Discrete Math Lecture 3

1 Counting

Combinatorics is the study of counting and enumerating objects and their arrangements. In computer science, many of the problems we study are combinatorial optimization problems, in the sense that we want to minimize or maximize some property of some discrete structure or search for a minimal or maximal such object. Combinatorics gives us the tools to analyze and solve these problems. In particular a lot of the techniques we see today will come up in your algorithms class. They will also be useful when we study discrete probability.

You will notice a surprising feature of some of the things we study today. Structures that seem to have definitions that have nothing to do with each other, especially in combinatorics, can end up being very closely related. You will see an example of this in the handout and this will keep coming up in mathematics you study in the future.

Counting is something that we all have some intuition for from real world experience. We will start with some intuitive examples and we will then formalize this intuition into a few useful rules.

Example 1. *I go shopping for clothes and want to buy a shirt, a pair of jeans and a pair of shoes. The store has* 6 *different kinds of shirts,* 5 *different kinds of jeans and* 3 *different kinds of shoes. How many distinct outfits can I construct out of these options?*

Intuitively, I have 6 choices of shirt, 5 choices of jeans and 3 choices of shoes, so there are $6 \times 5 \times 3 = 90$ distinct ways to pick one shirt, one jeans and one pair of shoes. More formally I have a set of shirts, S, a set of jeans J and a set of shoes H where $|S| = 6, |J| = 5, |H| = 3$, and it turns out the number of triples (s, j, h) where $s \in S, j \in J, h \in H$ is $|S| \times |J| \times |H| = 6 \times 5 \times 3 = 90$. Any such triple (s, j, h) is an element of the set $S \times J \times H$. This suggests the following rule.

Proposition 1 (Product rule)**.** *For finite sets* $A_1, A_2, \dots, A_n$

$$|A_1 \times A_2 \times \cdot \times A_n| = |A_1| \times |A_2| \times \cdot \times |A_n|$$

The following fact is very useful and we can prove it using the product rule

Proposition 2. *If a set S has n elements, it has 2^n distinct subsets*

Proof. To specify a subset we need to specify, for each element in S whether it is in or out of the subset. Furthermore, if we have specified this, then we have specified the subset completely. For each element $s \in S$, consider a set $\{0, 1\}$ where 1 corresponds to s being in the subset and 0 corresponds to not including S in the subset. Now we have one copy of the set $\{0, 1\}$ for each element in S. Call these sets $A_1, A_2 \dots, A_n$. A subset corresponds exactly to an element of $A_1 \times A_2 \times \cdots \times A_n$. By the product rule this is 2^n since each A_i has 2 elements. □

This is actually an example of *bijective counting*. Instead counting the size 1 subsets, the size 2 subsets and so on, we set up an exact correspondence between subsets of S and n tuples consisting of only the digits $0, 1$. It turned out to be easier to count the latter. This technique of counting one kind of thing by setting up a bijective mapping between those objects and a different set of objects that is easier to count, is very powerful and we will see more examples of it later today.

For now, let us return to another rule of counting that is even more intuitive than the product rule.

Example 2. *If I want to order either a sandwich or a salad for lunch and I go to a restaurant which serves* 5 *different sandwiches and* 3 *different salads, then how many lunch options do I have.*

The answer is just $5+3=8$. This corresponds to the sum rule below:

Proposition 3 (Sum rule)**.** *For finite sets $A_1, A_2, \ldots, A_n$ that are pairwise disjoint, i.e no two have any element in common, that is $\forall i \neq j, |A_i \cap A_j| = 0$*

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

A good thing to look out for is that when you have to choose something **and** something else, it is likely you want to apply the product rule and multiply things. When you want to pick something **or** something else, it is likely you want to add things and apply the sum rule. Of course don't blindly multiply when you see the word 'and' and add when you see 'or', but these can act as a useful hint towards what to do.

Example 3. *A password can be a sequence of any* 5 *digits or it can be a sequence of* 6 *capital English letters. How many possible passwords are there.*

By the product rule there are 9^5 sequences of 5 digits and 26^6 sequences of 6 English letters. By the sum rule, the total number of passwords is $9^5 + 26^6$.

Notice that it is important in the sum rule that the sets are disjoint.

Example 4. *I want to either get a sandwich for lunch or something that costs less than* 10 *dollars. The restaurant I visit has* 5 *sandwich options and in total has* 8 *menu items that are below* 10 *dollars in price, and only* 2 *of those are sandwiches. How many options do I have?*

Here we cannot just conclude that I have 5 options that are sandwiches and 8 options less than 10 dollars in price, so I have $5+8=13$ options in total. But this is not quite right! Some of those 8 options are actually sandwiches, and I already counted them among the sandwiches. In this example there are 2 sandwiches that are less than 10 dollars in price, and I have counted these twice, so I should reduce my total count by 2. Thus the number of options I actually have is $5+8-2=11$. This rule works for sets in general:

$$|A \cup B| = |A| + |B| - |A \cap B|$$

2 The inclusion-exclusion principle

But what if there are more sets? In particular let's say there are 3 sets that all intersect each other.

