1. Pigeon-hole principle

1.1. Basic version. The following is really obvious, but is a very important tool. The proof illustrates how to make “obvious” things rigorous. It is important to always keep this in mind especially in this course when many things you might want to use sound obvious. There are many interesting ways to use this theorem which are not obvious.

**Theorem 1.1** (Pigeon-hole principle (PHP)). Let \( n, k \) be positive integers with \( n > k \). If \( n \) objects are placed into \( k \) boxes, then there is a box that has at least 2 objects in it.

**Proof.** We will do proof by contradiction. So suppose that the statement is false. Then each box has either 0 or 1 object in it. Let \( m \) be the number of boxes that have 1 object in it. Then there are \( m \) objects total and hence \( n = m \). However \( m \leq k \) since there are \( k \) boxes, but this contradicts our assumption that \( n > k \). \( \square \)

Note that the objects can be anything and the boxes don’t literally have to be boxes.

**Example 1.2.**  
- Simple example: If we have 4 flagpoles and we put up 5 flags, then there is some flagpole that has at least 2 flags on it.  
- Draw 10 points in a square with unit side length. Then there is some pair of them that are less than \( .48 \) distance apart. There’s some content here since the corners on opposite ends have distance \( \sqrt{2} \approx 1.4 \). Also, if we only have 9 points, we could arrange them like so:
  
  \[
  \begin{array}{cccc}
  \cdot & \cdot & \cdot & \\
  \cdot & \cdot & \cdot & \\
  \cdot & \cdot & \cdot & \\
  \end{array}
  \]

  The pairs of points that are closest are \( .5 \) away from each other, so it is important that we have at least 10 points.

  To see why the statement holds, divide the square into 9 equal parts:

  \[
  \begin{array}{|c|c|c|}
  \hline
  & & \\
  & & \\
  \hline
  
  \end{array}
  \]

  Then some little square has to contain at least 2 points in it (is it ok if the points are on the boundary segments?). Each square has side length \( 1/3 \), and so the maximum distance between 2 points in the same square is given by the length of its diagonal (why?) which is \( \sqrt{(1/3)^2 + (1/3)^2} = \sqrt{2}/3 \approx 0.4714 \). \( \square \)
Here are some more to think about:

- At least 2 of the students in this class were born in the same month.
- If you have 10 white socks and 10 grey socks, and you grabbed 3 of them without looking, you automatically have a matching pair.
- Pick 5 different integers between 1 and 8. Then there must be a pair of them that add up to 9.
- Given 5 points on a sphere, there is a hemisphere that contains at least 4 of the points.
- There is a party with 1000 people. Some pairs of people have a conversation at this party. There must be at least 2 people who talked to the same number of people.
- Given an algorithm for compressing data, if there exist files whose length strictly decreases, then there exist files whose length strictly increases!

In mathematical terms: let’s represent a file by a sequence of 0’s and 1’s. Then an algorithm for compressing data can be thought of as a function that takes each sequence to some other sequence in such a way that different inputs must result in different outputs.

1.2. **General version.** Here’s a more general version of the PHP:

**Theorem 1.3** (General pigeon-hole principle). Let \( n, m, r \) be positive integers and suppose that \( n > rm \). If \( n \) objects are placed into \( m \) boxes, then there is a box that contains at least \( r + 1 \) objects in it.

If you set \( r = 1 \), then this is exactly the first version of the PHP.

**Proof.** We can again do this via proof by contradiction. Suppose the statement is false and label the boxes 1 up to \( m \). Let \( b_i \) be the number of objects in box number \( i \). Then \( b_i \leq r \) since the conclusion is false. Furthermore, we have \( n = b_1 + b_2 + \cdots + b_m \leq r + r + \cdots + r = rm \).

But this contradicts the assumption that \( n > rm \). \( \square \)

**Example 1.4.**

- Simple example: If we have 4 flagpoles and 9 flags distributed to them, then some flagpole must have at least 3 flags on it.
- Continuing from our geometry example from before: draw 9 points in a square of unit side length. Then there must be a triple of them that are contained in a single semicircle of radius 0.5. (Is this true if we only have 8 points?)

  For the solution, we divide up the square into 4 triangles as follows:

  ![Triangle Division](image)

  Then some triangle must contain at least 3 points. Furthermore, each triangle fits into a semicircle of radius 0.5. \( \square \)

In a crude sense, Ramsey theory is the natural generalization of PHP (we likely won’t discuss it, see Chapter 13 for a start if you’re interested).

2. **Induction**

Induction is a proof technique that I expect that you’ve seen and grown familiar with in a course on introduction to proofs. We will review it here.
2.1. **Weak induction.** Induction is used when we have a sequence of statements \( P(0), P(1), P(2), \ldots \) labeled by non-negative integers that we’d like to prove. For example, \( P(n) \) could be the statement: \( \sum_{i=0}^{n} i = n(n+1)/2 \). In order to prove that all of the statements \( P(n) \) are true using induction, we need to do 2 things:

- Prove that \( P(0) \) is true.
- Assuming that \( P(n) \) is true, use it to prove that \( P(n+1) \) is true.

Let’s see how that works for our example:

- \( P(0) \) is the statement \( \sum_{i=0}^{0} i = 0 \cdot 1/2 \). Both sides are 0, so the equality is valid.
- Now we assume that \( P(n) \) is true, i.e., that \( \sum_{i=0}^{n} i = n(n+1)/2 \). Now we want to prove that \( \sum_{i=0}^{n+1} i = (n+1)(n+2)/2 \). Add \( n+1 \) to both sides of the original identity.

Then the left side becomes \( \sum_{i=0}^{n+1} i \) and the right side becomes \( n(n+1)/2 + n + 1 = (n+1)(n/2+1) = (n+1)(n+2)/2 \), so the new identity we want is valid.

Since we’ve completed the two required steps, we have proven that the summation identity holds for all \( n \).

**Remark 2.1.** Why does this work? It is intuitively clear: if we wanted to know that \( P(3) \) is true, then we start with \( P(0) \), which is true by the first step. By the second step, we know \( P(1) \) holds, and again by applying the second step, we then have \( P(2) \) and \( P(3) \). This can be repeated for any value \( n \). In more rigorous terms, this works because the natural numbers are **well-ordered**: any subset of the natural numbers has a minimum element. To see why this is relevant, assume induction doesn’t work: then let \( S \) be the set of \( n \) such that \( P(n) \) is false. By well-ordering, \( S \) has a minimal element, call it \( N \). If \( N = 0 \), then we’ve contradicted the first step for induction. Otherwise, \( P(N-1) \) is true since \( N-1 \notin S \), but now we’ve contradicted the second step of induction.

This may seem like more work than is necessary, but actually induction can be carried out on index sets besides the natural numbers as long as there is some kind of well-ordering floating around. We won’t get into this generalization though. \( \square \)

**Remark 2.2.** We have labeled the statements starting from 0, but sometimes it’s more natural to start counting from 1 instead, or even some larger integer. The same reasoning as above will apply for these variations. The first step “Prove that \( P(0) \) is true” is then replaced by “Prove that \( P(1) \) is true” or wherever the start of your indexing occurs. \( \square \)

For the next statement, let’s clarify some terminology. A finite set of size \( n \) is a collection of \( n \) different objects (\( n \) could be 0 in which case we call it the empty set and denote it \( \emptyset \)). It could be \( \{1, 2, \ldots, n\} \) or something more strange like \( \{1, *, U\} \). The names of the elements aren’t really important. A **subset** \( T \) of a set \( S \) is another set all of whose elements belong to \( S \). We write this as \( T \subseteq S \). We allow the possibility that \( T \) is empty and also the possibility that \( T = S \).

**Theorem 2.3.** There are \( 2^n \) subsets of a set of size \( n \).

For example, if \( S = \{1, *, U\} \), then there are \( 2^3 = 8 \) subsets, and we can list them: \( \emptyset, \{1\}, \{*\}, \{U\}, \{1, *\}, \{1, U\}, \{U, *\}, \{1, *, U\} \).

