time while he's not making new sequences to torment future college students. Fibonacci starts his farm on month zero (being a mathematician), and at the start of month one he receives his first pair of rabbits. Each pair of rabbits takes a month to mature, and after that breeds to produce one new pair of rabbits each month. Fibonacci decides that in order never to run out of rabbits or money, every time a batch of new rabbits is born, he'll sell a number of newborn pairs equal to the total number of pairs he had three months earlier. Fibonacci is convinced that this way he'll never run out of stock.

(a) Define the number r_n of pairs of rabbits Fibonacci has in month n, using a recurrence relation. That is, define r_n in terms of various r_i where $i < n$.

(b) Let $R(x)$ be the generating function for rabbit pairs,

$$R(x) \triangleq r_0 + r_1 x + r_2 x^2 + \cdots$$

Express $R(x)$ as a quotient of polynomials.

(c) Find a partial fraction decomposition of the generating function $R(x)$.

(d) Finally, use the partial fraction decomposition to come up with a closed form expression for the number of pairs of rabbits Fibonacci has on his farm on month n.

Exercise 5.5.2 Less well-known than the *Towers of Hanoi*—but no less fascinating—are the Towers of Sheboygan. As in Hanoi, the puzzle in Sheboygan involves 3 posts and n rings of different sizes. The rings are placed on post #1 in order of size with the smallest ring on top and largest on bottom.

The objective is to transfer all n rings to post #2 via a sequence of moves. As in the Hanoi version, a move consists of removing the top ring from one post and dropping it onto another post with the restriction that a larger ring can never lie above a smaller ring. But unlike Hanoi, a local ordinance requires that *a ring can only be moved from post #1 to post #2, from post #2 to post #3, or from post #3 to post #1.* Thus, for example, moving a ring directly from post #1 to post #3 is not permitted.

(a) One procedure that solves the Sheboygan puzzle is defined recursively: to move an initial stack of n rings to the next post, move the top stack of $n-1$ rings to the furthest post by moving it to the next post two times, then move the big, nth ring to the next post, and finally move the top stack another two times to land on top of the big ring. Let s_n be the number of moves that this procedure uses. Write a simple linear recurrence for s_n.

(b) Let $S(x)$ be the *generating function* for the sequence $\langle s_0, s_1, s_2, \ldots \rangle$. Carefully show that

$$S(x) = \frac{x}{(1-x)(1-4x)}.$$

(c) Give a simple formula for s_n.

(d) A better (indeed optimal, but we won't prove this) procedure to solve the Towers of Sheboygan puzzle can be defined in terms of two *mutually recursive* procedures, procedure $P_1(n)$ for moving a stack of n rings 1 pole forward, and $P_2(n)$ for moving a stack of n rings 2 poles forward. This is trivial for $n = 0$. For $n > 0$, define:

$P_1(n)$: Apply $P_2(n-1)$ to move the top $n-1$ rings two poles forward to the third pole. Then move the remaining big ring once to land on the second pole. Then apply $P_2(n-1)$ again to move the stack of $n-1$ rings two poles forward from the third pole to land on top of the big ring.

$P_2(n)$: Apply $P_2(n-1)$ to move the top $n-1$ rings two poles forward to land on the third pole. Then move the remaining big ring to the second pole. Then apply $P_1(n-1)$ to move the stack of $n-1$ rings one pole forward to land on the first pole. Now move the big ring 1 pole forward again to land on the third pole. Finally, apply $P_2(n-1)$ again to move the stack of $n-1$ rings two poles forward to land on the big ring.

Let t_n be the number of moves needed to solve the Sheboygan puzzle using procedure $P_1(n)$. Show that

$$t_n = 2t_{n-1} + 2t_{n-2} + 3,$$

for $n > 1$.

Hint: Let u_n be the number of moves used by procedure $P_2(n)$. Express each of t_n and u_n as linear combinations of t_{n-1} and u_{n-1} and solve for t_n.

(e) Derive values a, b, c, α, β such that

$$t_n = a\alpha^n + b\beta^n + c.$$

Conclude that $t_n = o(s_n)$.