Proposition 4. *For any three finite sets A, B and C,*

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |B \cap C| - |C \cap A| + |A \cap B \cap C|$$

Proof.

$$|X \cup Y| = |X| + |Y| - |X \cap Y|$$

Replacing X with A and Y with $B \cup C$, and then repeatedly using the same formula, yields

$$\begin{aligned}
|A \cup B \cup C| &= |A \cup (B \cup C)| \\
&= |A| + |B \cup C| - |A \cap (B \cup C)| \\
&= |A| + (|B| + |C| - |B \cap C|) - (|(A \cap B) \cup (B \cap C)|) \\
&= |A| + |B| + |C| - |B \cap C| - (|A \cap B| + |B \cap C| - |(A \cap B) \cap (B \cap C)|) \\
&= |A| + |B| + |C| - |A \cap B| - |B \cap C| - |C \cap A| + |(A \cap B) \cap (B \cap C)| \\
&= |A| + |B| + |C| - |A \cap B| - |B \cap C| - |C \cap A| + |A \cap B \cap C|
\end{aligned}$$

$\square$

This suggests that we can actually use induction to prove the formula in general for n sets, which gives us the inclusion-exclusion formula.

Proposition 5. *For any sets $A_1, A_2, \ldots, A_n$:*

$$|\bigcup_{i=1}^{n} A_i| = \sum_{i=1}^{n} |A_i| - \sum_{1\le i<j\le n} |A_i \cap A_j| + \sum_{1\le i<j<k\le n} |A_i \cap A_j \cap A_k| - \cdots + (-1)^{n+1}|A_1 \cap A_2 \cap \cdots \cap A_n|$$

We can also prove this formula using DeMorgan's laws. We will only outline the idea for such a proof here. Lets first define a family of indicator functions. These functions are used to indicate if an element x is in a set or not. Thus for a set S, we can define $I_S(x) = 1$ if $x \in S$ and $I_S(x) = 0$ otherwise. Indicator functions are often a useful tool when it comes to counting things.

We can ask what is $I_{\bigcup_{i=1}^n A_i}(x)$? We can interpret indicator functions as Boolean valued functions (so $1 + 1 = 1$). Now we can write

$$I_{\bigcup_{i=1}^n A_i}(x) = I_{A_1}(x) + I_{A_2}(x) + \cdots I_{A_n}(x)$$

Applying DeMorgans's Law:

$$I_{\bigcup_{i=1}^n A_i}(x) = 1 - (1 - I_{A_1}(x))(1 - I_{A_2}(x)) \cdots (1 - I_{A_n}(x))$$

Convince yourself why this really is just an application of DeMorgan's Law. It is helpful to notice that $I_{S^c}(x) = 1 - I_S(x)$. Let us now abuse notation and write $I_S(x) = s$. Then the above formula becomes:

$$I_{\bigcup_{i=1}^n A_i}(x) = 1 - (1 - a_1)(1 - a_2) \cdots (1 - a_n)$$

But the RHS can be expanded to get

$$\sum a_i - \sum a_i a_j + \sum a_i a_j a_k - \cdots + (-1)^{n+1} a_1 a_2 \cdots a_n$$

This should remind you of the inclusion exclusion formula!

Excercise 1. *Fill in the details of the above proof sketch to derive the inclusion-exclusion principle. Notice that $I_{A\cap B}(x) = I_A(x) \cdot I_B(x)$.*

Let us now see an example of an application of the inclusion-exclusion formula:

Example 5. *A bit string is just a sequence of digits which consists only of 0's and 1's. How many bit strings of length exactly 8, start with 010 or end in 101?*

To solve this, notice that if we fix the first 3 digits, there are 5 more digits that can be either 0 or 1. So there are $2^5 = 32$ strings that start with 010 and similarly 32 strings that end with 101. Strings which start with 010 and end with 101 can have any 2 digits in between, so there are $2^2 = 4$ such strings. By inclusion-exclusion we obtain the number of bit strings of length exactly 8, that start with 010 or end in 101 to be $32 + 32 - 4 = 60$.

3 Permutations

A permutation, informally speaking, is a way to order a bunch of objects. This can be formalized by defining a permutation of a set A as a bijection from the set to itself. So if $A = \{1, 2, 3\}$, then one such permutation, f, might be $f(1) = 2$, $f(2) = 3$, $f(3) = 1$. Notice that we have mapped each and every element of A to a unique element of A. But how many ways are there of doing this?

For a 3 element set, we can literally count all permutations by hand. We can represent the permutation we just described as $(2, 3, 1)$. The list of all permutations is

$$(1, 2, 3); (1, 3, 2); (2, 1, 3); (2, 3, 1); (3, 1, 2); (3, 2, 1).$$

But as we get more and more objects, this soon gets out of hand. How many ways are there of ordering n objects? We can use the product rule to calculate this. Think of filling up a row of n blanks. There are n options to put in the first blank, $n-1$ options for the second and so on. This tells us that the number of ways of ordering n objects to be $n! = n \times (n-1) \times \cdots \times 2 \times 1$.