**Proof.** Let \( P(n) \) be the statement that any set of size \( n \) has exactly \( 2^n \) subsets.

We check \( P(0) \) directly: if \( S \) has 0 elements, then \( S = \emptyset \), and the only subset is \( S \) itself, which is consistent with \( 2^0 = 1 \).
Now we assume $P(n)$ holds and use it to show that $P(n+1)$ is also true. Let $S$ be a set of size $n+1$. Pick an element $x \in S$ and let $S'$ be the subset of $S$ consisting all elements that are not equal to $x$, i.e., $S' = S \setminus \{x\}$. Then $S'$ has size $n$, so by induction the number of subsets of $S'$ is $2^n$. Now, every subset of $S$ either contains $x$ or it does not. Those which do not contain $x$ can be thought of as subsets of $S'$, so there are $2^n$ of them. To count those that do contain $x$, we can take any subset of $S'$ and add $x$ to it. This accounts for all of them exactly once, so there are also $2^n$ subsets that contain $x$. All together we have $2^n + 2^n = 2^{n+1}$ subsets of $S$, so $P(n+1)$ holds.

Continuing with our example, if $x = 1$, then the subsets not containing $x$ are $\emptyset, \{\ast\}, \{U\}, \{\ast, U\}$, while those that do contain $x$ are $\{1\}, \{1, \ast\}, \{1, U\}, \{1, \ast, U\}$. There are $2^2 = 4$ of each kind.

A natural followup is to determine how many subsets have a given size. In our previous example, there is 1 subset of size 0, 3 of size 1, 3 of size 2, and 1 of size 3. We'll discuss this problem in the next section.

Some more to think about:
- Show that $\sum_{i=0}^{n} i^2 = n(n+1)(2n+1)/6$ for all $n \geq 0$.
- Show that $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$ for all $n \geq 0$.
- Show that $4n < 2^n$ whenever $n \geq 5$.

What happens with $\sum_{i=0}^{n} i^3$ or $\sum_{i=0}^{n} i^4$, or...? In the first two cases, we got polynomials in $n$ on the right side. You'll show on homework that this always happens.

2.2. Strong induction. The version of induction we just described is sometimes called “weak induction”. Here’s a variant sometimes called “strong induction”. We have the same setup: we want to prove that a sequence of statements $P(0), P(1), P(2), \ldots$ are true. Then strong induction works by completing the following 2 steps:

- Prove that $P(0)$ is true.
- Assuming that $P(0), P(1), \ldots, P(n)$ are all true, use them to prove that $P(n+1)$ is true.

You should convince yourself that this isn’t really anything logically distinct from weak induction. However, it can sometimes be convenient to use this variation.

Some examples to think about:
- Every positive integer can be written in the form $2^m n$ where $n \geq 0$ and $m$ is an odd integer.
- Every integer $n \geq 2$ can be written as a product of prime numbers.
- Define a function $f$ on the natural numbers by $f(0) = 1, f(1) = 2$, and $f(n+1) = f(n-1) + 2f(n)$ for all $n \geq 1$. Show that $f(n) \leq 3^n$ for all $n \geq 0$.
- A chocolate bar is made up of unit squares in an $n \times m$ rectangular grid. You can break up the bar into 2 pieces by breaking on either a horizontal or vertical line. Show that you need to make $nm - 1$ breaks to completely separate the bar into $1 \times 1$ squares (if you have 2 pieces already, stacking them and breaking them counts as 2 breaks).

3. Elementary counting problems

3.1. Functions. Let $X, Y$ be sets and $f : X \to Y$ a function from $X$ to $Y$. We make the following definitions:
• $f$ is **injective** / $f$ is an **injection** if, for all $x, x' \in X$, we have $f(x) = f(x')$ implies that $x = x'$. In other words, different elements in $X$ get sent to different values in $Y$.

• $f$ is **surjective** / $f$ is a **surjection** if, for all $y \in Y$, there is some $x \in X$ such that $f(x) = y$. In other words, all possible values in $Y$ are achieved.

• $f$ is **bijective** / $f$ is a **bijection** if it is both injective and surjective.

The last part of the following is very important for this course and forms the basis of “bijective proofs”.

**Theorem 3.1.** Let $X, Y$ be finite sets.

1. If there is an injection $f : X \to Y$, then $|X| \leq |Y|$.
2. If there is a surjection $f : X \to Y$, then $|X| \geq |Y|$.
3. If there is a bijection $f : X \to Y$, then $|X| = |Y|$.

**Proof.** Write the elements of $X$ as $X = \{x_1, \ldots, x_n\}$, so $|X| = n$.

1. The elements $f(x_1), \ldots, f(x_n)$ are all distinct elements of $Y$ since $f$ is an injection, so $Y$ contains a subset of size $n$, and hence $|Y| \geq n = |X|$.

2. If there is a surjection $f : X \to Y$, then every element of $Y$ is of the form $f(x_i)$ for some $i$. This means that $Y$ has at most $n$ elements (some of the values could coincide) which means that $|Y| \leq n = |X|$.

3. By (1) and (2), if there is a bijection $f : X \to Y$, then we would have $|X| \leq Y$ and $|X| \geq |Y|$ and hence $|X| = |Y|$.

The following can be helpful for establishing that a function is a bijection.

**Proposition 3.2.** Let $X, Y$ be finite sets and $f : X \to Y$ a function. Then $f$ is a bijection if any of the following 2 properties hold:

1. $f$ is injective,
2. $f$ is surjective,
3. $|X| = |Y|$.

**Proof.** Write the elements of $X$ as $X = \{x_1, \ldots, x_n\}$, so $|X| = n$.

We check all possibilities. If (1) and (2) hold, then $f$ is a bijection by definition.

Suppose that (1) and (3) hold. Since $f$ is injective, the elements $f(x_1), \ldots, f(x_n)$ give $n$ distinct elements of $Y$. But since $|X| = |Y|$, they must account for all of the elements of $Y$. This means that $f$ is surjective since every element of $Y$ is of the form $f(x_i)$ for some $i$. Hence $f$ is bijective.

Suppose that (2) and (3) hold. Since $f$ is surjective, every element of $Y$ is of the form $f(x_i)$ for some $i$. Since $|Y| = |X| = n$, the $n$ elements $f(x_1), \ldots, f(x_n)$ have to all be distinct (since they account for all of the elements of $Y$). Hence $f$ is injective, and so $f$ is bijective by definition.

Given two functions $f : X \to Y$ and $g : Y \to X$, we say that they are inverses if $f \circ g$ is the identity function on $Y$, i.e., $f(g(y)) = y$ for all $y \in Y$, and if $g \circ f$ is the identity function on $X$, i.e., $g(f(x)) = x$ for all $x \in X$. You should have seen the following before; if not, we’ll leave it as an exercise.

**Proposition 3.3.** $f : X \to Y$ is a bijection if and only if there exists an inverse $g : Y \to X$. 

3.2. 12-fold way, introduction. We have \( k \) balls and \( n \) boxes. Roughly speaking, this chapter is about counting the number of ways to put the balls into boxes. We can think of each assignment as a function from the set of balls to the set of boxes. Phrased this way, we will be examining how many ways to do this if we require \( f \) to be injective, or surjective, or completely arbitrary. Are the boxes supposed to be considered different or interchangeable (we also use the terminology distinguishable and indistinguishable)? And same with the balls, are they considered different or interchangeable? All in all, this will give us 12 different problems to consider, which means we want to understand the following table:

<table>
<thead>
<tr>
<th>balls/boxes</th>
<th>( f ) arbitrary</th>
<th>( f ) injective</th>
<th>( f ) surjective</th>
</tr>
</thead>
<tbody>
<tr>
<td>dist/dist</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>indist/dist</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dist/indist</td>
<td>( \begin{cases} 1 &amp; \text{if } n \geq k \ 0 &amp; \text{if } n &lt; k \end{cases} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>indist/indist</td>
<td>( \begin{cases} 1 &amp; \text{if } n \geq k \ 0 &amp; \text{if } n &lt; k \end{cases} )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Two situations have already been filled in and won't be considered interesting. We now study some problems that will allow us to fill in the rest of the table.