Exercise 5.5.3 Taking derivatives of generating functions is another useful operation. This is done termwise, that is, if

$$F(x) = f_0 + f_1x + f_2x^2 + f_3x^3 + \cdots,$$

then

$$F'(x) \triangleq f_1 + 2f_2x + 3f_3x^2 + \cdots.$$

For example,

$$\frac{1}{(1-x)^2} = \left(\frac{1}{(1-x)}\right)' = 1 + 2x + 3x^2 + \cdots$$

so

$$H(x) \triangleq \frac{x}{(1-x)^2} = 0 + 1x + 2x^2 + 3x^3 + \cdots$$

is the generating function for the sequence of nonnegative integers. Therefore

$$\frac{1+x}{(1-x)^3} = H'(x) = 1 + 2^2x + 3^2x^2 + 4^2x^3 + \cdots,$$

so

$$\frac{x^2+x}{(1-x)^3} = xH'(x) = 0 + 1x + 2^2x^2 + 3^2x^3 + \cdots + n^2x^n + \cdots$$

is the generating function for the nonnegative integer squares.

(a) Prove that for all $k \in \mathbb{N}$, the generating function for the nonnegative integer kth powers is a quotient of polynomials in x. That is, for all $k \in \mathbb{N}$ there are polynomials $R_k(x)$ and $S_k(x)$ such that

$$[x^n]\left(\frac{R_k(x)}{S_k(x)}\right) = n^k. \tag{5.14}$$

Hint: Observe that the derivative of a quotient of polynomials is also a quotient of polynomials. It is not necessary work out explicit formulas for R_k and S_k to prove this part.

(b) Conclude that if $f(n)$ is a function on the nonnegative integers defined recursively in the form

$$f(n) = af(n-1) + bf(n-2) + cf(n-3) + p(n)\alpha^n$$

where the $a, b, c, \alpha \in \mathbb{C}$ and p is a polynomial with complex coefficients, then the generating function for the sequence $f(0), f(1), f(2), \ldots$ will be a quotient of polynomials in x, and hence there is a closed form expression for $f(n)$.

Hint: Consider

$$\frac{R_k(\alpha x)}{S_k(\alpha x)}$$

5.6 Counting with Generating Functions

Generating functions are particularly useful for solving counting problems. In particular, problems involving choosing items from a set often lead to nice generating functions by letting the coefficient of x^n be the number of ways to choose n items.

5.6.1 Choosing Distinct Items from a Set

The generating function for binomial coefficients follows directly from the *Binomial Theorem*:

$$\begin{aligned}\left\langle \tbinom{k}{0}, \tbinom{k}{1}, \tbinom{k}{2}, \ldots, \tbinom{k}{k}, 0, 0, 0, \ldots \right\rangle \longleftrightarrow\ & \tbinom{k}{0} + \tbinom{k}{1}x + \tbinom{k}{2}x^2 + \cdots + \tbinom{k}{k}x^k \\ &= (1+x)^k\end{aligned}$$

Thus, the coefficient of x^n in $(1+x)^k$ is $\binom{k}{n}$, the number of ways to choose n distinct items[3] from a set of size k. For example, the coefficient of x^2 is $\binom{k}{2}$, the number of ways to choose 2 items from a set with k elements. Similarly, the coefficient of x^{k+1} is the number of ways to choose $k+1$ items from a size k set, which is zero.

[3] Watch out for the reversal of the roles that k and n played in earlier examples; we're led to this reversal because we've been using n to refer to the power of x in a power series.

5.6.2 Building Generating Functions that Count

Often we can translate the description of a counting problem directly into a generating function for the solution. For example, we could figure out that $(1+x)^k$ generates the number of ways to select n distinct items from a k-element set without resorting to the Binomial Theorem or even fussing with binomial coefficients! Let's see how.

First, consider a single-element set $\{a_1\}$. The generating function for the number of ways to select n elements from this set is simply $1+x$: we have 1 way to select zero elements, 1 way to select one element, and 0 ways to select more than one element. Similarly, the number of ways to select n elements from the set $\{a_2\}$ is also given by the generating function $1+x$. The fact that the elements differ in the two cases is irrelevant.

