Notes for Math 154
Steven V Sam

Contents

1. Induction 2
   1.1. Weak induction 2
   1.2. Strong induction 3
2. Elementary counting problems 4
   2.1. Bijection 4
   2.2. Sum and product principle 4
   2.3. Permutations and combinations 4
   2.4. Words 6
   2.5. Choice problems 7
3. Binomial theorem and generalizations 8
   3.1. Binomial theorem 8
   3.2. Multinomial theorem 10
4. Inclusion-exclusion 10
5. Graph theory, introduction 13
   5.1. Eulerian trails 14
   5.2. Directed graphs 15
   5.3. Hamiltonian cycles 16
   5.4. Graph isomorphisms 17
6. Trees 18
   6.1. Definition and basic properties 18
   6.2. Minimum-weight spanning trees 20
   6.3. Adjacency matrix 22
   6.4. Deletion-contraction 22
   6.5. Matrix-tree theorem 23
7. Coloring and matching 25
   7.1. Colorings 25
   7.2. Chromatic polynomials 25
   7.3. Bipartite graphs 27
   7.4. Matchings 27
8. Planarity 27
   8.1. Definitions 27
   8.2. Some equations and inequalities 28
   8.3. Obstructions to planarity 29
   8.4. The 5-color theorem 30
9. Pigeon-hole principle 30
   9.1. Basic version 30
   9.2. General version 31
10. Ramsey theory 32
    10.1. Ramsey’s theorem for graphs 32
    10.2. Ramsey’s theorem for hypergraphs 32
    10.3. Turán’s theorem 32
    10.4. Lower bounds on Ramsey numbers 32
1. 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 and discuss some different ways to use it.

1.1. **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 1.1.** 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. □

**Theorem 1.2.** 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 $\emptyset, \{*\}, \{U\}, \{*, U\}$, while those that do contain $x$ are $\{1\}, \{1, *\}, \{1, U\}, \{1, *, U\}$. 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.

1.2. 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.

Example 1.3. We know that every polynomial in $x$ is a linear combination of $1, x, x^2, x^3, \ldots$. We use strong induction to prove the statement that every polynomial is a linear combination of $1, (x-1), (x-1)^2, (x-1)^3, \ldots$.

Let $P(n)$ be the statement that every polynomial of degree $n$ is a linear combination of powers of $x-1$.

Then $P(0)$ is true: the only polynomials of degree 0 are constants, and we can write $c = c \cdot 1$.

Now assume that $P(0), P(1), \ldots, P(n)$ are all true. We will use them to show that $P(n+1)$ is true. Let $f(x)$ be an arbitrary polynomial of degree $n+1$. Let $\alpha$ be its leading term and define $g(x) = f(x) - \alpha \cdot (x-1)^{n+1}$. Then $g(x)$ is a polynomial of degree $\leq n$ since we have cancelled off the $x^{n+1}$ terms. So by strong induction, $g(x)$ is a linear combination of powers of $x-1$. If we add $\alpha \cdot (x-1)^{n+1}$ to this linear combination, we see that $f(x)$ is also a linear combination of powers of $x-1$. Since our argument applies to any polynomial of degree $n+1$, we have proved $P(n+1)$ is true. $\square$

Some examples to think about:

• There’s nothing particular about powers of $x$ or powers of $x-1$. For example, we can take powers of any linear polynomial $ax + b$ with $a \neq 0$. Adapt the argument to work for this generalization.
• Every positive integer can be written in the form $2^n m$ 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).
2. Elementary counting problems

2.1. Bijections. Given two functions \( f: X \rightarrow Y \) and \( g: Y \rightarrow X \), 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 \( x \in X \). In that case, the functions \( f \) and \( g \) are called bijections.

The following is a very important principle in counting arguments:

**Proposition 2.1.** If there exists a bijection between \( X \) and \( Y \), then \( |X| = |Y| \).

We can think of a bijection \( f \) between \( X \) and \( Y \) as a way of matching the elements of \( X \) with the elements of \( Y \). In particular, \( x \in X \) gets matched with \( y = f(x) \in Y \). Note that if \( x' \in X \) was also matched with \( y \), i.e., \( f(x') = f(x) \), then the existence of the inverse \( g \) shows us that \( g(f(x')) = g(f(x)) \), or more simply \( x = x' \). In other words, \( f \) is forced to be one-to-one (or injective). On the other hand, every element is matched with something, i.e., every \( y \in Y \) is of the form \( f(x) \) for some \( x \) because we can take \( x = g(y) \). In other words, \( f \) is forced to be onto (or surjective).

**Remark 2.2.** Bijections tell us that two sets have the same size without having to know how many elements are actually in the set.

Here’s a small example: imagine there is a theatre filled with hundreds of people and hundreds of seats. If we wanted to know if there are the same number of people as seats, we could count both. However, it would probably be much easier to just have each person take a seat and see if there are any empty or any standing people.

We’ll see some other examples later on.

2.2. Sum and product principle. Given two sets \( X \) and \( Y \) without any overlap, we have \( |X \cup Y| = |X| + |Y| \). We’ll just take this for granted, though you can call it the sum principle if you’d like a name for it.

The set of pairs of elements \((x, y)\) where \( x \in X \) and \( y \in Y \) is the Cartesian product \( X \times Y \). The related product principle says that \( |X \times Y| = |X| \cdot |Y| \). Again, we will take this for granted and not usually refer to it by name.

2.3. 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. More formally, if \(|S| = n\), then a permutation is a bijection \( f: S \rightarrow [n] \).

**Example 2.3.** There are 6 permutations of \( \{1, 2, 3\} \) which we list:

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

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:

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

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

**Theorem 2.4.** 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 $P(n+1)$. □

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 2.5. (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:

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

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 $\frac{15!}{10!5!}$.

(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. □

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 2.6. 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.

2.4. **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 2.7.** Say our alphabet is \( A = \{a, b\} \). The words of length \( \leq 2 \) are:

\[
\emptyset, \quad a, \quad b, \quad aa, \quad ab, \quad ba, \quad bb.
\]

**Theorem 2.8.** 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 2.9.** 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

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

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 \( \{0, 1\} \).