3.3. Permutations and combinations. Given a set \( S \) of objects, a permutation of \( S \) is a way to put all of the elements of \( S \) in order.

**Example 3.4.** There are 6 permutations of \( \{1, 2, 3\} \) which we list:

\[
123, \ 132, \ 213, \ 231, \ 312, \ 321. \quad \square
\]

To count permutations in general, we define the factorial as follows: \( 0! = 1 \) and if \( n \) is a positive integer, then \( n! = n \cdot (n-1)! \). Here are the first few values:

\[
0! = 1, \quad 1! = 1, \quad 2! = 2, \quad 3! = 6, \quad 4! = 24, \quad 5! = 120, \quad 6! = 720.
\]

In the previous example, we had 6 permutations of 3 elements, and 6 = 3!. This holds more generally:

**Theorem 3.5.** If \( S \) has \( n \) elements and \( n > 0 \), then there are \( n! \) different permutations of \( S \).

**Proof.** We do this by induction on \( n \). Let \( P(n) \) be the statement that a set of size \( n \) has exactly \( n! \) elements. The statement \( P(1) \) follows from the definition: there is exactly 1 way to order a single element, and 1! = 1. Now assume for our induction hypothesis that \( P(n) \) has been proven. Let \( S \) be a set of size \( n+1 \). To order the elements, we can first pick any element to be first, and then we have to order the remaining \( n \) elements. There are \( n+1 \) different elements that can be first, and for each such choice, there are \( n! \) ways to order the remaining elements by our induction hypothesis. So all together, we have \( (n+1) \cdot n! = (n+1)! \) different ways to order all of them, which proves \( P(n+1) \). \( \square \)

We can use factorials to answer related questions. For example, suppose that some of the objects in our set can’t be distinguished from one another, so that some of the orderings end up being the same.
Example 3.6. (1) Suppose we are given 2 red flowers and 1 yellow flower. Aside from their color, the flowers look identical. We want to count how many ways we can display them in a single row. There are 3 objects total, so we might say there are $3! = 6$ such ways. But consider what the 6 different ways look like:

$$RRY, \ RRY, \ RYR, \ YRR, \ YRR.$$ 

Since the two red flowers look identical, we don’t actually care which one comes first. So there are really only 3 different ways to do this – the answer $3!$ has included each different way twice, but we only wanted to count them a single time.

(2) Consider a larger problem: 10 red flowers and 5 yellow flowers. There are too many to list, so we consider a different approach. As above, if we naively count, then we would get $15!$ permutations of the flowers. But note that for any given arrangement, the 10 red flowers can be reordered in any way to get an identical arrangement, and same with the yellow flowers. So in the list of $15!$ permutations, each arrangement is being counted $10! \cdot 5!$ times. The number of distinct arrangements is then $\frac{15!}{10! \cdot 5!}$.

(3) The same reasoning allows us to generalize. If we have $r$ red flowers and $y$ yellow flowers, then the number of different ways to arrange them is $\frac{(r+y)!}{r!y!}$.

(4) How about more than 2 colors of flowers? If we threw in $b$ blue flowers, then again the same reasoning gives us $\frac{(r+y+b)!}{r!y!b!}$ different arrangements. \hfill \Box

Now we state a general formula, which again can be derived by the same reasoning as in (2) above. Suppose we are given $n$ objects, which have one of $k$ different types (for example, our objects could be flowers and the types are colors). Also, objects of the same type are considered identical. For convenience, we will label the “types” with numbers $1, 2, \ldots, k$ and let $a_i$ be the number of objects of type $i$ (so $a_1 + a_2 + \cdots + a_k = n$).

**Theorem 3.7.** The number of ways to arrange the $n$ objects in the above situation is

$$\frac{n!}{a_1!a_2!\cdots a_k!}.$$ 

As an exercise, you should adapt the reasoning in (2) to give a proof of this theorem.

The quantity above will be used a lot, so we give it a symbol, called the **multinomial coefficient**:

$$\binom{n}{a_1, a_2, \ldots, a_k} := \frac{n!}{a_1!a_2!\cdots a_k!}.$$ 

In the case when $k = 2$ (a very important case), it is called the **binomial coefficient**. Note that in this case, $a_2 = n - a_1$, so for shorthand, one often just writes $\binom{n}{a_1}$ instead of $\binom{n}{a_1, a_2}$. For similar reasons, $\binom{n}{a_2}$ is also used as a shorthand.

3.4. **Words.** A **word** is a finite ordered sequence whose entries are drawn from some set $A$ (which we call the **alphabet**). The **length** of the word is the number of entries it has. Entries may repeat, there is no restriction on that. Also, the empty sequence $\emptyset$ is considered a word of length 0.

**Example 3.8.** Say our alphabet is $A = \{a, b\}$. The words of length $\leq 2$ are:

$$\emptyset, \ a, \ b, \ aa, \ ab, \ ba, \ bb.$$ \hfill \Box

**Theorem 3.9.** If $|A| = n$, then the number of words in $A$ of length $k$ is $n^k$. 

Proof. A sequence of length \( k \) with entries in \( A \) is an element in the product set \( A^k = A \times A \times \cdots \times A \) and \( |A^k| = |A|^k \).

Alternatively, we can think of this as follows. To specify a word, we pick each of its entries, but these can be done independently of the other choices. So for each of the \( k \) positions, we are choosing one of \( n \) different possibilities, which leads us to \( n \cdot n \cdots n = n^k \) different choices for words.

For a positive integer \( n \), let \( [n] \) denote the set \( \{1, \ldots, n\} \).

Example 3.10. We use words to show that the number of subsets of \( [n] \) is \( 2^n \) (we’ve already seen this result, so now we’re using a different proof method).

Given a subset \( S \subseteq [n] \), we define a word \( w_S \) of length \( n \) in the alphabet \( \{0, 1\} \) as follows. If \( i \in S \), then the \( i \)th entry of \( w_S \) is 1, and otherwise the entry is 0. This defines a function

\[
f: \{\text{subsets of } [n]\} \to \{\text{words of length } n \text{ on } \{0, 1\}\}.
\]

We can also define an inverse function: given such a word \( w \), we send it to the subset of positions where there is a 1 in \( w \). We omit the check that these two functions are inverse to one another. So \( f \) is a bijection, and the previous result tells us that there are \( 2^n \) words of length \( n \) on \( \{0, 1\} \). □

How about words without repeating entries? Given \( n \geq k \), define the falling factorial by

\[
(n)_k := n(n-1)(n-2)\cdots(n-k+1).
\]

There are \( k \) numbers being multiplied in the above definition. When \( n = k \), we have \( (n)_n = n! \), so this generalizes the factorial function.

Theorem 3.11. If \( |A| = n \) and \( n \geq k \), then there are \( (n)_k \) different words of length \( k \) in \( A \) which do not have any repeating entries.

Proof. Start with a permutation of \( A \). The first \( k \) elements in that permutation give us a word of length \( k \) with no repeating entries. But we’ve overcounted because we don’t care how the remaining \( n-k \) things we threw away are ordered. In particular, this process returns each word exactly \( (n-k)! \) many times, so our desired quantity is

\[
\frac{n!}{(n-k)!} = (n)_k.
\]

Some further things to think about:
- A small city has 10 intersections. Each one could have a traffic light or gas station (or both or neither). How many different configurations could this city have?
- Using that \( (n)_k = n \cdot (n-1)_{k-1} \), can you find a proof for Theorem 3.11 that uses induction?
- Which additional entries of the 12-fold way table can we fill in now?

3.5. Choice problems. We finish up with some related counting problems. Recall we showed that an \( n \)-element set has exactly \( 2^n \) subsets. We can refine this problem by asking about subsets of a given size.

