Math 116 Professor: Padraic Bartlett

Lecture 1: Basic Counting

Week 1 *UCSB 2015*

1 How to Count

Combinatorics, at its heart, is the **art of counting**.

This might seem like a silly way to describe a field of mathematics; counting, after all, is something that you learned how to do at a very young age.

So let's clarify what we mean by "counting." On one hand, it is very easy to see that there are four elements in a set like

$$A = \{3, 5, 7, \text{Snape}\}.$$

In whatever definition we come up with for "counting," this should be an acceptable answer!

But this can get a little trickier.

1.1 How to Count: Multiplication

Consider the following problem:

Problem. Suppose that we have k different kinds of postcards, and n friends. In how many ways can we mail out all of our postcards at once to our friends?

As phrased above, this doesn't look like a question about counting, so let's rephrase it a bit. In the setup above, a valid "way" to mail postcards to friends is some way to assign each friend to a postcard, so that each friend is assigned to at least at least one postcard (because we're mailing each of our friends a postcard) and no friend is assigned to two different postcards at the same time. In other words, a "way" to mail postcards is just a function from the set[1] $[n] = \{1, 2, 3, \ldots n\}$ of postcards to our set $[k] = \{1, 2, 3, \ldots k\}$ of friends!

In other words, we want to "count" the number of elements of the following set:

$$A = \Big\{\text{all of the functions that map } [n] \text{ to } [k]\Big\}.$$

We can do this! Think about how any function $f : [n] \to [k]$ is constructed. For each value in $[n] = \{1, 2, \ldots n\}$, we have to pick exactly one value from $[k]$. Doing this for each value in $[n]$ completely determines our function; furthermore, any two functions f, g are different if and only if there is some value $m \in [n]$ at which we made a different choice (i.e. where $f(m) \neq g(m)$.)

$$\underbrace{\boxed{\text{k choices}} \cdot \boxed{\text{k choices}} \cdot \ldots \cdot \boxed{\text{k choices}}}_{n \text{ total slots}}$$

[1] Some useful notation: $[n]$ denotes the collection of all integers from 1 to n, i.e. $\{1, 2, \ldots n\}$.

Consequently, we have

$$\underbrace{k \cdot k \cdot \ldots \cdot k}_{n} = k^n$$

total ways in which we can construct distinct functions. This gives us this answer k^N to our problem!

This looks like an excellent sort of answer to a counting problem: given a set defined by parameters n, k, we created a closed-form algebraic expression k^n for the number of elements in that set! Again, under any theory of counting that we come up with, this should count as a pretty good answer.

Alongside our answer, we also came up with a fairly useful **method** for counting at the same time. Recall that given sets $S_1, \ldots S_n$, the **Cartesian product** of these sets, denoted as $S_1 \times S_2 \times \ldots S_n$ or $\prod_{i=1}^n S_i$, is simply the collection of all ordered n-tuples of elements from these sets, such that for any i, the i-th element in our n-tuple is a member of S_i: that is,

$$\prod_{i=1}^{n} S_i = \{(s_1, s_2, \ldots s_n) \mid \forall i, s_i \in S_i\}.$$

With this language in mind, our method is simply the following:

Observation. (**Multiplication[2] principle.**) Suppose that you have a set A that can be written as the Cartesian product of n sets $S_1, \ldots S_n$. Then

$$|A| = |S_1| \cdot |S_2| \cdot \ldots |S_n| = \prod_{i=1}^{n} |S_i|.$$

This is not the only way that multiplication comes up when counting sets. Let's try changing our postcard problem a bit from before:

Problem. Suppose that we have k different kinds of postcards, n friends, and that we still want to mail these postcards to our friends all at once. This time, however, we don't want to duplicate any of our postcards! In how many ways can we mail out postcards now?

We can solve this using the same reasoning as before. We can still describe each way of sending postcards as a sequence of choices:

$$\underbrace{\boxed{?\text{ choices}} \cdot \boxed{?\text{ choices}} \cdot \ldots \cdot \boxed{?\text{ choices}}}_{n \text{ total slots}}$$

As before, we still have k possibilities for what to send to our first friend. However, the multiplication principle doesn't apply here: because we don't want to have any repetitions,

[2]In practice, no-one will ever actually refer to this as the "multiplication principle:" this falls under the collection of mathematical results that are so commonly used they show up without comment.

we only have $k-1$ choices for our second slot, instead of k as before! In general, we have the following sequence of choices:

$$\underbrace{\boxed{\text{k choices}} \cdot \boxed{\text{k-1 choices}} \cdot \boxed{\text{k-2 choices}} \cdot \ldots \cdot \boxed{\text{k- (n-1) choices}}}_{n \text{ total slots}},$$

which translates into

$$k \cdot (k-1) \cdot \ldots \cdot (k-(n-1))$$

many choices in total.

A convenient way to describe the above is as the following quantity:

$$\begin{aligned} k \cdot (k-1) \cdot \ldots \cdot (k-(n-1)) &= \frac{\Big(k \cdot (k-1) \cdot \ldots \cdot (k-(n-1))\Big) \cdot \Big((k-n) \cdot (k-(n+1)) \cdot \ldots \cdot 3 \cdot 2 \cdot 1\Big)}{\Big((k-n) \cdot (k-(n+1)) \cdot \ldots \cdot 3 \cdot 2 \cdot 1\Big)} \\ &= \frac{k!}{(k-n)!}, \end{aligned}$$

where by $n!$ we mean **n-factorial**, the product of all of the natural numbers between 1 and n, inclusive. (By convention we define $0! = 1$, as another natural definition for $n!$ is the number of ways of ordering a list of n objects, and there is exactly one way to order an empty list.)