Let's see an example where this sort of thing might come up. Suppose you are a salesman who is selling products door to door. You plan to visit n cities and you would like to minimize the cost of travelling to these cities. The problem you have to solve is: what is the minimal cost route that allows you to visit all of these cities once, including your trip back?

We can try to solve this problem by checking each possible route. How many routes are there? This is the same as asking in which order we would like to travel to each city and how many total choices we have. So for n cities this is just $n!$. This doesn't seem so bad if you've only got four cities to go to, since $4! = 24$. However, if you're a bit more hardworking of a salesman and want to go to 8 cities, then you're looking at $8! = 40320$ different routes to think about. This is getting bad, but still somewhat manageable. But double the number of cities again, though, and we get $16! = 20922789888000$ which is way too many possible options to check even on a computer. Get up to a 100 cities and it turns out that $100!$ is a 157 digit long number. In fact $n!$ is approximately $\sqrt{2\pi n}\left(\frac{n}{e}\right)^n$; this is called Stirling's approximation.

This problem is the Travelling Salesman Problem, one of the most famous problems for which we do not have an efficient algorithm for solving. The problem dates back as far as the 1800s although it wasn't formally stated in its current form until around the 1930s.

But what if we didn't want to go to every city, but instead wanted to go to some subset of them. How many ways are there to do this? We can think about this informally: we carry out the same process as before, but stop when we reach our desired number of choices. This leads to the following definition and result.

Let $P(n, r)$ be the number of ways of listing some r elements out of n elements where the order of the elements matter. Using the exact same reasoning as before we argue that:

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

It is quite easy to do some algebra and conclude that

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

But let's prove this another way. Let's say we want to order n objects in a row. We have just discussed why there are $n!$ ways of doing this. But we could also count the number of ways of doing this as follows. First we order some r out these n objects and use these as the first r objects. There are, by definition, $P(n, r)$ ways of doing this. Then we are left with $(n-r)$ objects that can be ordered any way we like and there are $(n-r)!$ ways of doing this. We equate the results we got from counting the same thing in two different ways, resulting in

$$n! = P(n, r) \cdot (n-r)!$$

This gives us the result we want. It also illustrates a very useful technique that we will see a lot of. We can count the same thing in two different ways, and equate the two results, allowing us to derive all sorts of useful identities. This sort of argument is known as a combinatorial argument.

4 Combinations

While talking about permutations we cared about the order in which items appeared. $P(n, r)$ was the number of ways of listing some r elements out of n elements where the order of the elements matter. But sometimes we might care only about which items we select, the order in which they appear might not be of any significance.

$C(n, r)$, often written as $\binom{n}{r}$, is the number of ways of selecting r items out of n items, where the order of the items is of no significance.

Proposition 6.

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

Proof. We know that for every selection of r out of n objects, there are $r!$ different ways of ordering those objects, each of which results in a different permutation. Thus

$$\binom{n}{r} = \frac{P(n, r)}{r!}$$

which yields the desired result. □

Consider the following example. Suppose 7 drivers are trying to park in a parking lot where there are only 3 available spaces. How many different sets of drivers can be given a parking spot?

Notice that we are asking about the number of different sets of drivers that can park; we don't care which driver parks first, which second, and so on. Thus the answer is

$$\binom{7}{3} = \frac{7!}{3!4!} = \frac{7 \cdot 6 \cdot 5}{3 \cdot 2 \cdot 1} = 35$$

Excercise 2. *Solve the above problem but assume the parking lot has* 4 *available spaces. What do you notice?*

Observe that

$$\binom{7}{3} = \frac{7!}{3!4!} = \frac{7!}{3!4!} = \binom{7}{4}$$

We can in fact generalise the observation from the previous exercise.

Proposition 7. *For any natural number n and natural number, r such that $0 \leq r \leq n$*

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

Excercise 3. *Provide a combinatorial argument for this result.*

One useful way to think about counting objects in general is to imagine constructing/ specifying the object and count the choices you had to make along the way.

Example 6. *How many binary strings of length* 9 *have exactly* 3 *ones.*

Imagine constructing a binary string with this property. We can imagine 9 blank spaces. Notice that all we have to do to specify the string is to pick which 3 blank spaces are 1. Thus the answer is $\binom{9}{3} = \frac{9 \cdot 8 \cdot 7}{3 \cdot 2} = 84$

Excercise 4. *Show that the number of distinct ways of seating n people in a circle is* $(n-1)!$

5 Permutations and combinations with repetitions

Sometimes, rather than picking a bunch of things from a fixed set, we may want to choose some objects from a set of *types* of things.

Suppose you've been tasked with gathering six donuts and there is a choice of four types, say chocolate, plain, strawberry, and blueberry. How many different combinations can you make, with repetition? For an ordinary combination, we would only choose one of each type, but because we're concerned about classes of objects rather than single objects, we're able to choose multiples of a particular type.