Theorem 3.12. The number of \( k \)-element subsets of \( [n] \) is

\[
\binom{n}{k} = \frac{n!}{k!(n-k)!}.
\]
There are many ways to prove this, but we’ll just do one for now:

**Proof.** In the last section on words, we identified subsets of \([n]\) with words of length \(n\) on \(\{0, 1\}\), with a 1 in position \(i\) if and only if \(i\) belongs to the subset. So the number of subsets of size \(k\) are exactly the number of words with exactly \(k\) instances of 1. This is the same as arranging \(n - k\) 0’s and \(k\) 1’s from the section on permutations. In that case, we saw the answer is \(\frac{n!}{(n-k)!k!}\).

□

**Corollary 3.13.** \(\sum_{k=0}^{n} \binom{n}{k} = 2^n.\)

**Proof.** The left hand side counts the number of subsets of \([n]\) of some size \(k\) where \(k\) ranges from 0 to \(n\). But all subsets of \([n]\) are accounted for and we’ve seen that \(2^n\) is the number of all subsets of \([n]\). □

Here’s an important identity for binomial coefficients (we interpret \(\binom{n}{-1} = 0\):

**Proposition 3.14 (Pascal’s identity).** For any \(k \geq 0\), we have

\[
\binom{n}{k-1} + \binom{n}{k} = \binom{n+1}{k}.
\]

**Proof.** The right hand side is the number of subsets of \([n+1]\) of size \(k\). There are 2 types of such subsets: those that contain \(n + 1\) and those that do not. Note that the subsets that do contain \(n + 1\) are naturally in bijection with the subsets of \([n]\) of size \(k - 1\): to get such a subset, delete \(n + 1\). Those that do not contain \(n + 1\) are naturally already in bijection with the subsets of \([n]\) of size \(k\). The two sets don’t overlap and their sizes are \(\binom{n}{k-1}\) and \(\binom{n}{k}\), respectively. □

An important variation of subset is the notion of a multiset. Given a set \(S\), a **multiset** of \(S\) is like a subset, but we allow elements to be repeated. Said another way, a subset of \(S\) can be thought of as a way of assigning either a 0 or 1 to an element, based on whether it gets included. A multiset is then a way to assign some non-negative integer to each element, where numbers bigger than 1 mean we have picked them multiple times.

**Example 3.15.** There are 10 multisets of \([3]\) of size 3:

\[
\{1, 1, 1\}, \{1, 1, 2\}, \{1, 1, 3\}, \{1, 2, 2\}, \{1, 2, 3\},
\{1, 3, 3\}, \{2, 2, 2\}, \{2, 2, 3\}, \{2, 3, 3\}, \{3, 3, 3\}.
\]

Aside from exhaustively checking, how do we know that’s all of them? Here’s a trick: given a multiset, add 1 to the second smallest values (including ties) and add 2 to the largest value. What happens to the above:

\[
\{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}, \{1, 3, 4\}, \{1, 3, 5\},
\{1, 4, 5\}, \{2, 3, 4\}, \{2, 3, 5\}, \{2, 4, 5\}, \{3, 4, 5\}.
\]

We get all of the 3-element subsets of \([5]\). The process is reversible using subtraction, so there is a more general fact here. □

**Theorem 3.16.** The number of \(k\)-element multisets of \([n]\) is

\[
\binom{n + k - 1}{k}.
\]
Proof. We adapt the example above to find a bijection between \( k \)-element multisets of \([n]\) and \( k \)-element subsets of \([n+k-1]\). Given a multiset \( S \), sort the elements as \( s_1 \leq s_2 \leq \cdots \leq s_k \). From this, we get a subset \( \{s_1, s_2 + 1, s_3 + 2, \ldots, s_k + (k-1)\} \) of \([n+k-1]\). On the other hand, given a subset \( T \) of \([n+k-1]\), sort the elements as \( t_1 < t_2 < \cdots < t_k \). From this, we get a multiset \( \{t_1, t_2 - 1, t_3 - 2, \ldots, t_k - (k-1)\} \) of \([n]\). We will omit the details that these are well-defined and inverse to one another.

Some additional things:

- From the formula, we see that \( \binom{n}{k} = \binom{n}{n-k} \). This would also be implied if we could construct a bijection between the \( k \)-element subsets and the \((n-k)\)-element subsets of \([n]\). Can you find one?
- What other entries of the 12-fold way table can be filled in now?
- Given variables \( x, y, z \), we can form polynomials. A monomial is a product of the form \( x^a y^b z^c \), and its degree is \( a + b + c \). How many monomials in \( x, y, z \) are there of degree \( d \)? What if we have \( n \) variables \( x_1, x_2, \ldots, x_n \)?

4. Partitions and compositions

4.1. Compositions. Below, \( n \) and \( k \) are positive integers.

Definition 4.1. A sequence of non-negative integers \((a_1, \ldots, a_k)\) is a weak composition of \( n \) if \( a_1 + \cdots + a_k = n \). If all of the \( a_i \) are positive, then it is a composition. We call \( k \) the number of parts of the (weak) composition.

Theorem 4.2. The number of weak compositions of \( n \) with \( k \) parts is \( \binom{n+k-1}{n} = \binom{n+k-1}{k-1} \).

Proof. We will construct a bijection between weak compositions of \( n \) with \( k \) parts and \( n \)-element multisets of \([k]\). First, given a weak composition \((a_1, \ldots, a_k)\), we get a multiset which has the element \( i \) exactly \( a_i \) many times. Since \( a_1 + \cdots + a_k = n \), this is an \( n \)-element multiset of \([k]\). Conversely, given a \( n \)-element multiset \( S \) of \([k]\), let \( a_i \) be the number of times that \( i \) appears in \( S \), so that we get a weak composition \((a_1, \ldots, a_k)\) of \( n \).

Example 4.3. We want to distribute 20 pieces of candy (all identical) to 4 children. How many ways can we do this? If we order the children and let \( a_i \) be the number of pieces of candy that the \( i \)th child receives, then \((a_1, a_2, a_3, a_4)\) is just a weak composition of 20 into 4 parts, so we can identify all ways with the set of all weak compositions. So we know that the number of ways is \( \binom{20+4-1}{20} = \binom{23}{20} \).

What if we want to ensure that each child receives at least one piece of candy? First, hand each child 1 piece of candy. We have 16 pieces left, and we can distribute them as we like, so we’re counting weak compositions of 16 into 4 parts, or \( \binom{19}{16} \).

As we saw with the previous example, given a weak composition \((a_1, \ldots, a_k)\) of \( n \), we can think of it as an assignment of \( n \) indistinguishable objects into \( k \) distinguishable boxes, so this fills in one of the entries in the 12-fold way. A composition is an assignment which is required to be surjective, so actually this takes care of 2 of the entries.

Corollary 4.4. The number of compositions of \( n \) into \( k \) parts is \( \binom{n-1}{k-1} \).

Proof. If we generalize the argument in the last example, we see that compositions of \( n \) into \( k \) parts are in bijection with weak compositions of \( n-k \) into \( k \) parts.

Corollary 4.5. The total number of compositions of \( n \) (into any number of parts) is \( 2^{n-1} \).
Proof. The possible number of parts of a composition of \( n \) is anywhere between \( k = 1 \) to \( k = n \). So the total number of compositions possible is

\[
\sum_{k=1}^{n} \binom{n-1}{k-1} = \sum_{k=0}^{n-1} \binom{n-1}{k} = 2^{n-1}.
\]

The answer suggests that we should be able to find a bijection between compositions of \( n \) and subsets of \([n-1]\). Can you find one?

4.2. **Set partitions.** (Weak) compositions were about indistinguishable objects into distinguishable boxes. Now we reverse the roles and consider distinguishable objects into indistinguishable boxes.

**Definition 4.6.** Let \( X \) be a set. A **partition** of \( X \) is an unordered collection of nonempty subsets \( S_1, \ldots, S_k \) of \( X \) such that every element of \( X \) belongs to exactly one of the \( S_i \). The \( S_i \) are the **blocks** of the partition. Partitions of sets are also called **set partitions** to distinguish from integer partitions, which will be discussed next.