Now here is the main trick: *the generating function for choosing elements from a union of disjoint sets is the product of the generating functions for choosing from each set.* We'll justify this in a moment, but let's first look at an example. According to this principle, the generating function for the number of ways to select n elements from the $\{a_1, a_2\}$ is:

$$\underbrace{(1+x)}_{\text{select from }\{a_1\}} \cdot \underbrace{(1+x)}_{\text{select from }\{a_2\}} = \underbrace{(1+x)^2}_{\text{select from }\{a_1,a_2\}} = 1+2x+x^2.$$

Sure enough, for the set $\{a_1, a_2\}$, we have 1 way to select zero elements, 2 ways to select one element, 1 way to select two elements, and 0 ways to select more than two elements.

Repeated application of this rule gives the generating function for selecting n items from a k-element set $\{a_1, a_2, \ldots, a_k\}$:

$$\underbrace{(1+x)}_{\text{select from }\{a_1\}} \cdot \underbrace{(1+x)}_{\text{select from }\{a_2\}} \cdots \underbrace{(1+x)}_{\text{select from }\{a_k\}} = \underbrace{(1+x)^k}_{\substack{\text{select from}\\ \{a_1,a_2,\ldots,a_k\}}}$$

This is the same generating function that we obtained by using the Binomial Theorem. But this time around, we translated directly from the counting problem to the generating function.

We can extend these ideas to a general principle:

Rule 56 (*Convolution Rule*)**.** *Let $A(x)$ be the generating function for selecting items from set $\mathcal{A}$, and let $B(x)$ be the generating function for selecting items from set $\mathcal{B}$. If $\mathcal{A}$ and $\mathcal{B}$ are disjoint, then the generating function for selecting items from the union $\mathcal{A} \cup \mathcal{B}$ is the product $A(x) \cdot B(x)$.*

This rule is rather ambiguous: what exactly are the rules governing the selection of items from a set? Remarkably, the Convolution Rule remains valid under *many* interpretations of selection. For example, we could insist that distinct items be selected or we might allow the same item to be picked a limited number of times or any number of times. Informally, the only restrictions are that (1) the order in which items are selected is disregarded and (2) restrictions on the selection of items from sets $\mathcal{A}$ and $\mathcal{B}$ also apply in selecting items from $\mathcal{A} \cup \mathcal{B}$. (Formally, there must be a bijection between n-element selections from $\mathcal{A} \cup \mathcal{B}$ and ordered pairs of selections from $\mathcal{A}$ and $\mathcal{B}$ containing a total of n elements.)

To count the number of ways to select n items from $\mathcal{A} \cup \mathcal{B}$, we observe that we can select n items by choosing j items from $\mathcal{A}$ and $n-j$ items from $\mathcal{B}$, where j is any number from 0 to n. This can be done in $a_j b_{n-j}$ ways. Summing over all the possible values of j gives a total of

$$a_0 b_n + a_1 b_{n-1} + a_2 b_{n-2} + \cdots + a_n b_0$$

ways to select n items from $\mathcal{A} \cup \mathcal{B}$. By the Product Rule, this is precisely the coefficient of x^n in the series for $A(x)B(x)$.

5.6.3 Choosing Items with Repetition

The first counting problem we considered was the number of ways to select a dozen doughnuts when five flavors were available. We can generalize this question as follows: in how many ways can we select n items from a k-element set if we're allowed to pick the same item multiple times? In these terms, the doughnut problem asks how many ways we can select $n = 12$ doughnuts from the set of $k = 5$ flavors

$$\{\text{chocolate}, \text{lemon-filled}, \text{sugar}, \text{glazed}, \text{plain}\}$$

where, of course, we're allowed to pick several doughnuts of the same flavor. Let's approach this question from a generating functions perspective.

Suppose we make n choices (with repetition allowed) of items from a set containing a single item. Then there is one way to choose zero items, one way to choose one item, one way to choose two items, etc. Thus, the generating function for choosing n elements with repetition from a 1-element set is:

$$\langle 1, 1, 1, 1, \ldots \rangle \longleftrightarrow 1 + x + x^2 + x^3 + \cdots = \frac{1}{1-x}.$$

The Convolution Rule says that the generating function for selecting items from a union of disjoint sets is the product of the generating functions for selecting items from each set:

$$\underbrace{\frac{1}{1-x}}_{\text{choose } a_1\text{'s}} \cdot \underbrace{\frac{1}{1-x}}_{\text{choose } a_2\text{'s}} \cdots \underbrace{\frac{1}{1-x}}_{\text{choose } a_k\text{'s}} = \underbrace{\frac{1}{(1-x)^k}}_{\substack{\text{repeatedly choose from} \\ \{a_1, a_2, \ldots, a_k\}}}$$

Therefore, the generating function for choosing items from a k-element set with repetition allowed is $1/(1-x)^k$. Computing derivatives and applying the Taylor Series Rule, we can find that the coefficient of x^n in $1/(1-x)^k$ is

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

This is *the bookkeeper rule* from the principles of counting—namely there are $\binom{n+k-1}{n}$ ways to select n items with replication from a set of k items.