Let's begin by considering one possible selection $CPBCCB$ (three chocolate, one plain, two blueberry), assuming this is the order (chocolate, then plain, then blueberry, then chocolate and so on) in which we chose our goods. However, since this is a combination and some of the elements are indistinguishable anyway, the order doesn't really matter, so let's group them together into $CCCPBB$. Now, let's separate these so they don't touch each other and cross contaminate the flavours or something, and we have something that looks like $CCC|P|BB$.

We can play with this analogy further and suppose that the box we have has a compartment for each type, regardless of the number that we end up choosing, so we have something like $CCC|P||BB$ with an empty compartment for the strawberry donuts. Finally, we note that since each compartment contains a specific type, we don't need to specifically denote the type, and we can represent our choice by $***|*||**$.

Let's consider another possible choice: $*||*|****$, which is one chocolate, one strawberry, and four blueberry. What we observe is that each choice of six items from four classes can be represented by an arrangement of six stars representing the items and three bars representing the division of classes of items.

But now we just have to count how many possible strings are there with exactly six $*$'s and three $|$'s. We have done this before! There are 9 spaces to fill and we need to select which 3 will be filled by bars, so the answer is

$$\binom{9}{3} = \frac{9 \cdot 8 \cdot 7}{3 \cdot 2} = 84$$

This method of using stars and bars to denote the objects and categories was popularized by William Feller's An Introduction to Probability Theory and its Applications in 1950.

In fact we have the following result which you can prove via a 'stars and bars' style argument.

Excercise 5. *There are $\binom{n+r-1}{r-1}$ distinct ways of choosing n objects from a set of r distinguishable objects, where you are allowed to choose any given object as many times as you like.*

The 'stars and bars' method is really an example of bijective counting. We set up a bijection between arrangements of certain kinds of objects and strings of 'stars and bars.' Then we could just count the strings instead. We will see more examples of this later today.

We can also generalize permutations in this way:

Example 7. *How many distinguishable permutations of the word TOMATO are there?*

Here we cannot simply argue that we have a six letter word and we want to count how many ways to order all six of it's letters are, so the answer should be $P(6, 6) = 6!$.

This is because the two T's in the word, for example cannot be distinguished from each other. W e can instead reason as follows. There are 6 blank slots, We need to first pick 2 slots for the T's. That can be done in $\binom{6}{2}$ ways. Then of the remaining 4 slots, we need to pick 2 for the O's. That can be done in $\binom{4}{2}$ ways. Next, of the remaining 2 slots we need to pick 1 for the M and the last slot goes to the A, Thus the total number of ways is:

$$\binom{6}{2}\binom{4}{2}\binom{2}{1}\binom{1}{1} = \frac{6!}{2!2!1!1!} = 180$$

It is interesting we obtained $\frac{6!}{2!2!1!1!}$. This suggests the following alternative line of reasoning. Suppose all the letters were distinct. Then there would be 6! different permutations. But two T's are indistinguishable so we have over counted by a factor of 2!, because for each legitimately distinct ordering, we counted an extra ordering, obtained by swapping the positions of the T's. Similarly we over counted by a factor of 2! due to the two O's being indistinguishable. Thus, the answer is $\frac{6!}{2!2!}$.

This generalizes into the following result, the proof is left as an exercise (follow the logic of the example above).

Proposition 8. *The number of different permutations of n objects, where there are n_i indistinguishable objects of type i for $1 \leq i \leq k$ is*

$$\frac{n!}{n_1!n_2!\dots n_k!}$$

6 Distinguishable and Indistinguishable things

Notice the difference between counting distinguishable and indistinguishable objects. Suppose we want to count the number of ways to put 10 indistinguishable apples into 3 different baskets, where the baskets can be distinguished in some way. Maybe they look different, or they are arranged in a line and there is a notion of the first basket, second basket and so on. In this scenario, the number of ways of distributing the apples into baskets is $\binom{12}{2}$, which you can verify using the stars and bars method that we already discussed (here we are assuming you can leave some baskets empty).

On the other hand if you can't tell the difference between the baskets, that is to say we treat the divisions $(3, 3, 4)$ and $(4, 3, 4)$ as the same, then the answer is quite different! (You can try to figure out what it is, this Wikipedia Page, especially the recurrence in the section on Restricted part size or number of parts, may be of some interest.)

7 The binomial theorem

At this point, work through the handout provided for lecture 3.

As you may have noticed, the coefficients that appear in the expansion of $(a + b)^n$ are exactly $\binom{n}{i}$ for $i = 0$ to n. This is why $\binom{n}{i}$ are also called the binomial coefficients. But why should such a thing be true? What does the expansion of a binomial have to do with choosing objects from sets?