**Example 2.10.** How many pairs of subsets \( S, T \subseteq [n] \) satisfy \( S \subseteq T \)? We can also encode this problem as a problem about words. Let \( A \) be the alphabet of size 3 whose elements are: “in \( S \)”, “in \( T \) but not \( S \)” and “not in \( T \)”. Then each pair \( S \subseteq T \) gives a word of length \( n \) in \( A \): the \( i \)th entry of the word is the element which describes the position of \( i \). So there are \( 3^n \) such pairs.

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 2.11.** 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 2.11 that uses induction?

2.5. 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 2.12. 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 2.13. $\sum_{k=0}^{n} \binom{n}{k} = 2^n$.

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 2.14 (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 2.15.** 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. \(\square\)

**Theorem 2.16.** 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 \([n]\). I will omit the details that these are well-defined and inverse to one another. (But you should make sure that you could do this if asked.) \(\square\)

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?
- 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 \( x_1, x_2, \ldots, x_n \)?

3. Binomial theorem and generalizations

3.1. **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 3.1 (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 $\binom{n}{i}$. \hfill \qedsymbol

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 $x \cdot \binom{n}{k} x^{k-1} y^{n-k+1}$ 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. \hfill \qedsymbol

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 3.2. $2^n = \sum_{i=0}^{n} \binom{n}{i}.$

Proof. Substitute $x = y = 1$ into the binomial theorem. \hfill \qedsymbol

This says that the total number of subsets of $[n]$ is $2^n$ which is a familiar fact from before.

Corollary 3.3. 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. \hfill \qedsymbol

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

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 $x = y = 1$. \hfill \qedsymbol
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 $y$?

3.2. Multinomial theorem.

**Theorem 3.5** (Multinomial theorem). For $n, k \geq 0$, we have

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

**Proof.** The proof is similar to the binomial theorem. Consider expanding the product $(x_1 + x_2 + \cdots + x_k)^n$. 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 $\binom{n}{a_1, a_2, \ldots, a_k}$. □

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.

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

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

$$nk^{n-1} = \sum_{(a_1, a_2, \ldots, a_k)} a_1 \binom{n}{a_1, a_2, \ldots, a_k}.$$

4. **Inclusion-exclusion**

**Example 4.1.** 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](image)

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?

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.

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

What’s the pattern so far? We have to add up all of the sizes of the sets involved, then we subtract off the sizes of all ways of intersecting two of them, and then we add back the sizes of all ways of intersecting three of them. How does this continue? In general, the signs continue to alternate (add, subtract, add, subtract, ...) and at the \( j \)th step, we have to consider all sizes of intersecting \( j \) different sets.

**Theorem 4.2** (Inclusion-Exclusion). Let \( A_1, \ldots, A_n \) be finite sets. Then

\[
|A_1 \cup \cdots \cup A_n| = \sum_{j=1}^{n} (-1)^{j-1} \sum_{\{i_1, i_2, \ldots, i_j\}} |A_{i_1} \cap A_{i_2} \cap \cdots \cap A_{i_j}|,
\]

where the second sum is over all \( j \)-element subsets of \([n]\), i.e., we add up the sizes of all possible ways of intersecting \( j \) of the sets \( A_1, \ldots, A_n \).

**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 \( x \in A_{s_r} \). 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 \( S \):

\[
\sum_{T \subseteq S} (-1)^{|T|} = -\sum_{n=1}^{|S|} \left( \binom{|S|}{n} \right) (-1)^n.
\]

However, since \( |S| > 0 \), we have shown before that \( \sum_{n=1}^{|S|} \left( \binom{|S|}{n} \right) (-1)^n = 0 \), so the sum above is \( \binom{|S|}{0} = 1 \). \( \square \)
We can also prove this by induction on $n$. Can you see how?

We use this to solve the derangements problem. Here is a version of that problem: suppose we have $n$ people and they all put their hat into a box. The hats are redistributed to the people at random. What is the chance that nobody gets their own hat back? (We won’t solve this exactly, but see how to get a close approximation to the answer.)

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 4.3.** 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

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

$$= \sum_{j=1}^{n} (-1)^{j-1} \sum_{\{i_1, \ldots, i_j\}} (n-j)!$$

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

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

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

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

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 $e^{-1} \approx 0.368$, or roughly 36.8%.

5. Graph theory, introduction

The origin of graph theory is said to be Euler’s solution of the bridges of Königsberg problem.

(Taken from [https://commons.wikimedia.org/wiki/File:Konigsberg_bridges.png](https://commons.wikimedia.org/wiki/File:Konigsberg_bridges.png))

The problem was to find a path which starts and ends at the same point, and crosses each bridge exactly once.

There are a few things to note here: the lengths of the bridges are not important, nor are the sizes of the landmasses. All that really matters is: between any two landmasses, how many bridges connect them? So we can simplify the picture dramatically:

```
(Taken from [https://commons.wikimedia.org/wiki/File:Konigsberg_bridges.png](https://commons.wikimedia.org/wiki/File:Konigsberg_bridges.png))
```

This notion can be abstracted with the definition of a graph. Given a set $V$, let $\binom{V}{2}$ denote the set of 2-element subsets of $V$.

**Definition 5.1.** A graph $G$ is a pair of sets $(V, E)$ where $V$ is the set of vertices, and $E$ is a multiset from $V \cup \binom{V}{2}$, called the edges. The edges in $V$ are called loops. Given an edge, the vertices that uses are called its endpoints (they could be the same in the case of a loop). If there are no loops and each pair of vertices has at most one edge between them, then the graph is called simple.

We think of elements of $V$ as representing nodes, and the elements of $E$ in $\binom{V}{2}$ tell us which nodes are connected to each other (and how many times). We can think of the elements of $E$ in $V$ as representing self-connections, so we can draw them as loops beginning and ending at
the same node. Note there is nothing about locations of lengths in this definition. So while we can draw a graph as we have done above, such a pictorial representation is not unique.

**Example 5.2.** The picture above represents the graph with $V = \{A, B, C, D\}$ and

$E = \{\{A, B\}, \{B, D\}, \{B, C\}, \{A, C\}, \{A, C\}, \{C, D\}, \{C, D\}\}$. □

### 5.1. Eulerian trails.

**Definition 5.3.** A **walk** in a graph $G$ is a sequence $v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$ which alternates between vertices and edges such that for all $i = 1, \ldots, k$, $v_{i-1}$ and $v_i$ are the endpoints of $e_i$ (so they must be different unless $e_i$ is a loop). The beginning of the walk is $v_0$ and the ending is $v_k$. A walk is **closed** if $v_0 = v_k$. The **length** of the walk is $k$.