Practice Problems

Exercise 5.6.1 You would like to buy a bouquet of flowers. You find an online service that will make bouquets of **lilies**, **roses** and **tulips**, subject to the following constraints:

- there must be at most 2 lilies,
- there must be an even number of tulips,
- there can be any number of roses.

Example: A bouquet of 4 tulips, 5 roses and no lilies satisfies the constraints.

Let f_n be the number of possible bouquets with n flowers that fit the service's constraints. Express $F(x)$, the generating function corresponding to $\langle f_0, f_1, f_2, \ldots \rangle$, as a quotient of polynomials (or products of polynomials). You do not need to simplify this expression.

Exercise 5.6.2 Let $A(x) = \sum_{n=0}^{\infty} a_n x^n$. Then it's easy to check that

$$a_n = \frac{A^{(n)}(0)}{n!},$$

where $A^{(n)}$ is the nth derivative of A. Use this fact (which you may assume) instead of the *Convolution Counting Principle* from the course text, to prove that

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

So if we didn't already know *the bookkeeper rule*, we could have proved it from this calculation and the Convolution Rule for generating functions.

Exercise 5.6.3 We are interested in generating functions for the number of different ways to compose a bag of n donuts subject to various restrictions. For each of the restrictions in parts (5.6.3)-(5.6.3) below, find a closed form for the corresponding generating function.

(a) All the donuts are chocolate and there are at least 3.

(b) All the donuts are glazed and there are at most 2.

(c) All the donuts are coconut and there are exactly 2 or there are none.

(d) All the donuts are plain and their number is a multiple of 4.

(e) The donuts must be chocolate, glazed, coconut, or plain with the numbers of each flavor subject to the constraints above.

(f) Now find a closed form for the number of ways to select n donuts subject to the above constraints.

Exercise 5.6.4 Solve the following problems.

(a) Let

$$S(x) \triangleq \frac{x^2+x}{(1-x)^3}.$$

What is the coefficient of x^n in the generating function series for $S(x)$?

(b) Explain why $S(x)/(1-x)$ is the generating function for the sums of squares. That is, the coefficient of x^n in the series for $S(x)/(1-x)$ is $\sum_{k=1}^{n} k^2$.

(c) Use the previous parts to prove that

$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}.$$

Exercise 5.6.5 We will use generating functions to determine how many ways there are to use pennies, nickels, dimes, quarters, and half-dollars to give n cents change.

(a) Write the generating function $P(x)$ for for the number of ways to use only pennies to make n cents.

(b) Write the generating function $N(x)$ for the number of ways to use only nickels to make n cents.

(c) Write the generating function for the number of ways to use only nickels and pennies to change n cents.

(d) Write the generating function for the number of ways to use pennies, nickels, dimes, quarters, and half-dollars to give n cents change.

(e) Explain how to use this function to find out how many ways are there to change 50 cents; you do *not* have to provide the answer or actually carry out the process.

Exercise 5.6.6 Define the sequence $r_0, r_1, r_2, \ldots$ recursively by the rule that $r_0 \triangleq 1$ and

$$r_n \triangleq 7r_{n-1} + (n+1) \qquad \text{for } n > 0.$$

Let $R(x) \triangleq \sum_0^\infty r_n x^n$ be the generating function of this sequence. Express $R(x)$ as a quotient of polynomials or products of polynomials. You do *not* have to find a closed form for r_n.

5.6.4 Fruit Salad

In this chapter, we have covered a lot of methods and rules for using generating functions. We'll now do an example that demonstrates how the rules and methods can be combined to solve a more challenging problem—making fruit salad.

In how many ways can we make a salad with n fruits subject to the following constraints?

- The number of apples must be even.
- The number of bananas must be a multiple of 5.
- There can be at most four oranges.
- There can be at most one pear.