Suppose I want to know the coefficient of a^3b^2 in $(a + b)^5$. Let's write

$$(a + b)^5 = (a + b)(a + b)(a + b)(a + b)(a + b)$$

For a moment resist the impulse to expand this out. Instead, notice that when we do this product, how many times will a^3b^2 appear. We have 5 brackets, and from each we must pick an 'a' or a 'b'. We will have to pick exactly 3 a's and 2 b's. But how many ways are there of doing this? All we have to do is decide which of the brackets we will select the b's from, from the others we will take a's. Each way of doing this will produce a a^3b^2 term in the product. Thus the coefficient of a^3b^2 is exactly $\binom{5}{2}$.

This idea immediately leads us to the binomial theorem

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

Excercise 6. *We can also prove the binomial theorem by induction, which we leave as an exercise for the reader.*

Let us immediately use this theorem to prove something about the binomial coefficients:

Proposition 9.

$$\sum_{i=0}^{n} \binom{n}{i} = 2^n$$

Proof.

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

Set $a = b = 1$. □

Instead we could also have proved this identity by a combinatorial argument.

Proof. Consider the number of distinct subsets of an n element set. We already discussed why this should be 2^n. But instead, we could count the distinct subsets by adding together the sums of the number of 0 element subsets, 1 element subsets, 2 element subsets and so on. This count is $\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n}$. This observation completes the proof. □

Excercise 7. *Prove that*

$$\binom{n}{0} + \binom{n}{2} + \binom{n}{4} \cdots = \binom{n}{1} + \binom{n}{3} + \binom{n}{5} \cdots$$

We can in fact extend this idea to expanding trinomials such as $(x+y+z)^5$ and more generally to k-nomials. These expansions result in the multinomial coefficients, for more details see the Honors question in hw 3.

8 Pigeonhole principle

We have already seen a statement of the pigeonhole principle, we will restate it below

Proposition 10. *If there are $k > 0$ pigeons housed in strictly less than k pigeonholes, then there will be at least one pigeonhole with multiple pigeons in it.*

The principle is extraordinarily simple and kind of obvious. But it has some truly astonishing applications. The tricky part in any application, is what to define as the pigeons and what to define as the pigeonholes. Often the key to using this principle is to realise that it can be used for a given application in the first place. Let's see some examples.

Excercise 8. *In any cocktail party with two or more people, there must be at least two people who have the same number of friends. Assume that friendship is symmetric, If A is friends with B then B is friends with A.*

Let the people be pigeons and the number of friends they have as the pigeonholes. Assume that no one has 0 friends. Then the possible number of friends are 1 through n, i.e. there are $n-1$ possibilities or pigeonholes but n people (pigeons). Thus two of them must have the same number of friends. Work out what happens when someone has 0 friends.

Proposition 11. *Lossless data compression for all possible files is impossible.*

Let us model files as strings of 0's and 1's. What we are saying is that we cannot compress arbitrary n bit length files into files of any smaller length in a way that we can always recover the original file from the compressed file. Imagine there is a function f that maps all n bit binary strings to $n-1$ bit binary strings. Let the n bit strings be pigeons and $n-1$ bit binary strings be pigeon holes. What can we say about such a function and why does that imply that lossless data compression for arbitrary files is impossible?

Note that the number of n bit binary strings is 2^n while the number of $n-1$ bit strings is 2^{n-1}. Thus each original file cannot be mapped to a distinct compressed version. Thus such a mapping will not be injective. What happens if we map multiple files to the same compressed version? We won't always be able to get back the correct original.

But you might protest, image compression seems to work so well. The point is that you are not trying to compress arbitrary files, but rather files that have a lot of structure which can be exploited to represent the information more compactly.

Let's see another, slightly more complex application.

Proposition 12. *Among six people, where each pair are either friends or enemies, there exist either* 3 *mutual friends or* 3 *mutual enemies.*

The idea for the proof is as follows. Consider one person Alice and if she is friends or enemies with the other 5 people. By the pigeonhole principle either there are at least 3 people she is friends with or 3 people she is enemies with. Suppose there are 3 people she is friends with (the other case is exactly similar). Call these people Bob, Charlie and Eve. Either some two of these people are friends and then they along with Alice are a group of 3 mutual friends, or else no two of them are friends, they form a group of 3 mutual enemies.

This is a result from Ramsey theory. All of this can be stated more formally using graphs and cliques and colourings, but the party setting allows us to defer defining all of these things until we actually get to graph theory. Note that this argument hinges on the fact that there are at least 5 other people connected to Alice in order for us to apply the pigeonhole principle. If we had fewer (i.e. if we took groups of 5 or less), we wouldn't be able to guarantee either mutual friend/stranger triplet.

We can generalise the pigeonhole principle slightly

Proposition 13. *If there are n pigeons in k pigeonholes the there is a pigeonhole with at least $\lceil \frac{n}{k} \rceil$ pigeons.*

9 Exchanging the order of summation

This is another very simple idea that ends up being used in a bunch of applications. We begin with an illustration.