This is not quite the multiplication problem: in this setting, each time we make a choice it changes our possible later choices! Therefore, we cannot express our set as the product of several other sets, as in those setting the choice in any coördinate may affect our choices in other coördinates.

That said, the key idea from both problems — given a complicated task, break it down into smaller steps, and use all of the choices in those smaller steps to understand the larger problem — is a good one.

Let's tweak the postcard problem again![3]

Problem. We now only have one kind of postcard, of which we have k copies; we also have n friends. In how many ways can we send out cards all at once to our friends, if we want to not send any friend a card multiple times, and want to send as many friends cards as possible?

In this situation, we actually need to consider two cases. If $k \geq n$, our problem is very simple! We send each friend a card; we can do this, and by our restrictions above cannot do anything else. So there is exactly one solution.

In the case where $k < n$, things are a bit harder. We now need to "decide" which of our friends can get a card.

[3] In mathematics: whenever you have a problem at hand, constantly look for modifications like these to make to the problem! If you're stuck, it can give you different avenues to approach or think about the problem; conversely, if you think you understand the problem, this can be a way to test and deepen that understanding.

At first, you might think that this is a similar problem to the one above. In particular, think of making our choices postcard-by-postcard instead of friend by friend: then we have k choices to be made in order, one for each card.

$$\underbrace{\boxed{\text{? choices}} \cdot \boxed{\text{? choices}} \cdot \ldots \cdot \boxed{\text{? choices}}}_{k \text{ total slots}}$$

If we were to naïvely proceed as above, we might think that we have n friends to choose from for our first card, $n-1$ friends to choose from for our second card, and so on/so forth until we have $n-(k-1)$ choices for our k-th card.

$$\underbrace{\boxed{\text{n choices}} \cdot \boxed{\text{n-1 choices}} \cdot \ldots \cdot \boxed{\text{? choices}}}_{n-(k-1) \text{ total slots}}$$

This gives us $(n)(n-1) \cdot \ldots \cdot (n-(k-1))$ as our answer.

On getting this answer, our first reaction as mathematicians should be to be **skeptical**. Not because there's anything obviously wrong with this answer! We used methods that worked before, and seem to have gotten to a useful answer.

Rather, we should be skeptical because **we should always be skeptical.** Mathematics, seemingly ninety-five percent of the time, is the art of being wrong. When you are first starting out, you might hope that this is merely a consequence of being new to the field, and that after a sufficiently long amount of time you learn to stop making silly errors and you just produce completely correct things all the time! This is ... not true. If anything, I would claim that the rate of mistakes increases as you keep going; not because you keep making the same ones, but rather because you keep learning new and exciting ways to prove false things! Ask any researcher, and they can tell you stories of mistakes they traced out for weeks/months/years at a time.

Because of this, it is imperative that (whenever we come up with a mathematical result) we look for ways to **check** it. In combinatorics, this is usually a remarkably easy thing to do: we've made a claim about finite numbers of cards and friends, right? Well, let's check it for some small values of n and k!

For instance, let's try $n = 3, k = 2$. In this case, we have two cards, and want to pick two of our three friends to send cards to. Equivalently, we have two cards and we need to pick **one** of our friends to not send a card to, which tells us that there should be three ways to send such cards, one for each way to pick one friend out of three to neglect.

However, our formula says that because $n = 3, k = 2$ we have $n \cdot (n-1) = 3 \cdot 2 = 6$ many ways to do this. Six... is not three. So what went wrong?

Well, if you look at our formula, you can see that we're implicitly keeping track of the **order** we send out our k postcards: i.e. we would regard sending postcard 1 to friend 1 and postcard 2 to friend 2 as different from sending postcard 1 to friend 2 and postcard 2 to friend 1. This is a mistake, though! Our k postcards are all the same, and we're sending them all out at once; so there is no difference between those two cases!

Therefore, our process above is **over-counting** the total number of ways to send out postcards. To fix this, we need to correct for our over-counting errors above. Notice that for any given set of k distinct friends, there are $k!$ different ways to order that set. Therefore, if we are looking at the collection of ordered length-k sequences of friends,

- Each unordered sequence of k friends corresponds to $k!$ different ordered friend lists, and

- Each ordered sequence of k friends corresponds to exactly one unordered sequence of k friends.

In other words, if we let OF denote the collection of all ordered lists of k friends, and UF denote the collection of all unordered lists of k friends, we can break the set OF into $|UF|$-many disjoint pieces, one for each unordered friend list, and thus of size $k!$. But if OF consists of $|UF|$-many pieces all of size $k!$, we've proven that

$$k! \cdot |UF| = |OF|.$$

Therefore, if we use the fact that we've counted OF and proven above that $OF = (n)(n-1) \cdot \ldots \cdot (n-(k-1))$, we can conclude that there are

$$|UF| = \frac{n(n-1) \cdot \ldots (n-(k-1))}{k!}$$

many ways to pick k of our friends to send postcards.

This concept — given a set of k things, in how many ways can we pick m of them if we don't care about their order — is an incredibly useful one, and as such we have notation to describe this object. Namely, we denote this quantity via the **binomial coefficient** $\binom{k}{m}$, and define[4]

$$\binom{k}{m} := \frac{k!}{(k-m)! \cdot m!}.$$

1.2 How to Count: Addition

Our "multiplication principle" is not the only tool we have for counting things. Consider the following common-sense idea for how to count the elements in a set:

Observation. (**Summation principle**.) Suppose that you have a set A that you can write as the **union**[5]of several smaller disjoint[6] sets $A_1, \ldots A_n$.

Then the number of elements in A is just the summed number of elements in the A_i sets. If we let $|S|$ denote the number of elements in a set S, then we can express this in a formula:

$$|A| = |A_1| + |A_2| + \ldots + |A_n|.$$

[4]Often, in mathematical papers, when a mathematician defines some object via an equation, they will put the object being defined on the left, write := to draw attention to the fact that this is a definition, and put the quantity the object is being defined as on the right. So, if you were wondering why there a was a colon by the equals sign: this is why!