A **trail** is a walk that does not repeat edges (we treat multiple edges between two vertices as distinct). An **Eulerian trail** is a trail that uses every edge exactly once.

A **path** is a trail that does not repeat vertices (in particular, it has no loops). □

Now we can phrase the Königsberg problem in our new language: does the Königsberg graph have a closed Eulerian trail?

We need a few more definitions before answering this question.

**Definition 5.4.** Given a vertex $v$ in a graph $G$, its **degree** $\deg(v)$ is the number of edges connected to $v$, except that loops at $v$ must be counted twice. In other words, it is $2$ times the number of loops at $v$ plus the number of non-loop edges at $v$. □

Heuristically, we can think of an edge as being comprised of two “halves”, each one connected to one of the endpoints. This perspective is partially why we want to count loops twice: we really want to know how many half-edges are connected to $v$. Alternatively, imagine zooming in really close to a vertex. In that case, the two pieces of a loop would look like two separate edges.

**Definition 5.5.** A graph is **connected** if, for any two vertices $v$ and $w$, there exists a walk that begins at $v$ and ends at $w$.

In general, we can put an equivalence relation on the vertices of $G$, declaring that $v \sim w$ if there exists a walk beginning at $v$ and ending at $w$. The equivalence classes of this relation are the **connected components** of $G$. Each connected component is a connected graph. □

The following will be convenient to use repeatedly. I’ll leave it as homework to prove.

**Proposition 5.6.** If there is a walk from $x$ to $y$, then there is also a path from $x$ to $y$.

**Theorem 5.7 (Euler).** Let $G$ be a connected graph. Then $G$ has a closed Eulerian trail if and only if every vertex has even degree.

**Proof.** If $G$ has a closed Eulerian trail, then along this path, each vertex is entered the same number of times that it is exited. Since none of these edges are being repeated, we see that the number of “half-edges” being used must be even, and so $\deg(v)$ is even for all $v$.

Now we have to prove the opposite implication. So let $G$ be a connected graph such that $\deg(v)$ is even for all vertices $v$. We need to show that $G$ has a closed Eulerian trail. We will prove this statement by induction on the number of edges of $G$. If the number is $0$, then $G$ is a single vertex, and there is nothing to prove so our base case is done. Now suppose we have proven it for all connected graphs such that $\deg(v)$ is even for all $v$ that have $\leq n$ edges and let $G$ be a graph such that $\deg(v)$ is even for all $v$ and has $n + 1$ edges.
Step 1. Find any closed trail. To do this, begin with any edge. If it is a loop, we are done. Otherwise, it connects \( v \) and \( w \) with \( v \neq w \). Since \( \text{deg}(w) \) is even, there must be another edge that we can follow. Continue doing this until we hit a vertex we’ve seen before. Then our path will contain a closed trail. Let \( H \) be the graph consisting of the vertices and edges used on this trail.

Step 2. Delete the edges in \( H \) from \( G \). From the reasoning in the first part of the proof, this will change the degree of each vertex by an even amount, so the resulting graph \( G' \) still has all vertices with even degree. However, it may not be connected. But by induction, each connected component has a closed Eulerian trail.

Step 3. Each connected component of \( G' \) must share a vertex with one in \( H \). If not, pick a connected component that violates this condition and pick a vertex \( v \) in it and pick a vertex \( w \) in \( H \). In the original graph \( G \), there is a path between \( v \) and \( w \) since \( G \) is connected. If we follow this path from \( v \), there is a first time that it hits a vertex in \( H \), and the edges up to this point aren’t in \( H \) (because all edges of \( H \) go between two vertices in \( H \)), which means that some vertex in \( H \) is in the same connected component as \( v \).

So we can attach the closed Eulerian trails from each component to \( H \). The resulting path is a closed Eulerian trail for \( G \).

If a graph \( G \) has a closed Eulerian trail, we can “rotate” it so that it begins and ends at any particular vertex that we like. Formally, if our closed Eulerian trail is \( v_0, e_1, v_1, \ldots, v_k \), then we get another closed Eulerian trail \( v_1, e_2, \ldots, v_k, e_1, v_1 \). The important thing is that \( v_k = v_0 \) since the trail is closed. We can repeat this rotation as many times as needed to get the desired starting vertex and starting edge.

**Corollary 5.8.** Let \( G \) be a connected graph and let \( v, w \in V \) be different vertices. Then there is an Eulerian trail starting at \( v \) and ending at \( w \) if and only if \( \text{deg}(v), \text{deg}(w) \) are both odd, and \( \text{deg}(x) \) is even for all other vertices \( x \in V \).

**Proof.** Add a new edge \( f = \{v, w\} \) to \( G \) to get a graph \( G' \). Then the degree of every vertex of \( G' \) has even degree, so by the previous theorem, it has a closed Eulerian trail. By the above comments, we can rotate this trail until \( f \) is the starting edge. Delete \( f \) from this closed trail, and the result is an Eulerian trail of \( G \) which starts at one of \( v, w \) and ends at the other one. If it’s backwards, we can always reverse the trail. \( \square \)

Finally, a related statement about vertices of odd degree.

**Theorem 5.9.** Let \( G \) be a graph. There are an even number of vertices with odd degree.

**Proof.** Let \( v_1, \ldots, v_n \) be the vertices of \( G \) and let \( d_i = \text{deg}(v_i) \). Note that \( d_1 + \cdots + d_n \) is twice the number of edges, since each edge contributes 1 to the degree of each of its endpoints (still true for loops). In particular, \( d_1 + \cdots + d_n \) is even. If there are an odd number of vertices with odd degree, then the sum would be odd, and hence we know that it is not the case. \( \square \)

5.2. **Directed graphs.** Our definition of graph models when things are related by putting an edge between them. For example, our nodes could represent places and the edges could represent whether or not they are connected by a road. This suggests an equal relation between the two, but we might want to be able to talk about roads that only go in one direction, for example. For that, we can use the notion of a **directed graph.** Intuitively, this is the same as a graph except each edge now has an **orientation, i.e.,** it has a direction placed on it. Formally, instead of thinking of (non-loop) edges as a 2-element subset of the
vertex set, they are now elements of $V \times V$ (where the entries are unequal; loops then become the case when the entries are equal).