Consider the problem of summing a collection of numbers that have been doubly indexed:

$$\sum_{i,j} a_{i,j}$$

For example if i and j both take values among $\{1,2,3\}$, the aforementioned sum becomes

$$\sum_{i,j} a_{i,j} = a_{1,1} + a_{1,2} + a_{1,3} + a_{2,1} + a_{2,2} + a_{2,3} + a_{3,1} + a_{3,2} + a_{3,3}.$$

We'd like to find simple ways of organizing and computing sums of this type. First, notice in our example:

$$\sum_{i=1}^{3}\left(\sum_{j=1}^{3} a_{i,j}\right) = (a_{1,1}+a_{1,2}+a_{1,3})+(a_{2,1}+a_{2,2}+a_{2,3})+(a_{3,1}+a_{3,2}+a_{3,3}) = \sum_{i,j} a_{i,j}.$$

Similarly, $\sum_j \sum_i a_{i,j}$ gives the same result. This simple observation for finite sums comes from the commutativity and associativity of addition, but it has far reaching consequences. From now on we drop the comma in doubly indexed sequences, writing $a_{i,j}$ instead as a_{ij}.

Theorem 1. *Given a finite sum indexed by i and j we have*

$$\sum_{i,j} a_{ij} = \sum_i \left(\sum_j a_{ij}\right) = \sum_j \left(\sum_i a_{ij}\right)$$

We omit the proof, which is merely uses induction on the size of the sum and basic properties of addition. Here is a simple and well-known application, sometimes called the handshake lemma.

Theorem 2. *In a room of people, some pairs shake hands and some don't. No two people shake hands more than once and nobody shakes her own hand. Given a person p, let $n(p)$ denote the number of hands p shook. If the total number of handshakes is H, then*

$$\sum_p n(p) = 2H$$

Proof. Given a person p and handshake h let I_{ph} be 1 if person p participated in handshake h and 0 otherwise. This is another example of an indicator function. If we fix a person p and sum $\sum_h I_{ph}$, we obtain the number of hands p shook. That is,

$$\sum_h I_{ph} = n(p)$$

If we fix a handshake h and sum $\sum_p I_{ph}$ we obtain 2 , the number of people involved in any single handshake. This gives

$$\sum_p n(p) = \sum_p \sum_h I_{ph} = \sum_h \sum_p I_{ph} = \sum_h 2 = 2H$$

as desired. □

As a corollary, if 5 people in a room each claim to have shaken 3 hands then someone is lying - the number 15 is not even.

Note that we can only do this sort of exchange of the order of summation in such a carefree way when dealing with finite sets. When the sets we are summing over become infinite, this may not work. However we will not run into these situations in this course. Still it is good to keep this in mind. You can try to investigate what happens in the following example

Example 8. *Let's define a_{jk} for positive integers j, k. Whenever $j = k$ set $a_{jk} = 1$ and whenever $j = k+1$ let $a_{jk} = -1$. Otherwise, let $a_{jk} = 0$.*

For any k we have $\sum_j a_{jk} = 0$ and it follows that

$$\sum_k \sum_j a_{jk} = 0$$

When $j \geq 2$ we have $\sum_k a_{jk} = 0$, but when $j = 1$ we have $\sum_k a_{jk} = 1$. It follows that

$$\sum_j \sum_k a_{jk} = 1 \neq \sum_k \sum_j a_{jk}.$$

This phenomenon can only happen for infinite sums with both positive and negative terms.

In the previous example we only examined sums whose indices varied independently. There are many situations where the indices are constrained by each other. The sum-switching principle is no different in this case; rather, we need to learn how to describe the indices in multiple ways.

Example 9. *Sum the series*

$$\sum_{i=0}^{n} \sum_{j=i}^{n} \binom{j}{i}$$

We understand sums of the form $\binom{m}{0} + \binom{m}{1} + \binom{m}{2} + \cdots$ better than the sum written here. It would thus be advantageous to reverse the order and sum over i first. However, blind manipulation yields

$$\sum_{j=i}^{n} \sum_{i=0}^{n} \binom{j}{i}$$

which is meaningless! The "answer" to our problem would depend upon a dummy variable i, which makes no sense. Instead, we revert to the double sum:

$$\sum_{i=0}^{n} \sum_{j=i}^{n} \binom{j}{i} = \sum_{0 \leq i \leq j \leq n} \binom{j}{i}$$

and try to describe the double sum in the desired order. We want i on the 'inside' sum, so let's decide the bounds of j first. Ignoring i temporarily, we see that j varies from 0 to n. At a fixed j between 0 and n (which is how the inner summation sees j), the variable i can be as small as 0 and as large as j. That is,

$$\sum_{i=0}^{n} \sum_{j=i}^{n} \binom{j}{i} = \sum_{j=0}^{n} \sum_{i=0}^{j} \binom{j}{i}.$$

We know the inner sum is merely 2^j as we have already discussed previously today. We compute

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

as desired.

Notice that we used the fact that $\sum_{j=0}^{n} 2^j = 2^{n+1} - 1$. This is a very useful fact which we leave as an exercise to prove by induction.