[5]Given two sets A, B, we denote their **union**, $A \cup B$, as the set containing all of the elements in either A or B, or both. For example, $\{2\} \cup \{\text{lemur}\} = \{2, \text{lemur}\}$, while $\{1, \alpha\} \cup \{\alpha, \text{lemur}\} = \{1, \alpha, \text{lemur}\}$.

[6]Sets are called **disjoint**if they haven no elements in common. For example, $\{2\}$ and $\{\text{lemur}\}$ are disjoint, while $\{1, \alpha\}$ and $\{\alpha, \text{lemur}\}$ are not disjoint.

We work one simple example, and one trickier example:

Question 1. *Pizzas! Specifically, suppose Pizza My Heart has the following deal on pizzas: for 7\$, you can get a pizza with any two different vegetable toppings, or any one meat topping. There are m meat choices and v vegetable choices. As well, with any pizza you can pick one of c cheese choices.*

How many different kinds of pizza are covered by this sale?

Solution. Using the summation principle, we can break our pizzas into two types: pizzas with one meat topping, or pizzas with two vegetable toppings.

For the meat pizzas, we have $m \cdot c$ possible pizzas, by the multiplication principle (we pick one of m meats and one of c cheeses.)

For the vegetable pizzas, we have $\binom{v}{2} \cdot c$ possible pizzas (we pick two different vegetables out of v vegetable choices, and the order doesn't matter in which we choose them; we also choose one of c cheeses.)

Therefore, in total, we have $c \cdot \left(m + \binom{v}{2}\right)$ possible pizzas!

Question 2. *Demonstrate the following equality:*

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

Solution. While it is not particularly obvious, we can do this with the summation principle!

We do this as follows: take the set $\{1, \ldots n\}$, and consider all of the k-element subsets[7] of this set. On one hand, there are $\binom{n}{k}$ many such subsets — this is because there are precisely $\binom{n}{k}$ many ways to pick out k things from a set of n things if we don't care about the order in which we pick things, and that is precisely what we are doing when we are finding k-element subsets.

On the other hand, let's break our subsets of $\{1, \ldots n\}$ into two cases:

1. The subsets that contain the element n. How many such subsets are there? Well: to create any such subset, we have to pick the element n, and then we have to pick $k-1$ more elements out of a set of $n-1$ total possible objects (to fill in the rest of the set.) But this just means that there are $\binom{n-1}{k-1}$ many such sets!

2. The subsets that **do not** contain the element n. How many such subsets are there? Well: to create any such subset, we have to pick k elements out of a set of $n-1$ total possible objects (because we need k things that are not n.) But this just means that there are $\binom{n-1}{k}$ many such sets!

By the rule of sum, because each subset of $\{1, \ldots n\}$ falls into exactly one of the two cases above, we can conclude that the total number of k-element-sized subsets of $\{1, \ldots n\}$ is just

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

[7] We say that a set A is a subset of a set B, and write $A \subseteq B$, if every element of A is an element of B. For example, $\{1, 2, \text{Batman}\}$ is a subset of $\{1, 2, 3, 4, 5, \text{Batman}, 7\}$.

Combining our two observations above gives us that

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

1.3 How to Count: Double-Counting

There is another really useful idea in the above example, that bears noting: to show that $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$, we actually looked at a third object (the collection of all k-element subsets of $\{1, \ldots n\}$, and showed that if you count it in one way you get $\binom{n}{k}$, and if you count it in another way you get $\binom{n-1}{k} + \binom{n-1}{k-1}$. Consequently, we claimed that these quantities are equal!

This idea, of **double-counting**, is as useful as our earlier observations:

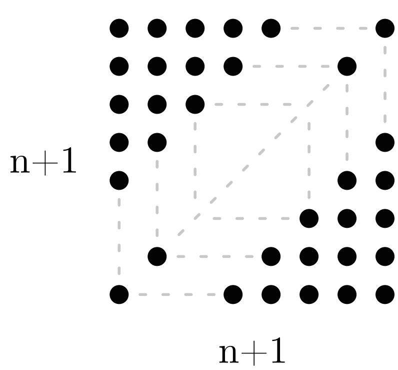
Observation. (**Double-counting principle**.) Suppose that you have a set A, and two different expressions that count the number of elements in A. Then those two expressions are equal.

We give one quick example of this in action:

Question 3. *Without using induction, prove the following equality:*

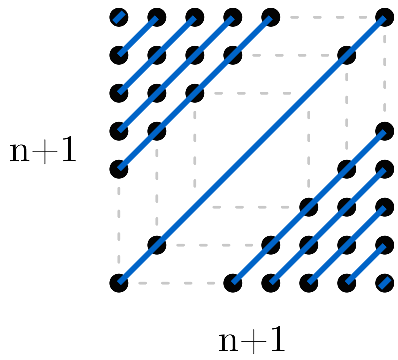
$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

Solution. First, make a $(n+1) \times (n+1)$ grid of dots:



How many dots are in this grid? On one hand, the answer is easy to calculate: it's $(n+1) \cdot (n+1) = n^2 + 2n + 1$.

On the other hand, suppose that we group dots by the following diagonal lines:



The number of dots in the top-left line is just one; the number in the line directly beneath that line is two, the number directly beneath that line is three, and so on/so forth until we get to the line containing the bottom-left and top-right corners, which contains $n+1$ dots. From there, as we keep moving right, our lines go down by one in size each time until we get to the line containing only the bottom-right corner, which again has just one point.

So, if we use the summation principle, we have that there are

$$1+2+3+\ldots+(n-1)+n+(n+1)+n+(n-1)+\ldots+3+2+1$$

points in total.