The definition of a walk in a directed graph is changed in the following way: it is now a sequence $v_0, e_1, \ldots, v_k$ where $e_i = (v_{i-1}, v_i)$, i.e., it has to be compatible with the orientations of the edges. All of the other definitions can then be adapted once we’ve made this change. We define a directed graph $G$ to be strongly connected if for all $v, w \in V$, there is a walk from $v$ to $w$.

Given a vertex $v$, its in-degree $\text{indeg}(v)$ is the number of edges of the form $(x, v)$ for some $x \in V$ and its out-degree $\text{outdeg}(v)$ is the number of edges of the form $(v, x)$ for some $x \in V$. Note that each loop at $v$ contributes 1 to its in-degree and 1 to its out-degree.

The directed version of Euler’s theorem is the following.

**Theorem 5.10.** Let $G$ be a strongly connected directed graph. Then $G$ has a closed Eulerian trail if and only if $\text{indeg}(v) = \text{outdeg}(v)$ for all $v \in V$.

The proof is fairly similar to the proof of the undirected version, so we won’t repeat it here.

5.3. **Hamiltonian cycles.** The last section was about walks that use all of the edges exactly once. Now we consider the dual idea where we use all of the vertices exactly once (but not necessarily all edges).

**Definition 5.11.** Let $G$ be a graph. A **cycle** is a closed trail such that each vertex is used at most once. In other words, it’s a closed walk that doesn’t repeat vertices or edges. A **Hamiltonian cycle** is a closed trail such that each vertex is used exactly once. A **Hamiltonian path** is a path that uses every vertex exactly once (the difference is that it is not required to be closed).

**Remark 5.12.** We would like to have a simple criterion which determines if a graph has a Hamiltonian cycle or path like Euler’s theorem. However, it is known that determining if a graph has a Hamiltonian cycle or path is an “NP-complete problem”. This means that there is unlikely to even be an efficient algorithm to determine this, so we probably don’t expect any simple condition.

While we don’t have a good condition for the existence of Hamiltonian cycles, we can give stronger conditions which imply their existence.

**Theorem 5.13.** Let $n \geq 3$. Let $G$ be a simple graph with $n$ vertices and assume that $\text{deg}(v) \geq n/2$ for all vertices $v$. Then $G$ has a Hamiltonian cycle.

**Proof.** Assume that the theorem is false. Then there exists a simple graph $G$ with $n \geq 3$ vertices such that $\text{deg}(v) \geq n/2$ for all vertices and $G$ does not have a Hamiltonian cycle. We want a maximal counterexample in the following sense: if there is an edge we can add to $G$ such that the result does not have a Hamiltonian cycle, then do it and keep doing it until it’s not possible. We end up with a counterexample, call it $G'$ with $n$ vertices which still has $\text{deg}(v) \geq n/2$ and no Hamiltonian cycle, with the additional property that adding any new edge will introduce a Hamiltonian cycle.

First, $G'$ must be missing at least one edge: if every edge is present, then there is clearly a Hamiltonian cycle. Let $x, y$ be a pair of vertices such that $\{x, y\}$ is not an edge. Let $G''$ be the result of adding $\{x, y\}$. By construction, $G''$ has a Hamiltonian cycle, and it must use $\{x, y\}$ (otherwise $G'$ has one too). In particular, if we consider the rest of the edges, we have
a Hamiltonian path starting at $x$ and ending at $y$ using only edges from $G'$. Let $z_1, \ldots, z_n$ be the order in which the vertices are visited on this path (so $z_1 = x$ and $z_n = y$).

Now define
\[
S = \{ i \mid 2 \leq i \leq n - 1, \{z_{i+1}, x\} \text{ is an edge in } G' \}
\]
\[
T = \{ i \mid 2 \leq i \leq n - 1, \{z_i, y\} \text{ is an edge in } G' \}.
\]
Then $|S| \geq \deg(x) - 1 \geq n/2 - 1$ (since we aren’t counting $z_2$ though it could be connected by an edge to $x$) and $|T| = \deg(y) \geq n/2$. Note that $S \cup T \subseteq \{2, \ldots, n-1\}$ and so $|S \cup T| \leq n - 2$, so we have (by inclusion-exclusion)
\[
|S \cap T| = |S| + |T| - |S \cup T| \geq n/2 - 1 + n/2 - (n - 2) = 1
\]
and so $S \cap T \neq \emptyset$. Pick $i \in S \cap T$. Then $\{x, z_{i+1}\}$ and $\{y, z_i\}$ are both edges of $G'$. That means that $G'$ does in fact have a Hamiltonian cycle: the order is $x = z_1, z_2, \ldots, z_i, y = z_n, z_{n-1}, \ldots, z_{i+2}, z_{i+1}$, which is a contradiction. \qed

The condition $\deg(v) \geq n/2$ is way too strong though. Consider the graph with vertices $[n]$ and edges $\{\{1, 2\}, \{2, 3\}, \{3, 4\}, \ldots, \{n, n+1\}, \{1, n\}\}$, so if you draw it, it is a cycle of length $n$. Then this has a Hamiltonian cycle but the degree of each vertex is only 2. You’ll see another example in homework where the condition is too strong to be applicable.

5.4. **Graph isomorphisms.** Graph isomorphisms formalize what it means for two graphs to “look the same” without literally being equal to each other. For example, consider the following 3 graphs

\[
\begin{array}{ccc}
1 & \rightarrow & 2 \\
\mid & \mid & \mid \\
3 & \rightarrow & 4
\end{array}
\hspace{1cm}
\begin{array}{ccc}
1 & \rightarrow & 4 \\
\mid & \mid & \mid \\
3 & \rightarrow & 2
\end{array}
\]

Formally, both graphs have vertex set $[4]$, the first graph has edges $\{\{1, 2\}, \{1, 3\}, \{2, 4\}, \{3, 4\}\}$ and the second graph has edges $\{\{1, 2\}, \{1, 4\}, \{3, 4\}, \{2, 3\}\}$, so they are not the same. However, consider the bijection $f: [4] \to [4]$ defined by $f(1) = 1$, $f(2) = 4$, $f(3) = 2$, $f(4) = 3$. Applying this function to the first graph gives the second graph.