**Example 4.7.** Let \( X = \{1, 2, 3\} \). There are 5 partitions of \( X \):

\[
\{\{1, 2, 3\}\}, \ \{\{1, 2\}, \{3\}\}, \ \{\{1, 3\}, \{2\}\}, \ \{\{2, 3\}, \{1\}\}, \ \{\{1\}, \{2\}, \{3\}\}.
\]

When we say unordered collection of subsets, we mean that \( \{\{1, 2\}, \{3\}\} \) and \( \{\{3\}, \{1, 2\}\} \) are to be considered the same partition.

The notation above is a little cumbersome, so we can also write the above partitions as follows:

\[
123, \ 12|3, \ 13|2, \ 23|1, \ 12|3.
\]

The number of partitions of \( X \) with \( k \) blocks only depends on the number of elements of \( X \). So for concreteness, we will usually assume that \( X = [n] \).

**Example 4.8.** If we continue with our previous example of candy and children: imagine the 20 pieces of candy are now labeled 1 through 20 and that the 4 children are all identical clones. The number of ways to distribute candy to them so that each gets at least 1 piece of candy is then the number of partitions of \([20]\) into 4 blocks.

**Definition 4.9.** We let \( S(n, k) \) be the number of partitions of \([n]\) into \( k \) blocks. These are called the **Stirling numbers of the second kind**. By convention, we define \( S(0, 0) = 1 \). Note that \( S(n, k) = 0 \) if \( k > n \).

So \( S(n, k) \) is, by definition, an answer to one of the 12-fold way entries: how many ways to put \( n \) distinguishable objects into \( k \) indistinguishable boxes. It will be generally hard to get nice, exact formulas for \( S(n, k) \), but we can do some special cases:

**Example 4.10.** For \( n \geq 1 \), \( S(n, 1) = S(n, n) = 1 \). For \( n \geq 2 \), \( S(n, 2) = 2^{n-1} - 1 \) and \( S(n, n - 1) = \binom{n}{2} \). Can you see why?

We also have the following recursive formula:

**Theorem 4.11.** If \( k \leq n \), then

\[
S(n, k) = S(n - 1, k - 1) + kS(n - 1, k).
\]
Proof. Consider two kinds of partitions of \([n]\). The first kind is when \(n\) is in its own block. In that case, if we remove this block, then we obtain a partition of \([n-1]\) into \(k-1\) blocks. To reconstruct the original partition, we just add a block containing \(n\) by itself. So the number of such partitions is \(S(n-1, k-1)\).

The second kind is when \(n\) is not in its own block. This time, if we remove \(n\), we get a partition of \(n-1\) into \(k\) blocks. However, it’s not possible to reconstruct the original block because we can’t remember which block it belonged to. So in fact, there are \(k\) different ways to try to reconstruct the original partition. This means that the number of such partitions is \(kS(n-1, k)\).

If we add both answers, we account for all possible partitions of \([n]\), so we get the identity we want. 

Here’s a table of small values of \(S(n, k)\):

<table>
<thead>
<tr>
<th>(n \backslash k)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>15</td>
<td>25</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

We define \(B(n)\) to be the number of partitions of \([n]\) into any number of blocks. This is the \(n\)th Bell number. By definition,

\[
B(n) = \sum_{k=0}^{n} S(n, k).
\]

We have the following recursion:

**Theorem 4.12.** \(B(n+1) = \sum_{i=0}^{n} \binom{n}{i} B(i)\).

Proof. We separate all of the set partitions of \([n+1]\) based on the number of elements in the block that contains \(n+1\). Consider those where the size is \(j\). To count the number of these, we need to first choose the other elements to occupy the same block as \(n+1\). These numbers come from \([n]\) and there are \(j-1\) to be chosen, so there are \(\binom{n}{j-1}\) ways to do this. We have to then choose a set partition of the remaining \(n+1-j\) elements, and there are \(B(n+1-j)\) many of these. So the number of such partitions is \(\binom{n}{j-1}B(n+1-j)\). The possible values for \(j\) are between 1 and \(n+1\), so we get the identity

\[
B(n+1) = \sum_{j=1}^{n+1} \binom{n}{j-1}B(n+1-j).
\]

Re-index the sum by setting \(i = n+1-j\) and use the identity \(\binom{n}{i} = \binom{n}{n-i}\) to get the desired identity. \(\square\)

4.3. Integer partitions. Now we come to the situation where both balls and boxes are indistinguishable. In this case, the only relevant information is how many boxes are empty, how many contain exactly 1 ball, how many contain exactly 2 balls, etc. We use the following structure:
Definition 4.13. An **partition** of an integer \( n \) is a sequence of non-negative integers \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k) \) so that \( \lambda_1 + \lambda_2 + \cdots + \lambda_k = n \) and so that \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k \). The \( \lambda_i \) are the parts of \( \lambda \). We use the notation \(|\lambda| = n\) (size of the partition) and \( \ell(\lambda) \) (length of the partition) is the number of \( \lambda_i \) which are positive. These are also called **integer partitions** to distinguish from set partitions.

We will consider two partitions the same if they are equal after removing all of the parts equal to 0.

The number of partitions of \( n \) is denoted \( p(n) \), and the number of partitions of \( n \) with \( k \) parts is denoted \( p_k(n) \).

We’ve reversed the roles of \( n \) and \( k \), but the partition \( (\lambda_1, \ldots, \lambda_k) \) encodes an assignment of \( n \) balls to \( k \) boxes where some box has \( \lambda_1 \) balls, another box has \( \lambda_2 \) balls, etc. Remember we don’t distinguish the boxes, so we can list the \( \lambda_i \) in any order and we’d get an equivalent assignment. But our convention will be that the \( \lambda_i \) are listed in weakly decreasing order.

Example 4.14. \( p(5) = 7 \) since there are 7 partitions of 5:

\( (5), \quad (4,1), \quad (3,2), \quad (3,1,1), \quad (2,2,1), \quad (2,1,1,1), \quad (1,1,1,1,1). \)

We can visualize partitions using **Young diagrams**. To illustrate, the Young diagram of \((4,2,1)\) is

\[
Y(\lambda) = \begin{array}{ccc}
\square & \square & \\
\square & & \\
\end{array}
\]

In general, it is a left-justified collection of boxes with \( \lambda_i \) boxes in the \( i \)th row (counting from top to bottom).

The **transpose** (or **conjugate**) of a partition \( \lambda \) is the partition whose Young diagram is obtained by flipping \( Y(\lambda) \) across the main diagonal. For example, the transpose of \((4,2,1)\) is \((3,2,1,1)\):

\[
\begin{array}{ccc}
\square & \square & \square \\
\square & & \\
\end{array}
\]

Note that we get the parts of a partition from a Young diagram by reading off the row lengths. The transpose is obtained by instead reading off the column lengths. The notation is \( \lambda^T \). If we want a formula: \( \lambda^T_j = |\{i \mid \lambda_i \geq j\}| \).

Note that \((\lambda^T)^T = \lambda\). A partition \( \lambda \) is **self-conjugate** if \( \lambda = \lambda^T \).

Example 4.15. Some self-conjugate partitions: \((4,3,2,1), (5,1,1,1,1), (4,2,1,1)\):

\[
\begin{array}{ccc}
\square & \square & \square \\
\square & & \\
\end{array}, \quad
\begin{array}{ccc}
\square & \square & \square \\
\square & & \\
\end{array}, \quad
\begin{array}{ccc}
\square & \square & \square \\
\square & & \\
\end{array}
\]

Theorem 4.16. The number of partitions \( \lambda \) of \( n \) with \( \ell(\lambda) \leq k \) is the same as the number of partitions \( \mu \) of \( n \) such that all \( \mu_i \leq k \).
Proof. We get a bijection between the two sets by taking transpose. Details omitted. □

**Theorem 4.17.** The number of self-conjugate partitions of \( n \) is equal to the number of partitions of \( n \) using only distinct odd parts.