Therefore, by our double-counting principle, we have just shown that

$$n^2+2n+1 = 1+2+3+\ldots+(n-1)+n+(n+1)+n+(n-1)+\ldots+3+2+1.$$

Rearranging the right-hand side using summation notation lets us express this as

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

subtracting $n+1$ from both sides and dividing by 2 gives us finally

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

which is our claim!

1.4 How to Count: Recursion

In each of the above problems, the answers we got to,

$$k^n \quad , \quad \frac{k!}{(k-n)!} \quad , \quad c \cdot \left(m + \binom{v}{2} \right) \quad , \quad \frac{n(n+1)}{2} \quad ,$$

all looked like "good" answers for how to count the elements in our sets! This is because these answers were all **closed-form** expressions for the sizes of these sets: in other words, each of our answers above are short, simple expressions that we can calculate in a finite number of steps given values for our parameters.

However, not all questions obviously admit such solutions. Consider the following problem:

Question 4. *Suppose that we have a petri dish in which we're growing a population of amoebae, each of which can be in two possible states (**small** and **large**).*

Amoebas grow as follows: if an amoeba is small at some time t, then at time $t+1$ it becomes large, by eating food around it. If an amoeba is large at some time t, then at time $t+1$ it splits into one large amoeba and one small amoeba.

Suppose our petri dish starts out with one small amoeba at time $t = 1$. How many amoebae in total will be in this dish at time $t = n$, for any natural number n?

Answer. One approach to this problem is to use the technique of **recursion**. Roughly speaking, recursion, (or induction, depending on where you start) is the art of taking a larger problem and finding a way of breaking it into smaller instances of the same problem. For example, **multiplication** is a task that can be described recursively! Suppose you wanted to calculate $m \times n$, for two natural numbers m, n.

1. If $n = 0$, then $m \cdot n$ is 0. Otherwise, go to line 2.
2. If $n > 0$, notice that $m \cdot n = n + m \cdot (n-1)$. Therefore, it suffices to calculate $m \cdot (n-1)$. To do this, use $m, (n-1)$ as your inputs to line 1.

For example, we could recursively calculate that

$$\begin{aligned}
8 \cdot 5 = 8 + (8 \cdot 4) = 8 + (8 + (8 \cdot 3)) &= (8 + (8 + (8 + (8 \cdot 2)))) \\
&= (8 + (8 + (8 + (8 + (8 \cdot 1))))) \\
&= (8 + (8 + (8 + (8 + (8 + (8 \cdot 0)))))) \\
&= (8 + (8 + (8 + (8 + (8 + 0))))) \\
&= 40.
\end{aligned}$$

Recursion, or "divide and conquer,' is an incredibly useful and common tactic to solve problems in mathematics. It can often be easier to see how to break a problem into smaller cases than to see how to solve it all at once! Furthermore, if we can find some sort of a recursive relationship, we can often use this to get a full answer to our problem — it can be a lot easier to solve a few smaller or base cases for a problem rather than solving all of them at once!

Formally speaking, we can define[8] recursion as follows:

Definition. Suppose that we have an set $A(n)$ that we would like to count, that depends on some natural number n. (For example, in our amoeba problem above, $A(n)$ could be "the number of amoebae in our petri dish at time $t = n$.)

A **recurrence relation** for $A(n)$ is just an expression for $A(n)$ in terms of earlier-occurring terms $A(m)$. (For example, suppose that $A(n) = 2^n$. Then $A(n) = 2 \cdot A(n-1)$: this is a recurrence relation that defines $A(n)$ in terms of an earlier-occurring term.)

To give a slightly more interesting example of a recurrence relation: let's consider our amoeba problem! To help find any such recurrence relation, let's make a chart of our

[8] Informally speaking, we can define recursion with footnote 9.

amoeba populations over the first six time steps:

	1	2	3	4	5	6
Large	0	1	1	2	3	5
Small	1	0	1	1	2	3
Total	1	1	2	3	5	8

This chart lets us make the following observations:

1. The number of large amoebae at time n is precisely the total number of amoebae at time $n-1$. This is because every amoeba at time $n-1$ either grows into a large amoeba or already was an amoeba!

2. The number of small amoebae at time n is the number of large amoebae at time $n-1$. This is because the only source of small amoebae are the large amoebae from the earlier step when they split!

3. By combining 1 and 2 together, we can observe that the number of small amoebae at time n is the total number of amoebae at time $n-2$!

4. Consequently, because we can count the total number of amoebae by adding the large amoebae to the small amoebae, we can conclude that **the total number of amoebae at time** n is the total number of amoebae at time $n-1$, plus the total number of amoebae at time $n-2$. In symbols,

$$A(n) = A(n-1) + A(n-2).$$

This is a recurrence relation! Moreover, I claim that this is a **good answer** to our question earlier, about how many amoebae are in our petri dish.

However, when we look at an expression like k^n, we immediately know how to calculate this quantity — this is part of why we liked this as an answer to our earlier counting problems! Can the same be said for our recurrence-relation answer?

I claim that with a bit of thought, and in specific the concept of **induction**, the answer to this question is yes!

If it's been a while since Math 8, we quickly review induction here:

1.5 Review: Induction

Sometimes, in mathematics, we will want to prove the truth of some statement $P(n)$ that depends on some variable n. For example:

- $P(n) =$ "The sum of the first n natural numbers is $\frac{n(n+1)}{2}$."
- $P(n) =$ "If $q \geq 2$, we have $n \leq q^n$.
- $P(n) =$ "Every polynomial of degree n has at most n roots."

For any fixed n, we can usually use our previously-established methods to prove the truth or falsity of the statement. However, sometimes we will want to prove that one of these statements holds for **every** value $n \in \mathbb{N}$. How can we do this?