**Definition 5.14.** Let $G = (V, E)$ and $H = (V', E')$ be graphs. A function $f: V \to V'$ is a graph isomorphism if it is a bijection, and for all $x, y \in V$, the number of edges between $x$ and $y$ equals the number of edges of $f(x), f(y) \in V'$.

If a graph isomorphism exists between $G$ and $H$, then $G$ and $H$ are **isomorphic**. If $G = H$, then a graph isomorphism is called an automorphism. \qed

If $f$ defines a graph isomorphism between $G$ and $H$, then $f^{-1}: V' \to V$ defines a graph isomorphism between $H$ and $G$. Furthermore, if we have an isomorphism from $H$ to a third graph $I$, then we can compose these isomorphisms to get one between $G$ and $I$.

Many properties of graphs are preserved by isomorphism, meaning that if $G$ has that property then so does any graph which is isomorphic to $G$. Heuristically, any property or quantity which does not depend on the specific way that the vertices are named will be preserved by isomorphism. Such properties are called **isomorphism invariants**. Some examples:

- Number of vertices
• Number of edges
• The multiset of degrees of vertices
• Whether or not a graph has a Hamiltonian cycle.

**Remark 5.15.** Given two graphs, there is a naive algorithm for determining whether or not they are isomorphic: first, check if they have the same number of vertices, and if so, test all bijections between their vertex sets. But this is a pretty bad algorithm. One can ask whether or not there is an efficient (polynomial-time) algorithm and this is an open problem. □

Automorphisms of $G$ capture its symmetries and can be used to simplify proofs and provide justification for the phrase “without loss of generality...”. Here is an example which illustrates what I mean.

**Example 5.16.** Consider the graph $G$ with vertices $[n]$ ($n \geq 3$) and edges $\{i, i+1\}$ for $i = 1, \ldots, n-1$ and $\{1, n\}$. Consider the following statement: every path $v_0, e_1, v_1, e_2, v_2$ of length 2 can be extended uniquely to a Hamiltonian cycle. When $v_0 = 1, v_1 = 2, v_2 = 3$, we can prove this directly since the unique way to extend to a Hamiltonian cycle is to continue traveling around the graph, i.e., take $v_i = i + 1$ for $i = 0, \ldots, n-1$ and $v_{n+1} = 1$.

But there are other paths of length 2. However, up to applying an automorphism, they can be turned into the one we just analyzed. Namely, “rotation” gives us an automorphism, i.e., $f(i) = i - 1$ for $i = 2, \ldots, n$ and $f(1) = n$ as does “reflection”, i.e., $f(1) = 1$ and $f(i) = n + 2 - i$ for $i \neq 1$. If we repeatedly apply rotation, we can turn $v_0$ into 1. Then either $v_1 = 2$ (in which case $v_2 = 3$ and we’re done) or $v_1 = n$, in which case one application of reflection will turn this into what we want. □

How many simple graphs with $n$ vertices are there up to isomorphism? The numbers for $n = 1, \ldots, 8$ are 1, 2, 4, 11, 34, 156, 1044, 12346. There probably isn’t a simple formula in general.

6. Trees

**6.1. Definition and basic properties.**

**Definition 6.1.** A **tree** is a connected simple graph that does not contain a cycle. A **forest** is a simple graph that does not contain a cycle. □

The difference is that forests are not required to be connected, but they can be. So every tree is a forest, but not vice versa. The connected components of a forest are all trees, so we can think of forests as being made up of trees (hence the name).

**Example 6.2.** Here are two examples of trees on 5 vertices:

![Trees](image)

We’ll give a few different ways to characterize trees.

**Theorem 6.3.** Let $G$ be a connected simple graph. Then $G$ is a tree if and only if for all vertices $x, y \in V$, there is exactly one path going from $x$ to $y$. □
Proof. Suppose that $G$ is a tree. Since $G$ is connected, for any vertices $x, y$, there is at least one walk going from $x$ to $y$, and hence there is at least one path going from $x$ to $y$. We claim that there is always exactly one path. If not, then let $x, y$ be a pair that has at least 2 paths between them. Furthermore, among all such pairs, pick $x, y$ so that the length of a path between them is as small as possible. Let $e_1, \ldots, e_k$ be the shortest possible path between them (in case of ties, pick any) and let $f_1, \ldots, f_\ell$ be the next shortest path between them. I claim that $e_1, \ldots, e_k, f_\ell, f_{\ell-1}, \ldots, f_1$ is a cycle, i.e., it doesn’t visit the same vertex more than once (except $x$). This is left as a homework problem.

Now suppose that $G$ is not a tree. Since $G$ is connected, it contains a cycle. We can split this into two pieces to get two different paths that go between the same pair of vertices. □

Definition 6.4. Let $G$ be a simple connected graph. Then $G$ is **minimally connected** if deleting any edge causes it to become disconnected. □

Theorem 6.5. Let $G$ be a connected simple graph. Then $G$ is a tree if and only if $G$ is minimally connected.

Proof. First suppose that $G$ is a tree. If $G$ has 1 vertex, there’s nothing to say. If it has more, then it has at least one edge, pick any of them $\{x, y\}$. This edge gives a path from $x$ to $y$, so by uniqueness (previous theorem), there isn’t another one. So if we delete this edge, there’s no way to get from $x$ to $y$, which means the result is disconnected. Since this is true for any edge, $G$ is minimally connected.

Now suppose that $G$ is not a tree. Since $G$ is connected, that means there is a cycle. If we delete any edge $\{x, y\}$ from that cycle, the resulting graph is still connected: for any two vertices $a, b$, if we had to use that edge to get from $a$ to $b$ before, then we can instead replace that part with the rest of the cycle. So $G$ is not minimally connected. □

Proposition 6.6. Let $G$ be a connected simple graph with $n$ vertices. Then $G$ has at least $n - 1$ edges.