For example, there are 7 ways to make a salad with 6 fruits:

Apples	6	4	4	2	2	0	0
Bananas	0	0	0	0	0	5	5
Oranges	0	2	1	4	3	1	0
Pears	0	0	1	0	1	0	1

These constraints are so complicated that the problem seems hopeless! But generating functions can solve the problem in a straightforward way.

Let's first construct a generating function for choosing apples. We can choose a set of 0 apples in one way, a set of 1 apple in zero ways (since the number of apples must be even), a set of 2 apples in one way, a set of 3 apples in zero ways, and so forth. So we have:

$$A(x) = 1 + x^2 + x^4 + x^6 + \cdots = \frac{1}{1-x^2}.$$

Similarly, the generating function for choosing bananas is:

$$B(x) = 1 + x^5 + x^{10} + x^{15} + \cdots = \frac{1}{1-x^5}.$$

We can choose a set of 0 oranges in one way, a set of 1 orange in one way, and so on. However, we can not choose more than four oranges, so we have the generating function:

$$O(x) = 1 + x + x^2 + x^3 + x^4 = \frac{1-x^5}{1-x}.$$

Here we're using the geometric sum formula. Finally, we can choose only zero or one pear, so we have:

$$P(x) = 1 + x.$$

The Convolution Rule says that the generating function for choosing from among all four kinds of fruit is:

$$A(x)B(x)O(x)P(x) = \frac{1}{1-x^2}\frac{1}{1-x^5}\frac{1-x^5}{1-x}(1+x)$$

$$= \frac{1}{(1-x)^2}$$
$$= 1 + 2x + 3x^2 + 4x^3 + \cdots .$$

Almost everything cancels! We're left with $1/(1-x)^2$, which we found a power series for earlier: the coefficient of x^n is simply $n+1$. Thus, the number of ways to make a salad with n fruits is just $n+1$. This is consistent with the example we worked out at the start, since there were 7 different salads containing 6 fruits. *Amazing!*

Practice Problems

Exercise 5.6.7 Let a_n be the number of ways to make change for $\$n$ using \$2 and \$3 coins. For example, $a_5 = 1$ because the only way to make change for \$5 is with one \$2 coin and one \$3 coin, but $a_6 = 2$ because there are two ways to make change for \$6, namely using three \$2 coins or using two \$3 coins.

Express the generating function for the sequence of a_n's as a rational function (quotient of products of polynomials). You need not simplify your formula or solve for a_n.

Exercise 5.6.8 Write a formula for the generating function whose successive coefficients are given by the sequence:

(a) 0, 0, 1, 1, 1,...

(b) 1, 1, 0, 0, 0,...

(c) 1, 0, 1, 0, 1, 0, 1,...

(d) 1, 4, 6, 4, 1, 0, 0, 0,...

(e) 1, 2, 3, 4, 5,...

(f) 1, 4, 9, 16, 25,...

(g) 1, 1, 1/2, 1/6, 1/24, 1/120,...

Exercise 5.6.9 Miss McGillicuddy never goes outside without a collection of pets. In particular:

- She brings a positive number of songbirds, which always come in pairs.
- She may or may not bring her alligator, Freddy.
- She brings at least 2 cats.
- She brings two or more chihuahuas and labradors leashed together in a line.

Let P_n denote the number of different collections of n pets that can accompany her, where we regard chihuahuas and labradors leashed in different orders as different collections.

For example, $P_6 = 4$ since there are 4 possible collections of 6 pets:

- 2 songbirds, 2 cats, 2 chihuahuas leashed in line
- 2 songbirds, 2 cats, 2 labradors leashed in line
- 2 songbirds, 2 cats, a labrador leashed behind a chihuahua
- 2 songbirds, 2 cats, a chihuahua leashed behind a labrador

(a) Let
$$P(x) \triangleq P_0 + P_1 x + P_2 x^2 + P_3 x^3 + \cdots$$
be the generating function for the number of Miss McGillicuddy's pet collections. Verify that
$$P(x) = \frac{4x^6}{(1-x)^2(1-2x)}.$$

(b) Find a closed form expression for P_n.

Exercise 5.6.10 Generating functions provide an interesting way to count the number of strings of matched brackets. To do this, we'll use a description of these strings as the set GoodCount of strings of brackets with a *good count*.