One method for proving such claims for every $n \in \mathbb{N}$ is the method of **mathematical induction**! Proofs by induction are somewhat more complicated than the previous two methods. We sketch their structure below:

- To start, we take our claim $P(n)$, that we want to prove holds for every $n \in \mathbb{N}$.
- The first step in our proof is the **base step**:in this step, we explicitly prove that the statement $P(1)$ holds, using normal proof methods.
- With this done, we move to the **induction step** of our proof: here, we prove the statement $P(n) \implies P(n+1)$, for every $n \in \mathbb{N}$. This is an implication; we will usually prove it directly by assuming that $P(n)$ holds and using this to conclude that $P(n+1)$ holds.

Once we've done these two steps, the principle of induction says that we've actually proven our claim for all $n \in \mathbb{N}$! The rigorous reason for this is the **well-ordering principle**, which we discussed in class; however, there are perhaps more intuitive ways to think about induction as well.

The way I usually think of inductive proofs is to think of toppling dominoes. Specifically, think of each of your $P(n)$ propositions as individual dominoes – one labeled $P(1)$, one labeled $P(2)$, one labeled $P(3)$, and so on/so forth. With our inductive step, we are insuring that all of our dominoes are *lined up* – in other words, that if one of them is true, that it will "knock over" whichever one comes after it and force it to be true as well! Then, we can think of the base step as "knocking over" the first domino; once we do that, the inductive step makes it so that all of the later dominoes also have to fall, and therefore that our proposition must be true for all n (because all the dominoes fell!)

To illustrate how these kinds of proofs go, here's an example:

Definition. The n-th **Fibonacci number** is defined recursively as follows:

- $f_0 = 0$.
- $f_1 = 1$.
- $f_n = f_{n-2} + f_{n-1}$.

Example. Here is the start of the Fibonacci number sequence:

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, \ldots$$

Notice that the Fibonacci sequence is precisely our amoeba-growth pattern from our earlier example!

The Fibonacci sequence satisfies a number of properties that can be demonstrated via induction. We study one such example here:

Claim. The n-th Fibonacci number is even iff n is a multiple of 3.

Proof. As noted above, the "recursive" construction of the Fibonacci numbers, where it looks like we're building our numbers out of earlier-defined numbers, suggests induction.

So: let's try! Our base case for a number of immediate cases is trivially true: we can check by hand that $f_0 = 0, f_1 = 1$, and $f_2 = 1$.

So: inductively, suppose we know that f_n is even if and only if n is a multiple of 3. Does this help us with f_{n+1}?

The answer is ... no, not really. Because

$$f_{n+1} = f_n + f_{n-1},$$

we really need more information than just f_n: we also need to know whether f_{n-1} is even or odd! Our normal induction isn't strong enough here; we need something... stronger. Like ... **strong** induction!

Strong induction proceeds very similarly to normal induction:

- To start, we take our claim $P(n)$, that we want to prove holds for every $n \in \mathbb{N}$.
- The first step in our proof is the **base step**:in this step, we explicitly prove that the statement $P(1)$ holds, using normal proof methods.
- With this done, we move to the **induction step** of our proof. For strong induction, we prove the statement that the combined statements $P(1) \land P(2) \land \ldots P(n)$ implies $P(n+1)$. If you go back to our domino analogy, this is like asking whether knowing that **every** domino from 1 to n falling over tells you that the $n+1$-th domino falls over.

So: if we're using strong induction, then we're allowed to assume that our inductive hypothesis holds for every value between 0 and n, and prove that it holds for $n+1$. We know that

$$f_{n+1} = f_n + f_{n-1};$$

so, it suffices to consider the three possible cases for $n+1$. Either $n+1$ is a multiple of 3, in which case we can write it as $3k$, or it is 1 mod 3, in which case it can be written as $3k+1$, or it's 2 mod 3, in which case we can write it as $3k+2$. In any case, we can use our inductive hypothesis to see that

$$\begin{aligned}
f_{3k} &= f_{3k-1} + f_{3k-2} = \text{odd} + \text{odd} = \text{even},\\
f_{3k+1} &= f_{3k} + f_{3k-1} = \text{even} + \text{odd} = \text{odd},\\
f_{3k+2} &= f_{3k+1} + f_{3k} = \text{odd} + \text{even} = \text{odd},
\end{aligned}$$

and thus that our property holds in any of these three cases. Thus, by induction, we know that f_n is even if and only if it is a multiple of 3. $\square$

1.6 How to Count: Recursion (again!)

The reason we mention induction here is that it's basically **recursion in reverse**. In one situation, we took a complicated problem and broke it down into smaller pieces; in the other, we took a base case and grew it up into larger cases!

In particular, the neat thing about induction is that it tells us how to count things we've measured recursively! Let's think about our amoeba problem once more.

- We know that $A(1) = 1$, by assumption, and that $A(2) = 1$, by looking at the data we collected earlier!
- We also know that $A(n) = A(n-1) + A(n-2)$.

I claim that by using induction, we actually know what $A(n)$ is for any n! This is a pretty silly, but honestly legitimate notion of how induction works:

- As a base case, we clearly know what $A(1)$ and $A(2)$ are, as demonstrated above.
- For an inductive step: assume that we "know" what $A(m)$ is, for every $m < n$. Then because $A(n) = A(n-1) + A(n-2)$, we also know what $A(n)$ is!

Therefore, by induction, our recurrence relation is actually a valid way to count!

It's not necessarily the fastest way to count, mind you: to find $A(n)$ with our formula above, we have to perform n addition operations, starting from our base cases. This is different than say a closed-form expression like k^n, which we can usually evaluate very quickly using clever algorithms for exponentiation. So, if you're programming, you may find yourself objecting to a recurrence relation as a "good" way to count. As well, it can be hard to tell from a glance how "fast" our recurrence relation is growing — unlike an expression like n^3, it's hard to tell roughly how big $A(n)$ will be for a given input n. This can cause issues with approximation problems, where we want to count something tricky but don't care about exactly what it is — a recurrence relation doesn't immediately tell us that information!