Proof. Consider deleting all of the edges of $G$ and then re-adding them one at a time. At first, we have $n$ connected components since all of the vertices are isolated. Every time we add an edge, we are either connecting two vertices in the same component (in which case the number of components stays the same), or we are connected two vertices in different components (in which case the two components get merged into one). We conclude that to go from $n$ connected components to 1 connected component, we will need to add at least $n - 1$ edges. □

Corollary 6.7. Let $F$ be a forest with $n$ vertices and $m$ edges. Then it has $n - m$ connected components.

Proof. Go through the proof above. Each time we add an edge, we are decreasing the number of connected components by 1 (otherwise we’d introduce a cycle). □

Theorem 6.8. Let $G$ be a connected simple graph with $n$ vertices. Then $G$ is a tree if and only if $G$ has exactly $n - 1$ edges.

Proof. Suppose that $G$ is a tree. In particular, it is a forest so by the previous corollary, it has $n - m = 1$ connected components where $m$ is the number of edges. Solving for $m$ gives $m = n - 1$. □
Now suppose that $G$ has exactly $n - 1$ edges. Pick one and remove it. By the previous proposition, the result cannot be connected since it has too few edges. This means that $G$ is minimally connected, and hence a tree.

We have given several different properties of connected simple graph to be a tree. Here is the summary:

**Theorem 6.9.** Let $G$ be a connected simple graph. The following conditions are all equivalent:

1. $G$ is a tree.
2. $G$ has no cycles.
3. For all vertices $x, y \in V$, there is exactly one path starting at $x$ and ending at $y$.
4. $G$ is minimally connected.
5. $G$ has $n - 1$ edges.

In other words, if any of these properties hold, then all of them hold. If any of them fail, then they all fail.


**Definition 6.10.** Let $G = (V, E)$ be a graph. A subgraph of $G$ is a graph $G' = (V', E')$ such that $V' \subseteq V$ and $E' \subseteq E$.

A subgraph $G'$ is called a spanning tree of $G$ if $V' = V$ and $G'$ is a tree.

**Proposition 6.11.** If $G$ is connected, then it has at least one spanning tree.

**Proof.** If $G$ is a tree, then we are done. Otherwise, $G$ is not minimally connected, so we can remove some edge in such a way that the resulting graph is still connected. Continue doing this until we get down to a tree.

It’s possible to get a formula for the number of spanning trees, and we’ll see that later.

We take a detour to solve an algorithmic problem. Suppose we are given a graph $G = (V, E)$ together with a real-valued function on the edge set $w: E \rightarrow \mathbb{R}_{\geq 0}$ which takes non-negative values. Given a spanning tree $T \subseteq G$, define the weight $w(T)$ to be the sum of $w(e)$ over all edges $e \in T$. Our goal is to find a tree that has the smallest possible weight (there could be multiple solutions). Such a tree is called a minimum-weight spanning tree.

**Example 6.12.** The vertices might represent cities and the edges represent possible railroad tracks that could be built between them. The function $w$ could represent the cost for building that track. If we wanted to connect all of the cities by rail by spending as little money as possible, then we want a minimum-weight spanning tree.

[* Steven: put in picture *]

Here is Kruskal’s greedy algorithm. Let $G = (V, E)$ be a connected graph with $n$ vertices.

- Set $T = (V, \emptyset)$.
- If $T$ is a spanning tree, then terminate. Otherwise, consider all possible edges that we can add to $T$ so that the result does not have a cycle and add one that has the smallest possible weight to $T$. Repeat this step until $T$ is a spanning tree.

We have to be careful that the second step is well-defined. At each step, we have some subgraph $T$ that has no cycles and isn’t a spanning tree. Why is it always possible that we can find an edge to add to it so that we don’t create a cycle? (Left for homework.)
Theorem 6.13. The result of Kruskal’s greedy algorithm is a minimum-weight spanning tree.

In order to prove this, we need to know the “exchange property”.

Lemma 6.14 (Exchange property). Let \( F \) and \( F' \) be forests with the same vertex. Assume that \( F' \) has strictly more edges than \( F \) does. Then there exists an edge \( e \) of \( F' \) not belonging to \( F \) such that adding \( e \) to \( F \) results in a forest.

Proof. Suppose this is false. Then for all edges \( e \) of \( F' \), \( F \cup \{ e \} \) has a cycle. In particular, if \( x \) and \( y \) are connected by an edge in \( F' \), then they are in the same connected component of \( F \). Being in the same connected component is a transitive property, so this also implies that if \( x \) and \( y \) are in the same connected component of \( F' \), then they are also in the same connected component of \( F \). Let \( c, c' \) be the number of connected components of \( F, F' \), respectively. This means that \( c' \geq c \). However, for a forest, the number of connected components is the number of vertices minus the number of edges by Corollary 6.7. If \( n \) is the number of vertices, then we see that \( n - c' \leq n - c \) but this contradicts that \( F' \) has strictly more edges than \( F \).

Now we can prove the theorem about Kruskal’s greedy algorithm.

Proof of Theorem 6.13. Let \( T \) be the output of Kruskal’s greedy algorithm. Suppose it is not a minimum-weight spanning tree. Then there is another spanning tree \( H \) such that \( w(H) < w(T) \). Let \( h_1, \ldots, h_{n-1} \) be the edges of \( H \) ordered so that \( w(h_1) \leq \cdots \leq w(h_{n-1}) \) and similarly let \( t_1, \ldots, t_{n-1} \) be the edges of \( T \) ordered so that \( w(t_1) \leq \cdots \leq w(t_{n-1}) \).

Let \( S \) be the set of \( 1 \leq k \leq n - 1 \) such that \( \sum_{i=1}^{k} w(h_i) < \sum_{i=1}^{k} w(t_i) \). Note that \( S \) is non-empty because \( n - 1 \in S \). Let \( m \) be the minimal element of \( S \). Note that \( m > 1 \) because otherwise it means that \( w(h_1) < w(t_1) \) which contradicts how we picked the first edge in the algorithm. By definition of \( m \), we have

\[
\sum_{i=1}^{m-1} w(h_i) \geq \sum_{i=1}^{m-1} w(t_i) \quad \text{and} \quad \sum_{i=1}^{m} w(h_i) < \sum_{i=1}^{m} w(t_i).
\]

This is only possible if \( w(h_m) < w(t_m) \).