Namely, one precise way to determine if a string is matched is to start with 0 and read the string from left to right, adding 1 to the count for each left bracket and subtracting 1 from the count for each right bracket. For example, here are the counts for the two strings above

$$\begin{array}{ccccccccccccc} & [&] &] & [& [& [& [& [&] &] &] &] \\ 0 & 1 & 0 & -1 & 0 & 1 & 2 & 3 & 4 & 3 & 2 & 1 & 0 \end{array}$$

$$\begin{array}{ccccccccccc} & [& [& [&] &] & [&] &] & [&] \\ 0 & 1 & 2 & 3 & 2 & 1 & 2 & 1 & 0 & 1 & 0 \end{array}$$

A string has a *good count* if its running count never goes negative and ends with 0. So the second string above has a good count, but the first one does not because its count went negative at the third step.

Definition. Let

$$\text{GoodCount} \triangleq \{s \in \{], [\}^* \mid s \text{ has a good count}\}.$$

The matched strings can now be characterized precisely as this set of strings with good counts.

Let c_n be the number of strings in GoodCount with exactly n left brackets, and let $C(x)$ be the generating function for these numbers:

$$C(x) \triangleq c_0 + c_1 x + c_2 x^2 + \cdots .$$

(a) The *wrap* of a string s is the string, $[s]$, that starts with a left bracket followed by the characters of s, and then ends with a right bracket. Explain why the generating function for the wraps of strings with a good count is $xC(x)$.

Hint: The wrap of a string with good count also has a good count that starts and ends with 0 and remains *positive* everywhere else.

(b) Explain why, for every string s with a good count, there is a unique sequence of strings $s_1, \ldots, s_k$ that are wraps of strings with good counts and $s = s_1 \cdots s_k$. For example, the string $r \triangleq [[]][][[][]] \in \text{GoodCount}$ equals $s_1 s_2 s_3$ where $s_1 \triangleq [[]], s_2 \triangleq [], s_3 \triangleq [[][]]$, and this is the only way to express r as a sequence of wraps of strings with good counts.

(c) Conclude that

$$C = 1 + xC + (xC)^2 + \cdots + (xC)^n + \cdots \qquad \text{(i)}$$

so

$$C = \frac{1}{1 - xC} \qquad \text{(ii)}$$

and hence

$$C = \frac{1 \pm \sqrt{1 - 4x}}{2x}. \qquad \text{(iii)}$$

Let $D(x) \triangleq 2xC(x)$. Expressing D as a power series

$$D(x) = d_0 + d_1 x + d_2 x^2 + \cdots ,$$

we have

$$c_n = \frac{d_{n+1}}{2}. \qquad \text{(iv)}$$

(d) Use (iii), (iv), and the value of c_0 to conclude that

$$D(x) = 1 - \sqrt{1-4x}.$$

(e) Prove that

$$d_n = \frac{(2n-3)\cdot(2n-5)\cdots 5\cdot 3\cdot 1\cdot 2^n}{n!}.$$

Hint: $d_n = D^{(n)}(0)/n!$

(f) Conclude that

$$c_n = \frac{1}{n+1}\binom{2n}{n}.$$

Exercise 5.6.11 T-Pain is planning an epic boat trip and he needs to decide what to bring with him.

- He must bring some burgers, but they only come in packs of 6.
- He and his two friends can't decide whether they want to dress formally or casually. He'll either bring 0 pairs of flip flops or 3 pairs.
- He doesn't have very much room in his suitcase for towels, so he can bring at most 2.
- In order for the boat trip to be truly epic, he has to bring at least 1 nautical-themed pashmina afghan.

(a) Let $B(x)$ be the generating function for the number of ways to bring n burgers, $F(x)$ for the number of ways to bring n pairs of flip flops, $T(x)$ for towels, and $A(x)$ for Afghans. Write simple formulas for each of these.

(b) Let g_n be the the number of different ways for T-Pain to bring n items (burgers, pairs of flip flops, towels, and/or afghans) on his boat trip. Let $G(x)$ be the generating function $\sum_{n=0}^{\infty} g_n x^n$. Verify that

$$G(x) = \frac{x^7}{(1-x)^2}.$$

(c) Find a simple formula for g_n.