Later in this course, we'll talk about how to fix those two problems (because they are fixable!)

For now, however, let's look at a few more examples of recursion:

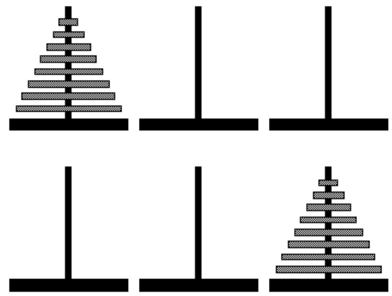
1.7 Recursion examples!

Example. The Towers of Hanoi is the following puzzle: Start with 3 rods. On one rod, place n disks with radii $1, 2, \ldots n$, so that the disk with radius n is on the bottom, the disk with radius $n-1$ is on top of that disk, and so on/so forth.

The goal of this puzzle is to move all of the disks from one rod to another rod, obeying the following rules:

- You can move only one disk at a time.
- Each move consists of taking the top disk off of some rod and placing it on another rod.

- You cannot place a disk A on top of any disk B with radius smaller than A.



What is the smallest number of moves needed to move all of our disks from the first peg to the third peg?

Answer. If you haven't played around with this puzzle before, take a break now and do so!

(waits)

Ok! I'm assuming now that you've done some basic work on this problem, and in particular worked some small cases out! Along the way, you've probably made the following observations:

1. If there is only one disk, we can do this in one move; if there are two disks, it takes us three moves; if there are three disks, it takes us seven moves, and if there are four disks, it takes us fifteen moves at best. (So, if you were to guess a pattern, $2^n - 1$ seems like a good bet!)

2. You can break down the task of moving a stack of n disks into three steps:

   (a) Move the first $n-1$ disks onto the second pole.

   (b) Move the n-th disk to the third pole.

   (c) Move the remaining $n-1$ disks on top of this n-th disk.

3. Furthermore, we know that all of these tasks are necessary: to move all of the disks to the third pole, we need to start by getting the n-th disk to the third pole. To do this, we need to get the first $n-1$ disks out of the way by putting them somewhere else (namely, the second pole!) Finally, when we have moved the n-th disk to the third pole, moving the remaining $n-1$ disks on top of that disk is exactly what it means to finish our puzzle.

These last two observations, when combined, give us a **recurrence relation**: if H_n is the total number of moves it takes to move a tower of n disks, we have just shown that

$$H_n = H_{n-1} + 1 + H_{n-1},$$

because we must move a tower of $n-1$ disks, move one disk, and then move that $n-1$ tower again!

Using this recurrence relation, we can prove our guess that $H_n = 2^n - 1$ via induction:

Base case: We know $H_1 = 1 = 2^1 - 1$ from our case work.

Ind. step: Assume that $H_n = 2^n - 1$ for each n from 1 to m; we will seek to prove that $H_{m+1} = 2^{m+1} - 1$. This is pretty quick: notice that

$$\begin{aligned} H_{m+1} &= H_m + 1 + H_m, \text{ by our recurrence relation,} \\ &= (2^m - 1) + 1 + (2^m - 1), \text{ by our inductive hypothesis,} \\ &= 2 \cdot 2^m - 1 \\ &= 2^{m+1} - 1, \text{ as claimed.} \end{aligned}$$

Example. Draw some straight lines in the plane. Notice that when we do this, we divide the plane up into regions bounded by these lines. What is the maximum number of regions we can divide the plane into with n lines?

Answer. Again, take a moment to work out some base cases and figure out what's going on here!

Here's a few observations you're likely to have made:

1. No lines break the plane into one piece, as we've not split anything up! One line breaks the plane into two pieces; two lines breaks the plane into up to four pieces if those lines are not parallel; three lines can break the plane up into seven pieces if we are careful to not let all three lines intersect at the same place; and four lines can break the plane up into up to 11 pieces if we are again careful to not have any more than two lines intersect at any point, and also not have any parallel lines! In general, it looks like n lines is giving us $1 + \frac{n(n+1)}{2}$ regions, given enough data and staring at things.
2. In general, it looks like the n-th line is adding at most n new regions to our plane. To see why this might hold in general, consider the process of drawing any line.
   (a) If our line intersects any region, it divides that region into two pieces! This is the only way our line creates new regions.
   (b) Our line enters a region if and only if it crosses one of the lines that bounds that region.
   (c) As well, before our line crosses any other regions, it by default starts in some region already.
   (d) Therefore, the total number of times our line intersects other lines, plus one, is the total number of new regions created!
   (e) We can cross each other line at most once, as our lines are straight.
   (f) Therefore, if there are n lines in existence, we can create at most $n + 1$ new regions by adding a $n + 1$-th line.
3. Furthermore, notice that it is always possible to draw such a line! To draw a line, we need to give two pieces of information:

(a) Its slope needs to not be parallel to any other existing line's slope, to insure that it can intersect that line. There are only n slopes currently used and infinitely many possibilities, so this is always possible.

(b) Given a slope, we need to pick a x-intercept for our line. Furthermore, we want to do this so that our line does not intersect any other lines at places where multiple lines are already intersecting: this would make our line use up multiple "intersecting other line" instances, while only entering one region (which means we wouldn't get to $n+1$!) There are only finitely many such existing intersection points, and infinitely many choices of x-intercept; so we can also avoid all of these possibilities.

Therefore, it is possible to always draw a line that intersects other lines in n places, and thus that creates $n+1$ new regions!