Let \( T' \) be the forest with all vertices of \( G \) and the edges \( t_1, \ldots, t_{m-1} \) and let \( H' \) be the forest with all vertices of \( G \) and the edges \( h_1, \ldots, h_m \). By the exchange property, there is some \( h_i \) such that \( h_i \) is different from the \( t_1, \ldots, t_{m-1} \) and the edges \( t_1, \ldots, t_{m-1}, h_i \) form a forest, i.e., has no cycle. However, \( w(h_i) \leq w(h_m) < w(t_m) \), which contradicts our choice of edge to add in the \( m \)th step of the greedy algorithm. In particular, \( H \) cannot exist, so \( T \) is a minimum-weight spanning tree.

Remark 6.15. The context of the greedy algorithm can be expanded quite a bit. The keypoint is the exchange property. Here is another familiar situation where it holds: let \( S, T \) be linearly independent sets of vectors in a vector space and assume that \( |T| > |S| \). Then there is always a choice of vector \( v \in T \) such that \( S \cup \{ v \} \) is linearly independent (try to prove it).

In particular, imagine we had a collection of vectors that spans our vector space and each one has a weight and our goal is to find a basis from this collection that minimizes the total weight. The greedy algorithm, adapted suitably, tells us that we can successively pick vectors of smallest possible weight that don’t create a linear dependence with the vectors we’ve already chosen and the result will have the desired property.
Structures that satisfy the exchange property are called matroids.

6.3. Adjacency matrix. If $G$ is a graph with $n$ vertices, then the adjacency matrix is an $n \times n$ matrix which encodes $G$. There are 2 versions, depending on if $G$ is a directed graph or a plain graph.

If $G$ is a graph with vertices $v_1, \ldots, v_n$, then set $a_{ij}$ to be the number of edges between $v_i$ and $v_j$. Define its adjacency matrix $A_G$ to be the $n \times n$ matrix whose $(i, j)$ entry is $a_{ij}$. Note that $A_G$ is a symmetric matrix since $a_{ij} = a_{ji}$.

[* Steven: example *]

If $G$ is a directed graph with vertices $v_1, \ldots, v_n$, then set $a_{ij}$ to be the number of edges between $v_i$ and $v_j$ that are going from $v_i$ to $v_j$ and again $A_G$ is the $n \times n$ matrix whose $(i, j)$ entry is $a_{ij}$. In general, $A_G$ need not be symmetric.

[* Steven: example *]

**Theorem 6.16.** Let $G$ be a graph (directed or not) with vertices $v_1, \ldots, v_n$. Let $A = A_G$ be its adjacency matrix. Then for all integers $k > 0$, the number of walks of length $k$ starting at $v_i$ and ending at $v_j$ is $(A^k)_{i,j}$.

**Proof.** We prove this by induction on $k$.

When $k = 1$, the $(i, j)$ entry of $A$ is defined so that it is the number of edges going from $v_i$ to $v_j$ (in both the directed and undirected cases).

Now suppose the result is known for $k$, we need to prove it for $k + 1$. Let $B = A^k$. Then

$$(A^{k+1})_{i,j} = (BA)_{i,j} = \sum_{\ell=1}^{\ell,k} B_{i,\ell} A_{\ell,j}$$

by definition of matrix multiplication. The term $B_{i,\ell} A_{\ell,j}$ counts the following: the number of pairs of walks from $v_i$ to $v_\ell$ of length $k$ and the number of walks frmo $v_\ell$ to $v_j$ of length $1$. For each such pair, we can concatenate the walks to get a walk from $v_i$ to $v_j$ of length $k + 1$. Every such walk is accounted for if we sum over all $\ell$ since the last step before reaching $v_j$ is some $v_\ell$ (and we’re including all of them). In particular, the sum counts the number of walks of length $k + 1$ from $v_i$ to $v_j$, so we’ve proven the statement for $k + 1$.

**Remark 6.17.** A general fact from linear algebra (probably not discussed in Math 18) is that an $n \times n$ symmetric matrix whose entries are all real numbers is always diagonalizable (this is the spectral theorem). In particular, this applies to $A_G$ in the undirected case. So we can write $A_G = BDB^{-1}$ where $D$ is a diagonal matrix whose entries are the eigenvalues of $A_G$. In particular, $A_G^k = BD^k B^{-1}$. So if we want general formulas for the number of walks of length $k$ as $k$ varies, it’s enough to diagonalize $A_G$. So we see that the eigenvalues of the adjacency matrix are relevant for counting walks, which is surprising.

In the directed case, $A_G$ need not be diagonalizable in general.

6.4. Deletion-contraction. Let $G$ be a graph and $e$ an edge of $G$ which is not a loop. There are two important operations (deletion and contraction) that we can perform on $G$ using $e$ and which are useful for certain kinds of induction proofs.

The deletion of $e$ is denoted $G \setminus e$ and is a graph with the same vertices as $G$, and the same edges, except we don’t use $e$.

The contraction of $e$ is denoted $G/e$. Let $e = \{x, y\}$. To define it, take the vertices of $G$, replace the two vertices $x, y$ with a single vertex that we will call $z$. For each edge in $G$ that does not use $x$ or $y$, add it into $G/e$. For each vertex $a$ different from $x$ and $y$, the number
of edges between $a$ and $z$ in $G/e$ is the number of edges between $a$ and $x$ plus the number of edges between $a$ and $y$. The number of loops at $z$ is the number of loops at $x$ plus the number of loops at $y$ plus the number of edges between $x$ and $y$ different from $e$.

To visualize this, pretend we are shrinking $e$ until $x$ and $y$ become the same point (hence the use of the word contraction).

Here’s a small example to illustrate. Say our graph is as follows (I put numbers on the edges to denote multiple edges):

$$G = \begin{array}{c}
\text{•} \\
\text{•} \\
4 \quad 3 \\
\text{•} \quad \text{•} \\
2
\end{array}$$

Let $e$ be one of the edges between the bottom two vertices. Then

$$G \setminus e = \begin{array}{c}
\text{•} \\
\text{•} \\
4 \quad 3 \\
\text{•} \quad \text{•} \\
2
\end{array} \quad \quad G/e = \begin{array}{c}
\text{•} \\
\text{•} \\
4 \quad 5 \\
\text{•} \quad \text{•} \\
2
\end{array}$$

Visually, $G/e$ is the result of shrinking the bottom edge of $G$ towards its midpoint. The two other bottom edges become loops.