10 More combinatorial arguments

We provide a couple of more examples of combinatorial arguments. Recall that the idea here is to count the same thing in two different ways and then equate these in order to derive the required result. The creativity is in coming up with the right thing to count. this method is also somethimes called "double counting" or "counting two ways"

Let's look at another curious thing you might have observed while working on your handout. The numbers in the n-th row of Pascal's triangle are exactly the same as the binomial coefficients $\binom{n}{k}$. But we obtained these numbers in the triangle just by adding two numbers in the previous row. If $p(i,j)$ is the j-th number in row i of Pascal's triangle, then Pascal's numbers are defined by the simple recursive formula (and appropriate base cases):

$$p(i,j) = p(i-1,j-1) + p(i-1,j)$$

That is to get a number in the triangle you just add the number directly above that position to the number above and one space to the left of the position. But as we have noticed, these numbers are exactly the binomial coefficients, which are exactly $\binom{n}{k}$. These three things seem completely different: one is the outcome of a simple linear recurrence, the second has to do with coefficients in polynomial identities and the third is the number of ways of selecting things and the third and it's formula is in terms of ratios of factorials!

We have already seen why $\binom{n}{k}$ are the binomial coefficents when we discussed proving the binomial theorem. If you try to prove the binomial theorem inductively, the connection to Pascal's numbers becomes clearer. In fact it turns out that

Proposition 14. *(Pascal's identity)*

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

where n and k are natural numbers so that the above terms are well defined.

Notice the similarity to the recurrence for the numbers in Pascal's triangle! We can of course verify this algebraically, but we will demonstrate a combinatorial proof.

Proof. (Sketch) The LHS is just the number of ways of selecting k objects out of n distinct objects. But now let us fix one particular object A. There are two cases

- A is selected. Now we just have to choose $k-1$ objects from the remaining $n-1$ options. There are $\binom{n-1}{k-1}$ ways to do this.
- A is not selected. Now we still have to choose k objects but have only $n-1$ options left. There are $\binom{n}{k-1}$ ways to do this.

Summing these two terms together gives us the RHS, which concludes the proof. □

This is exactly the sort of reasoning you will do while writing recurrences to solve problems via dynamic programming, which is something you will see in your algorithms class.

We demonstrate a combinatorial argument for another identity

Proposition 15. *(Vandermonde's identity) For natural numbers n,m and r*

$$\binom{n+m}{r} = \sum_{k=0}^{r} \binom{n}{k}\binom{m}{r-k}$$

Proof. Sketch Suppose there are $n+m$ soldiers. n of them are part of the infantry while the remaining m of them are part of the cavalry. Now you want to select a team of r soldiers composed of any mix of infantry and cavalry. How many ways are there to do this?

One way to count this is to ignore the infantry/cavalry designations and say that I just want to pick r soldiers out of a total of $m+n$. This leads to the LHS. But there is another way to count them. We can consider the cases of choosing 0,1,2 etc soldiers from the infantry. In general if you want to pick k soldiers from the infantry, there are $\binom{n}{k}$ ways of doing that and then you still have to pick the rest, i.e, $r-k$ soldiers from the cavalry and there are $\binom{m}{r-k}$ ways of doing that. All these cases are disjoint, so by the sum rule we get the total number of ways of forming the team to be the RHS as well. □

11 Bijective counting

Bijective counting is an extremely powerful technique where instead of counting elements of some set, A, which for some reason or the other are hard to count, we prove that there is a bijection between this set A and another set B. Then we count the elements of B and conclude that this is the number of elements in A. Recall that a bijection from A to B is a function that

- maps every single element of A to a distinct element of B (called as being 'one-to-one' or injective)
- maps some element of A to every single element of B (also called being 'onto' or surjective).

Bijections between sets of things, in a certain sense, tell us that we can see the objects in one set as corresponding exactly to objects in the other set, which can be very useful information even beyond just for the sake of counting these objects. One application of bijections that you will surely see while studying about algorithms is the notion of a *reduction*. Here we try to solve one algorithmic problem by instead converting our problem to a different algorithmic problem which we know how to solve and instead solving that. We want to have a bijection between solutions of our original problem and solutions of the new problem.

The notion of bijections works for infinite sets as well. In fact that is a way to define when two infinite sets have the same size, exactly when there is a bijection between them. For example it's easy to construct a bijection between natural numbers and even numbers ($f(n) = 2n$) even though it might feel like there are obviously more natural numbers than even natural numbers. But be warned, just because two sets are infinite does not mean that you can construct a bijection between them. For example Cantor, with a very elegant diagonalization argument, showed that there can be no bijection between real numbers and natural numbers, it turns out that the number of real numbers is in this sense a bigger infinity than the number of natural numbers. And, what's more, there are no infinities of size strictly bigger than natural numbers but strictly smaller than the real numbers! However we won't go into details in this class.