By the above, we have a rather nice recurrence relation: if L_n is the total number of regions that we can divide the plane up into with n lines, we have

$$L_{n+1} = (n+1) + L_n.$$

Using this, we can prove that our guess of $L_n = 1 + \frac{n(n+1)}{2}$ is right via induction:

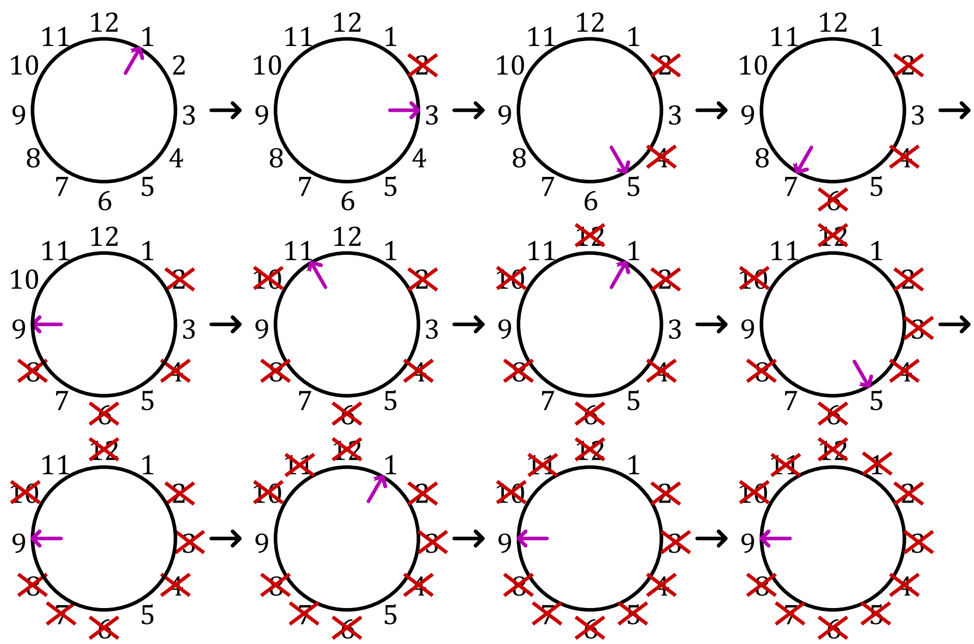
Base case: We know $L_0 = 1 = 1 + \frac{0 \cdot 1}{2}$ from our case work.

Ind. step: Assume that $L_n = 1 + \frac{n(n+1)}{2}$ for each n from 1 to m; we will seek to prove that $L_{m+1} = 1 + \frac{(m+1)(m+2)}{2}$. This is pretty quick: notice that

$$\begin{aligned} L_{m+1} &= (m+1) + L_m, \text{ by our recurrence relation,} \\ &= (m+1) + \frac{m(m+1)}{2} + 1, \text{ by our inductive hypothesis,} \\ &= \frac{m^2 + m + 2m + 2}{2} + 1 \\ &= \frac{(m+1)(m+2)}{2} + 1, \text{ as claimed.} \end{aligned}$$

We work one last example here: the Josephus problem! This is a problem with a lot of fascinating history behind it (see this Wikipedia article for a quick gloss); in the interest of space, we simply give the mathematical version here; a longer story is present in the homework!

Example. Take any natural number n, and write down the integers from 1 to n clockwise around a circle. Starting from 1, go clockwise around our circle and eliminate every second not-yet-eliminated number until only one number remains: we call this number the Josephus number for n, and denote it $J(n)$.



The Josephus system for $n = 12$. The purple arrow tells us where we currently are in the process, and the red cross-outs denote eliminating a number.

What is $J(n)$, for any n?

Answer. As in our last two examples, we start by making observations and collecting data. Take some time to check this problem out on your own! When you return, verify the following results:

1. We can calculate a small table for values of $J(n)$ by hand:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$J(n)$	1	1	3	1	3	5	7	1	3	5	7	9	11	13	15	1

   The pattern here can be hard to see; after some thought, however, it looks like our rule is the following: take any number n. Write $n = 2^m + l$, for some $0 \leq l \leq 2^m - 1$; we can do this for any number (why?) If we do this, then it appears that our Josephus numbers $J(n)$ are given by the equation $J(n) = 2l + 1$; it sets $J(2^m) = 1$ for every m, and gives us the odd-number-progressions that we're seeing in between all of these powers of 2!

2. When looking at the above table, we can notice that there are no even Josephus numbers! This is not too hard to see. If we are trying to find the Josephus number of $2k$, then after we have gone k steps into our game all of the even numbers are eliminated, and we have a circle with just the values $1, 3, 5, 7, 9, \ldots 2k - 1$ on it, where we're starting with 1.

   Similarly, if we're trying to find the Josephus number of $2k - 1$, then going $k - 1$ steps into our game gives us a circle with the values $1, 3, 5, 7, \ldots 2k - 1$, where we're starting at $2k - 1$; going one step more gives us a circle of the form $3, 5, 7, \ldots 2k - 1$ where we're starting at 3.

3. This gives us a recurrence relation! In fact, our first observation tells us that the Josephus number of $2k$ is related to the Josephus number of k, via the relation

$$J(2k) = 2J(k) - 1.$$

Similarly, our second observation tells us that the Josephus number of $2k-1$ is related to the Josephus number of $k-1$, via the relation

$$J(2k-1) = 2J(k-1) + 1.$$

(If you don't see why these relations are true, think about this for a while!)

We can use induction to prove our claimed pattern. First, notice that by induction we have $J(2^m) = 1$ for all m:

Base case: We know $J(2^0) = J(1) = 1$.