**Proof.** Given a self-conjugate partition, take all of the boxes in the first row and column of its Young diagram. Since it’s self-conjugate, there are an odd number of boxes. Use this as the first part of a new partition. Now remove those boxes and repeat. For example, starting

with

\[
\begin{array}{|c|c|c|}
\hline
& & \\
\hline
& & \\
\hline
\end{array}
\]

we get

\[
\begin{array}{|c|c|c|}
\hline
& & \\
\hline \\
\end{array}
\]

and starting with

\[
\begin{array}{|c|c|c|}
\hline
& & \\
\hline
& & \\
\hline
\end{array}
\]

we get

\[
\begin{array}{|c|c|c|}
\hline
& & \\
\hline
\end{array}
\]

In formulas, if \( \lambda \) is self-conjugate, then

\[
\mu_i = \lambda_i - (i - 1) + \lambda'_i - (i - 1) - 1 = 2\lambda_i - 2i + 1
\]

and so \( \mu_1 > \mu_2 > \cdots \).

This process is reversible: let \( \mu \) be a partition with distinct odd parts. Each part \( \mu_i \) can be turned into a shape with a single row and column, both of length \( (\mu_i + 1)/2 \). Since the \( \mu_i \) are distinct, these shapes can be nested into one another to form the partition \( \lambda \) (this is easiest to understand by studying the two examples above). □

4.4. **12-fold way, summary.** We have \( k \) balls and \( n \) boxes. We want to count the number of assignments \( f \) of balls to boxes. We considered 3 conditions on \( f \): arbitrary (no conditions at all), injective (no box receives more than one ball), surjective (every box has to receive at least one ball). We also considered conditions on the balls: indistinguishable (we can’t tell the balls apart) and distinguishable (we can tell the balls apart) and similarly for the boxes: they can be distinguishable or indistinguishable.

<table>
<thead>
<tr>
<th>balls/boxes</th>
<th>( f ) arbitrary</th>
<th>( f ) injective</th>
<th>( f ) surjective</th>
</tr>
</thead>
<tbody>
<tr>
<td>dist/dist</td>
<td>( n^k ), see (1)</td>
<td>( (n)_k ), see (2)</td>
<td>( n!S(k,n) ), see (3)</td>
</tr>
<tr>
<td>indist/dist</td>
<td>( \binom{n+k-1}{k} ), see (4)</td>
<td>( \binom{n}{k} ), see (5)</td>
<td>( \binom{k-1}{n-1} ), see (6)</td>
</tr>
<tr>
<td>dist/indist</td>
<td>( \sum_{i=1}^{n} S(k,i) ), see (7)</td>
<td>{ 1 if ( n \geq k ), see (8) }</td>
<td>( S(k,n) ), see (9)</td>
</tr>
<tr>
<td>indist/indist</td>
<td>( \sum_{i=1}^{n} p_i(k) ), see (10)</td>
<td>{ 1 if ( n \geq k ), see (11) }</td>
<td>( p_n(k) ), see (12)</td>
</tr>
</tbody>
</table>

(1) These are words of length \( k \) in an alphabet of size \( n \).
(2) These are words of length \( k \) without repetitions in an alphabet of size \( n \). Recall that

\[
(n)_k = n(n-1)(n-2) \cdots (n-k+1).
\]

(3) These are set partitions of \( [k] \) into \( n \) blocks that have an ordering on the blocks. Recall that \( S(k,n) \) is the Stirling number of the second kind, i.e., the number of partitions of \( [k] \) into \( n \) blocks.
(4) These are multisets of \( [n] \) of size \( k \); equivalently, weak compositions of \( k \) into \( n \) parts.
(5) These are subsets of \( [n] \) of size \( k \).
(6) These are compositions of \( k \) into \( n \) parts.
(7) These are set partitions of \( [k] \) where the number of blocks is \( \leq n \).
(8) If \( n < k \), then we can’t assign \( k \) balls to \( n \) boxes without some box receiving more than one ball (pigeonhole principle), so the answer is 0 in that case. If \( n \geq k \), then
there is certainly a way to make an assignment, but they’re all the same: we can’t tell the boxes apart, so it doesn’t matter where the balls go.

(9) These are set partitions of \([k]\) into \(n\) blocks.

(10) These are the number of integer partitions of \(k\) where the number of parts is \(\leq n\). Remember that \(p_i(k)\) is the notation for the number of integer partitions of \(k\) into \(i\) parts.

(11) The reasoning here is the same as (8).

(12) These are the number of integer partitions of \(k\) into \(n\) parts.

5. Binomial theorem and generalizations

5.1. Binomial theorem. The binomial theorem is about expanding powers of \(x + y\) where we think of \(x, y\) as variables. For example:

\[
(x + y)^2 = x^2 + 2xy + y^2, \\
(x + y)^3 = x^3 + 3x^2y + 3xy^2 + y^3.
\]

**Theorem 5.1** (Binomial theorem). For any \(n \geq 0\), we have

\[
(x + y)^n = \sum_{i=0}^{n} \binom{n}{i} x^i y^{n-i}.
\]

Here’s the proof given in the book.

**Proof.** Consider how to expand the product \((x + y)^n = (x + y)(x + y) \cdots (x + y)\). To get a term, from each expression \((x + y)\), we have to either pick \(x\) or \(y\). The final term we get is \(x^i y^{n-i}\) if the number of times we chose \(x\) is \(i\) (and hence the number of times we’ve chosen \(y\) is \(n - i\)). The number of times this term appears is therefore the number of different ways we could have chosen \(x\) exactly \(i\) times. For each way of doing this, we can associate to it a subset of \([n]\) of size \(i\): the number \(j\) is in the subset if and only if we chose \(x\) in the \(j\)th copy of \((x + y)\). We have already seen that the number of subsets of \([n]\) of size \(i\) is \(\binom{n}{i}\). \(\Box\)

Here’s a proof using induction.

**Proof.** For \(n = 0\), the formula becomes \((x + y)^0 = 1\) which is valid.

Now suppose the formula is valid for \(n\). Then we have

\[
(x + y)^{n+1} = (x + y)(x + y)^n = (x + y) \sum_{i=0}^{n} \binom{n}{i} x^i y^{n-i}.
\]

For a given \(k\), there are at most 2 ways to get \(x^k y^{n+1-k}\) on the right side: either we get it from \(x \cdot \binom{n}{k-1} x^{k-1} y^{n-k+1}\) or from \(y \cdot \binom{n}{k} x^k y^{n-k}\). If we add these up, then we get \(\binom{n+1}{k}\) by Pascal’s identity. \(\Box\)

We can manipulate the binomial theorem in a lot of different ways (taking derivatives with respect to \(x\) or \(y\), or doing substitutions). This will give us a lot of new identities. Here are a few of particular interest (some are old):

**Corollary 5.2.** \(2^n = \sum_{i=0}^{n} \binom{n}{i}\).

**Proof.** Substitute \(x = y = 1\) into the binomial theorem. \(\Box\)
This says that the total number of subsets of $[n]$ is $2^n$ which is a familiar fact from before.

**Corollary 5.3.** For $n > 0$, we have $0 = \sum_{i=0}^{n} (-1)^i \binom{n}{i}$.

**Proof.** Substitute $x = -1$ and $y = 1$ into the binomial theorem. \qed

If we rewrite this, it says that the number of subsets of even size is the same as the number of subsets of odd size. It is worth finding a more direct proof of this fact which does not rely on the binomial theorem.

**Corollary 5.4.** $n2^{n-1} = \sum_{i=0}^{n} i \binom{n}{i}$.

**Proof.** Take the derivative of both sides of the binomial theorem with respect to $x$ to get $n(x+y)^{n-1} = \sum_{i=0}^{n} i \binom{n}{i} x^{i-1} y^{n-i}$. Now substitute $x = y = 1$. \qed

It is possible to interpret this formula as the size of some set so that both sides are different ways to count the number of elements in that set. Can you figure out how to do that? How about if we took the derivative twice with respect to $x$? Or if we took it with respect to $x$ and then with respect to $y$?