We will illustrate counting via bijections with a couple of examples:

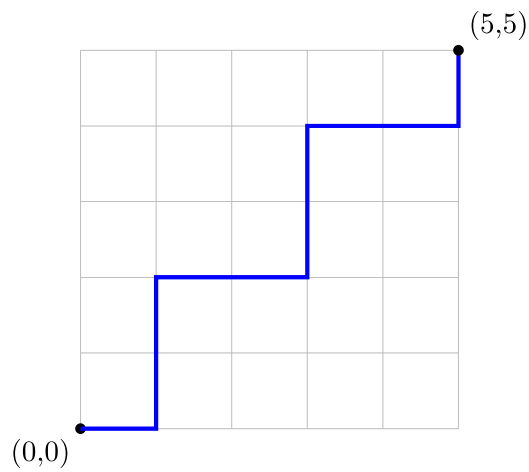
Example 10. 1500 *teams compete in a soccer tournament. The organizers let the teams know that every game must have a winner and that the team that loses a game is immediately excluded from the tournament. How many games will be played till the champion is known?*

You might initially suspect that this might depend on how the pairings are done. However note that the very loser lost in some game and every game produces a loser. Thus there is a bijection between games played and losers produced. This immediately reveals that no matter how we pair teams up, 1499 games will have to be played to get a champion.

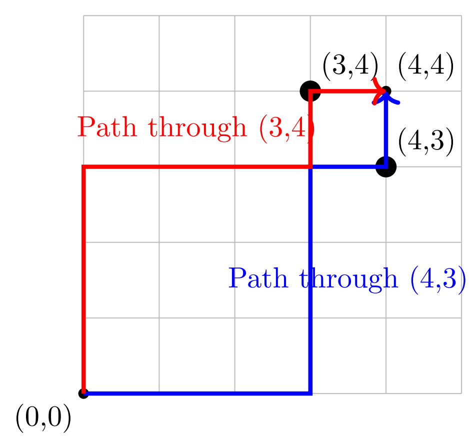
In fact a great many of the arguments we made today can be thought of as examples of bijective counting. For example the 'stars and bars' method also consists of establishing a bijection between the arrangements of various objects and strings of stars and bars.

Let's now consider a slightly more complex example.

Example 11. *Consider a $n \times n$ grid, with $(0,0)$ being the bottom left corner and (n,n) the top right corner. We start at $(0,0)$ and at each step move one unit to the right or one unit upwards. How many distinct paths are there from $(0,0)$ to (n,n).*



Let us start by writing down the number of ways to get from $(0,0)$ to each lattice point. If you do this, you might be surprised to see the numbers from Pascals triangle start to emerge! But a little more reflection makes this less mysterious: after all the number of ways to get to (i,j) is just the number of ways of getting to $(i-1,j)$ plus the number of ways of getting to $(i,j-1)$, since the only way to get to a lattice point is from the point below it or from the point to the left of it. This is starting to look very similar to the definition of Pascal's numbers!



We could continue reasoning in this manner to get answer. But instead let's reason differently. Each path must consist of n steps upward and n steps to the right. Furthermore, if I tell you the order in which these up and right steps are made, that completely fixes the path and indeed every path can be represented in this way. But we can represent a sequence of up and right moves with a $2n$ length string of the letters u and r. Let us consider a function that maps a path to the corresponding sequence of u's and r's. We claim that this is a bijection. We know how to count such strings: we just need to choose which of the $2n$ positions in the string will be u. There are

$\binom{2n}{n}$ ways of making this choice, so that is the number of paths!

To formally prove that this function is a bijection we need to show two things:

- No two different paths get mapped to the same string.
- Every string of this form can be arrived at by applying the function to some valid path.

When ever you are doing an argument based on bijections, in particular bijective counting, to be completely formal you must prove that the function you have defined is indeed a bijection. You can do this by checking the two conditions we stated above.) In more general terms these are:

To show that f is a bijection from A to B, show that

- There are no two distinct $a_1, a_2 \in A$ such that for some $b \in B$, $f(a_1) = f(a_2) = b$.
- For every $b \in B$ there is an $a \in A$ such that $f(a) = b$.

Another useful fact is that if you only know that there is an f satisfying the second condition, then for finite sets, this is enough to conclude that $|A| \geq |B|$.

The last Honors question in this week's HW considers a variation of the paths problem we just discussed: how many such paths are there that do not cross the line $y = x$. Here the bijection is trickier. This variation of the problem gives rise to the Catalan numbers which come up in a huge number of contexts and can be defined in a large number of equivalent ways. This points to just one example where there are bijections between a wide array of combinatorial structures that might seem very different at first sight. Recall how the binomial coefficients, the number of combinations and Pascal's numbers turned out to be the same things. This sort of phenomenon, where you discover surprising connections between things that superficially seem like they have nothing to do with each other, is a recurrent theme in combinatorics, and in fact in math in general.

Excercise 9. *On a 7×7 grid, how many such paths are there from $(0, 0)$ to $(7, 7)$ that pass through $(3, 4)$?*