Ind. step: Assume that $J(2^n) = 1$ for each n from 1 to m; we will seek to prove that $J(2^{m+1}) = 1$. This is almost trivial;

$$\begin{aligned} J(2^{m+1}) &= 2J(2^m) - 1, \text{ by our recurrence relation,} \\ &= 2 \cdot 1 - 1, \text{ by our inductive hypothesis,} \\ &= 1, \text{ as claimed.} \end{aligned}$$

Now, suppose we're looking at $J(2^m + l)$. If we take l moves in this game, we now have a Josephus circle with 2^m entries on it, starting at the location $2l + 1$. We know that playing on this board, the surviving entry is precisely wherever we start, by our inductive proof above! Therefore, $J(2^m + l) = 2l + 1$, as claimed.

2 Generating Functions

All of the above methods are great tools for counting, and they are things we will rely on through the rest of this class. However, they are not exactly the founding blocks for a field; indeed, if this is all that there was to combinatorics, the field wouldn't exist!

To get an idea of how this gets stranger, consider one last postcard problem:

Problem. Suppose that we have just one postcard, and one friend. To actually physically mail out our postcard, we need to put stamps on it that total to k cents (for whatever value of k the country we're in charges to mail out cards.)

Suppose further that we're stamp collectors, and we have arbitrarily many stamps of every integer value in terms of cents. In how many ways can we apply k cents worth of stamps to our card, if we don't care about the order in which the stamps are attached?

To give a feel for this problem, let's consider some small cases. For $k = 1$, for example, there is exactly one way to do this:

1

If we have $k = 2$, then there are precisely two ways to do this:

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

If we have $k = 3$, then there are precisely three ways to do this:

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

Notice that we don't list $\textcircled{1}\textcircled{2}$ and $\textcircled{2}\textcircled{1}$ separately, as we don't care about the order in which we attach our stamps.

You might hope at this point in time that there are k ways to do this for any k. This (as often is the case with very small examples) false, as the next case demonstrates: for $k = 4$, we have five ways of paying for our postage.

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

You can check (do so, if you're curious!) that the following table holds for the first seventeen values of k, if we let $p(k)$ denote the number of ways to pay k-cents of postage with arbitrary stamps:

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
p(k)	1	1	2	3	5	7	11	15	22	30	42	56	77	101	135	176	231	297

At this stage, given that there aren't any discernible patterns in the above table that look a lot like a natural closed form, the next thing to do is probably to try to find a different way to approach this problem! In particular, we can reframe our problem not as the task of paying for a postcard with stamps, but as the mathematical task of writing a natural number k as the sum of other natural numbers, where we do not care about the order of summation. This is a well-studied problem that you have probably heard of before: we're looking for **partitions** of k!

If you haven't seen this before, here's the relevant definition:

Definition. Take any natural number k. A **partition** of k is any way to write k as a sum of positive integers. We think of two partitions as being the same if they contain the same terms, regardless of the order that they are written in. For any k, we let $p(k)$ denote the total number of partitions of k.

Another way to have came up with this reframing (if partitions aren't things you're used to) is to search online for the first few terms of our sequence, and see what comes up! For instance, searching for "$1, 1, 2, 3, 5, 7, 11, 15, 22, 30, 42, 56$" pulls up

https://oeis.org/A000041

from the OEIS (Online Encyclopedia of Integer Sequences), which helpfully defines this sequence as the sequence of partition numbers $\{a(n)\}_{n=0}^{\infty}$ (I don't know why they use $a(n)$ instead of $p(n)$.) They offer several other characterizations of this sequence:

- Also number of nonnegative solutions to $b + 2c + 3d + 4e + \ldots = n$ and the number of nonnegative solutions to $2c + 3d + 4e + \ldots \leq n$. - Henry Bottomley, Apr 17 2001

- $a(n)$ is also the number of conjugacy classes in the symmetric group S_n (and the number of irreducible representations of S_n).

- Also the number of rooted trees with $n+1$ nodes and height at most 2.

- Coincides with the sequence of numbers of nilpotent conjugacy classes in the Lie algebras $gl(n)$. A006950, A015128 and this sequence together cover the nilpotent conjugacy classes in the classical A,B,C,D series of Lie algebras. - Alexander Elashvili, Sep 08 2003

- $a(n) = a(0)b(n) + a(1)b(n-2) + a(2)b(n-4) + \ldots$ where b=A000009.

- Number of distinct Abelian groups of order p^n, where p is prime (the number is independent of p). - Lekraj Beedassy, Oct 16 2004

- Number of graphs on n vertices that do not contain P3 as an induced subgraph. - Washington Bomfim, May 10 2005

As is often the case when reading a reference article, like 50-75% of the references here will likely not make sense to you; that's OK! Just focus on the ones that seem useful, and wikipedia the others if desperate.

With this reframing, we're in more luck! You could look up partition numbers on Wikipedia, and do some google scholar / google books searching to find a **ton** of information about them; they've been widely studied! In fact, with sufficient work, you'd eventually find this stackexchange post linking to the following 2011 paper "Algebraic Formulas of the Coefficients of Half-Integral Weight Harmonic Weak Maass Forms," which notes that it is the first paper to find an expression for $p(n)$ that is a closed form! If you read the paper, however, you'll find that their "closed form" is a sum that takes half a page to express and an entire undergraduate education to unpack (take classes in complex analysis, number theory, topology, and abstract algebra for starters.)

This seems. . . like a bit too much work to send a postcard. I would claim that this is because a single closed form solution — as appealing as they are in cases where we get answers like $k!$ or $\binom{n}{k}$ — is often not the "correct" answer to a counting problem.

You could start trying to hunt from here for "other" ways to count our partition function, like a recurrence relation; these exist, and indeed on the homework you'll examine some of them!

In the next week's notes, we will introduce a **new** method of counting: the concept of **generating functions**! This will give us (amongst other things) a much more efficient way to answer our stamp problem!