### 5.2. Multinomial theorem.

**Theorem 5.5 (Multinomial theorem).** For $n, k \geq 0$, we have

$$(x_1 + x_2 + \cdots + x_k)^n = \sum_{(a_1, a_2, \ldots, a_k) \atop a_1 + \cdots + a_k = n} \left( \binom{n}{a_1, a_2, \ldots, a_k} \right) x_1^{a_1} x_2^{a_2} \cdots x_k^{a_k}.$$

**Proof.** The proof is similar to the binomial theorem. Consider expanding the product $(x_1 + \cdots + x_k)^n$. To do this, we first have to pick one of the $x_i$ from the first factor, pick another one from the second factor, etc. To get the term $x_1^{a_1} x_2^{a_2} \cdots x_k^{a_k}$, we need to have picked $x_1$ exactly $a_1$ times, picked $x_2$ exactly $a_2$ times, etc. We can think of this as arranging $n$ objects, where $a_i$ of them have “type $i$”. In that case, we’ve already discussed that this is counted by the multinomial coefficient $\binom{n}{a_1, a_2, \ldots, a_k}$. \qed

By performing substitutions, we can get a bunch of identities that generalize the one from the previous section. I’ll omit the proofs, try to fill them in.

$$k^n = \sum_{(a_1, a_2, \ldots, a_k) \atop a_1 + \cdots + a_k = n} \binom{n}{a_1, a_2, \ldots, a_k},$$

$$0 = \sum_{(a_1, a_2, \ldots, a_k) \atop a_1 + \cdots + a_k = n} (1-k)^{a_1} \binom{n}{a_1, a_2, \ldots, a_k},$$

$$nk^{n-1} = \sum_{(a_1, a_2, \ldots, a_k) \atop a_1 + \cdots + a_k = n} a_1 \binom{n}{a_1, a_2, \ldots, a_k}.$$
6. **Inclusion-exclusion**

**Example 6.1.** Suppose we have a room of students, and 14 of them play basketball, 10 of them play football. How many students play at least one of these? We can't answer the question because there might be students who play both. But we can say that the total number is 24 minus the amount in the overlap.

![Diagram of sets B and F]

Alternatively, let $B$ be the set who play basketball and let $F$ be the set who play football. Then what we’ve said is:

$$|B \cup F| = |B| + |F| - |B \cap F|.$$  

New situation: there are additionally 8 students who play hockey. Let $H$ be the set of students who play hockey. What information do we need to know how many total students there are?

![Diagram of sets B, F, and H]

Here the overlap region is more complicated: it has 4 regions, which suggest that we need 4 more pieces of information. The following formula works:

$$|B \cup F \cup H| = |B| + |F| + |H| - |B \cap F| - |B \cap H| - |F \cap H| - |B \cap F \cap H|.$$  

To see this, the total diagram has 7 regions and we need to make sure that students in each region get counted exactly once in the right side expression. For example, consider students who play basketball and football, but don’t play hockey. They get counted in $B$, $F$, $B \cap F$ with signs $+1$, $+1$, $-1$, which sums up to 1. How about students who play all 3? They get counted in all terms with 4 $+1$ signs and 3 $-1$ signs, again adding up to 1. You can check the other 5 to make sure the count is right. \qed

The examples above have a generalization to $n$ sets, though the diagram is harder to draw beyond 3.

**Theorem 6.2 (Inclusion-Exclusion).** Let $A_1, \ldots, A_n$ be finite sets. Then

$$|A_1 \cup \cdots \cup A_n| = \sum_{j=1}^{n} (-1)^{j-1} \sum_{1 \leq i_1 < i_2 < \cdots < i_j \leq n} |A_{i_1} \cap A_{i_2} \cap \cdots \cap A_{i_j}|.$$  

**Proof.** We just need to make sure that every element $x \in A_1 \cup \cdots \cup A_n$ is counted exactly once on the right hand side. Let $S = \{s_1, \ldots, s_k\}$ be all of the indices such that $x \in A_{s_i}$. Then $x$ belongs to $A_{i_1} \cap \cdots \cap A_{i_j}$ if and only if $\{i_1, \ldots, i_j\} \subseteq S$. So the relevant contributions
for $x$ is a sum over all of the nonempty subsets of $S$:

$$
\sum_{T \subseteq S} (-1)^{|T|-1} = -\sum_{n=1}^{|S|} \binom{|S|}{n} (-1)^n.
$$

However, since $|S| > 0$, we have shown before that $\sum_{n=0}^{|S|} \binom{|S|}{n} (-1)^n = 0$, so the sum above is $\binom{|S|}{0} = 1$. □

We can also prove this by induction on $n$. Can you see how?

We use this to address two counting problems.

First, we can think of a permutation of $[n]$ as the same thing as a bijection $f: [n] \rightarrow [n]$ (given the bijection, $f(i)$ is the position in the permutation where $i$ is supposed to appear).

A derangement of size $n$ is a permutation such that for all $i$, $i$ does not appear in position $i$. Equivalently, it is a bijection $f$ such that $f(i) \neq i$ for all $i$.

**Theorem 6.3.** The number of derangements of size $n$ is

$$
\sum_{i=0}^n (-1)^i \frac{n!}{i!}.
$$

**Proof.** It turns out to be easier to count the number of permutations which are not derangements and then subtract that from the total number of permutations. For $i = 1, \ldots, n$, let $A_i$ be the set of bijections $f$ such that $f(i) = i$. Then the set of non-derangements is $A_1 \cup \cdots \cup A_n$. To apply inclusion-exclusion, we need to count the size of $A_{i_1} \cap \cdots \cap A_{i_j}$ for some choice of indices $i_1, \ldots, i_j$. This is the set of bijections $f: [n] \rightarrow [n]$ such that $f(i_1) = i_1, \ldots, f(i_j) = i_j$. The remaining information to specify $f$ are its values outside of $i_1, \ldots, i_j$, which we can interpret as a bijection of $[n] \setminus \{i_1, \ldots, i_j\}$ to itself. So there are $(n-j)!$ of them. So we get

$$
|A_1 \cup \cdots \cup A_n| = \sum_{j=1}^n (-1)^{j-1} \sum_{1 \leq i_1 < \cdots < i_j \leq n} |A_{i_1} \cap \cdots \cap A_{i_j}|
$$

$$
= \sum_{j=1}^n (-1)^{j-1} \sum_{1 \leq i_1 < \cdots < i_j \leq n} (n-j)!
$$

$$
= \sum_{j=1}^n (-1)^{j-1} \binom{n}{j} (n-j)!
$$

$$
= \sum_{j=1}^n (-1)^{j-1} \frac{n!}{j!}.
$$

Remember that we have to subtract this from $n!$. So the final answer simplifies as so:

$$
n! - \sum_{j=1}^n (-1)^{j-1} \frac{n!}{j!} = \sum_{j=0}^n (-1)^j \frac{n!}{j!}.
$$

□

The problem with formulas coming from inclusion-exclusion is the alternating sign. It can generally be hard to estimate the behavior of the quantity as $n$ grows. For example, binomial
coefficients \( \binom{n}{i} \) (for fixed \( i \)) limit to infinity as \( n \) goes to infinity. However, the alternating sum
\[
\sum_{i=0}^{n} (-1)^i \binom{n}{i}
\]
is 0. For derangements, we can use the following observation. We have a formula for the exponential function
\[
e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}.
\]
If we plug in \( x = -1 \) and only take the terms up to \( i = n \), then we get the number of derangements divided by \( n! \), i.e., the percentage of permutations that are derangements. From calculus, taking the first \( n \) terms of a Taylor expansion is supposed to be a good approximation for a function, so for \( n \to \infty \), the proportion of permutations that are derangements is \( e^{-1} \approx .368 \), or roughly 36.8%.

We can also use inclusion-exclusion to get an alternating sum formula for Stirling numbers.

**Theorem 6.4.** For all \( n \geq k > 0 \),
\[
S(n, k) = \frac{1}{k!} \sum_{i=0}^{k} (-1)^i \binom{k}{i} (k-i)^n = \sum_{i=0}^{k} (-1)^i \frac{(k-i)^n}{i!(k-i)!}.
\]

*Proof.* As we discussed before, \( k!S(n, k) \) counts the number of surjective functions \( f: [n] \to [k] \). So we will count this quantity. For \( i = 1, \ldots, k \), let \( A_i \) be the set of functions \( f: [n] \to [k] \) such that \( i \) is not in the image of \( f \). The surjective functions are the complement of \( A_1 \cup \cdots \cup A_k \) from the set of all functions (there are \( k^n \) total functions). To apply inclusion-exclusion, we need to count the size of \( A_1 \cap \cdots \cap A_j \) for \( 1 \leq i_1 < \cdots < i_j \leq k \). This is the set of functions so that \( \{i_1, \ldots, i_j\} \) are not in the image; equivalently, this is identified with the set of functions \( f: [n] \to [k] \setminus \{i_1, \ldots, i_j\} \), so there are \( (k-j)^n \) of them. So we can apply inclusion-exclusion to get
\[
|A_1 \cup \cdots \cup A_k| = \sum_{j=1}^{n} (-1)^{j-1} \sum_{1 \leq i_1 < \cdots < i_j \leq k} |A_{i_1} \cap \cdots \cap A_{i_j}|
\]
\[
= \sum_{j=1}^{n} (-1)^{j-1} \binom{k}{j} (k-j)^n
\]
\[
= \sum_{j=1}^{n} (-1)^{j-1} \binom{k}{j} (k-j)^n.
\]
Remember we have to subtract:
\[
k!S(n, k) = k^n - \sum_{j=1}^{n} (-1)^{j-1} \binom{k}{j} (k-j)^n = \sum_{j=0}^{n} (-1)^j \binom{k}{j} (k-j)^n.
\]
Now divide both sides by \( k! \) to get the first equality. The second equality comes from canceling the \( k! \) from the binomial coefficient. \( \square \)
7. Ordinary generating functions

7.1. Formal power series. A formal power series (in the variable $x$) is an expression of the form $A(x) = \sum_{n=0}^{\infty} a_n x^n$ where the $a_n$ are scalars (usually integers or rational numbers). Instead of writing the sum from 0 to $\infty$, we will usually just write $A(x) = \sum_{n \geq 0} a_n x^n$. We can treat these like infinite degree polynomials.

Let $B(x) = \sum_{n \geq 0} b_n x^n$ be a formal power series. The sum of two formal power series is defined by $A(x) + B(x) = \sum_{n \geq 0} (a_n + b_n) x^n$.

The product is defined by $A(x)B(x) = \sum_{n \geq 0} c_n x^n$, where $c_n = \sum_{i=0}^{n} a_i b_{n-i}$.

This is what you get if you just distribute like normal. As a special case, if $a_i = 0$ for $i > 0$, we just get $a_0 B(x) = \sum_{n \geq 0} a_0 b_n x^n$.

Addition and multiplication are commutative, so $A(x) + B(x) = B(x) + A(x)$ and $A(x)B(x) = B(x)A(x)$. They are also associative, so it is unambiguous how to add or multiply 3 or more power series.

**Example 7.1.** Let $A(x) = B(x) = \sum_{n \geq 0} x^n$. Then $A(x) + B(x) = \sum_{n \geq 0} 2x^n$, $A(x)B(x) = \sum_{n \geq 0} (n+1) x^n$. \hfill \box

A formal power series $A(x)$ is invertible if there is a power series $B(x)$ such that $A(x)B(x) = 1$. In that case, we write $B(x) = A(x)^{-1} = 1/A(x)$ and call it the inverse of $A(x)$. If it exists, then $B(x)$ is unique.

**Example 7.2.** Let $A(x) = \sum_{n \geq 0} x^n$ and $B(x) = 1 - x$. Then $A(x)B(x) = 1$, so $B(x)$ is the inverse of $A(x)$. For that reason, we will use the expression $\frac{1}{1-x} = \sum_{n \geq 0} x^n$.

However, the formal power series $x$ is not invertible: the constant term of $xB(x)$ is 0 no matter what $B(x)$ is, so there is no way that an inverse exists. \hfill \box

**Theorem 7.3.** A formal power series $A(x)$ is invertible if and only if its constant term is nonzero.

**Proof.**

It is important to emphasize that *formal* here means that we are not considering questions of convergence. We can take infinite sums and infinite products of formal power series as long as the coefficient of $x^n$ involves only finitely many multiplications and additions for each
n (adding 0 or multiplying by 1 infinitely many times is ok). For example, if we have formal power series \(A_1(x), A_2(x), \ldots\), then the infinite sum

\[A_1(x) + A_2(x) + A_3(x) + \cdots\]

is defined as long as the coefficient of \(x^n\) in \(A_i(x)\) is only nonzero for finitely many \(i\).

The precise conditions for infinite products are more tricky to characterize, but an important case that we will use often is when all of the constant terms are equal to 1 and, for each \(n > 0\), the coefficient of \(x^n\) in \(A_i(x)\) is nonzero only for finitely many \(i\).

Given two formal power series \(A(x)\) and \(B(x)\), suppose that \(A(x)\) has no constant term. Then we can define the composition by

\[(B \circ A)(x) = B(A(x)) = \sum_{n \geq 0} b_n A^n(x).\]

This looks like it could have problems with infinite sums, but because \(A(x)\) has no constant term, for each \(d\), the coefficient of \(x^d\) is 0 in \(A^n(x)\) whenever \(n > d\), so to compute the coefficient of \(x^d\) in the above expression, we only do finitely many multiplications and additions.

**Example 7.4.** Let \(d\) be a positive integer, \(A(x) = x^d\) and \(B(x) = \sum_{n \geq 0} x^n\). Then \(B(A(x)) = \sum_{n \geq 0} x^{dn}\). We can do this substitution into the identity

\[(1 - x)B(x) = 1\]

to get

\[(1 - x^d) \sum_{n \geq 0} x^{dn} = 1,\]

from which we conclude that

\[\frac{1}{1-x^d} = \sum_{n \geq 0} x^{dn}.\]

We can also take the derivative \(D\) of a formal power series. We define it as follows:

\[(DA)(x) = A'(x) = \sum_{n \geq 0} n a_n x^{n-1} = \sum_{n \geq 0} (n + 1) a_{n+1} x^n.\]

All of the familiar properties of derivatives hold:

\[D(A + B) = DA + DB\]
\[D(A \cdot B) = (DA) \cdot B + A \cdot (DB)\]
\[D(B \circ A) = (DA) \cdot (DB \circ A)\]
\[D(1/A) = -\frac{D(A)}{A^2}\]
\[D(A^n) = nD(A)A^{n-1}.\]

**Example 7.5.** We have \(\frac{1}{1-x} = \sum_{n \geq 0} x^n\). Taking the derivative of the left side gives \(\frac{1}{(1-x)^2}\). Taking the derivative of the right side gives \(\sum_{n \geq 0} nx^{n-1} = \sum_{n \geq 0} (n + 1)x^n\). We’ve already seen that these two expressions are equal.
How would we simplify \( B(x) = \sum_{n \geq 0} nx^n \)? We have a few options. First:

\[
B(x) = \sum_{n \geq 0} (n+1)x^n - \sum_{n \geq 0} x^n = \frac{1}{(1-x)^2} - \frac{1}{1-x} = \frac{1 - (1-x)}{(1-x)^2} = \frac{x}{(1-x)^2}.
\]

Or more directly:

\[
B(x) = x \sum_{n \geq 0} nx^{n-1} = x \frac{1}{(1-x)^2}.
\]

\( \square \).

We will use \( e^x \) to denote the formal power series \( \sum_{n \geq 0} \frac{1}{n!}x^n \).

7.2. Binomial theorem (general form).

7.3. Linear recurrence relations.

7.4. Combinatorial interpretations.

7.5. Catalan numbers.

7.6. Composition of ordinary generating functions.

8. Exponential generating functions

9. Partially ordered sets