6.5. **Matrix-tree theorem.** Let $\tau(G)$ (that letter is TAU) be the number of spanning trees of $G$. Our goal in this section is to find a formula for $\tau(G)$.

Let $v_1, \ldots, v_n$ be the vertices of $G$. Define $D_G$ be the $n \times n$ diagonal matrix whose $i$th diagonal is the number of edges connected to $v_i$ (loops only get counted once). The **Laplacian matrix** of $G$ is defined to be $L_G = D_G - A_G$. Next, given any $n \times n$ matrix $B$ and $1 \leq i \leq n$, let $B[i]$ be the matrix obtained from $B$ by deleting the $i$th column and $i$th row.

**Theorem 6.18** (Kirchhoff’s Matrix tree theorem). For any $1 \leq i \leq n$, we have $\tau(G) = \det(L_G[i])$. (By convention, the determinant of the $0 \times 0$ matrix is 1.)

Although it sounds more general to let $i$ be anything between 1 and $n$, if we remember that we’re allowed to order the vertices any way we please, then it’s enough to prove the theorem when $i = 1$ since reordering the rows and columns in the same way does not change the determinant of a matrix. We’ll usually assume $i = 1$ since it makes the notation a little simpler to explain (but remember it’s not less general).

To prove the matrix-tree theorem, we will prove that both sides satisfy the same recurrence, namely a deletion-contraction type recurrence.

**Proposition 6.19.** Let $e$ be an edge that connects $\{v_1, v_2\}$. Then

$$\det(L_G[1]) = \det(L_{G\setminus e}[1]) + \det(L_{G/e}[1]).$$

[* Steven: put example *]
Proof. Define an \((n - 1) \times (n - 1)\) matrix \(M\) as follows. The bottom right \((n - 2) \times (n - 2)\) matrix will be a copy of \(L_{G/e}[1]\). The first row will be \((1, 0, 0, \ldots, 0)\). The remaining entries are \(M_{i,1}\) where \(2 \leq i \leq n - 1\) and we set it to be \(L_{G/e}[1]_{i,1}\). Then \(\det(M) = \det(L_{G/e}[1])\) (see this by expanding \(\det(M)\) along the first row).

Next, we claim that \(M, L_{G/e}[1], L_{G}[1]\) are all identical in rows 2, \ldots, \(n - 1\). This is clear outside of the first column because they all record information about vertices \(v_3, \ldots, v_n\) which are exactly the same in \(G, G \setminus e, G/e\). For the first column, these involve information about edges between \(v_2\) and \(v_3, \ldots, v_n\). Again, this is the same between \(G\) and \(G \setminus e\) and we have defined \(M\) to have the same entries as \(L_{G/e}[1]\).

Finally, the first row of \(L_{G}[1]\) is the sum of the first row of \(M\) and the first row of \(L_{G/e}[1]\) (I'll leave this to you to fill in the details why). By multilinearity of determinants, this implies

\[
\det(L_{G}[1]) = \det(L_{G/e}[1]) + \det(M).
\]

But \(\det(M) = \det(L_{G/e}[1])\), so we get the identity we want. \(\square\)

Proposition 6.20. Let \(e\) be any edge of \(G\). Then

\[
\tau(G) = \tau(G \setminus e) + \tau(G/e).
\]

Proof. Write \(e = \{x, y\}\). \(\tau(G \setminus e)\) counts the number of spanning trees in \(G\) that do not use the edge \(e\) while \(\tau(G/e)\) counts the number of spanning trees in \(G\) that do use the edge \(e\).

The second requires some more explanation. First, let \(n\) be the number of vertices of \(G\) and let \(z\) be the new vertex formed by merging \(x\) and \(y\). If we have a spanning tree \(T\) of \(G\) that uses \(e\) and we contract \(e\), the remaining edges of \(T\) become a spanning tree of \(G/e\): \(T\) has \(n - 1\) edges and is connected, \(T/e\) has \(n - 2\) edges and \(n - 1\) vertices and is still connected. We can reverse this: by the way we defined it, there is a bijection between the edges of \(G/e\) that aren’t loops at \((x, y)\) and the edges of \(G\) whose endpoints aren’t \(\{x, y\}\). So if we have a spanning tree \(T’\) of \(G/e\), take the edges in \(G\) that correspond to the edges of \(T’\) and add \(e\) to get a spanning tree of \(G\). We just defined a bijection between spanning trees of \(G\) using \(e\) and spanning trees of \(G/e\). One thing to note: spanning trees never use more than one edge with the same endpoints, and never use loops; edges with the same endpoints as \(e\) correspond to loops in \(G/e\).

Every spanning tree of \(G\) either uses \(e\) or doesn’t, so we get the desired identity. \(\square\)

Now we can finish the proof.

Proof of Theorem 6.18. We prove the theorem by induction on the number of non-loop edges of \(G\). For the base case, let \(G\) be a graph with no non-loop edges. If the number of vertices is 1, then \(\tau(G) = 1\), and if it has more than 1 vertex, then \(\tau(G) = 0\). In either case, \(L_{G}[1]\) is the \(n \times n\) matrix consisting of 0’s. If \(n = 1\), then \(L_{G}[1]\) is the 0 \(\times 0\) matrix and so has determinant 1, which is what we want. Otherwise, \(L_{G}[1]\) is a 0 matrix of positive size so has determinant 0 for each \(i\), again which is what we want.

Now assume that we have proven the statement for all graphs with \(\leq n\) edges, and let \(G\) be a graph with \(n + 1\) edges. Order the vertices so that there is an edge \(e\) between \(v_1\) and \(v_2\).

Then both \(G/e\) and \(G \setminus e\) have less non-loop edges than \(G\), so know by induction that

\[
\tau(G \setminus e) = \det(L_{G \setminus e}[1]), \quad \tau(G/e) = \det(L_{G/e}[1]).
\]

However, we have previously proven the identities

\[
\tau(G) = \tau(G \setminus e) + \tau(G/e), \quad \det(L_{G}[1]) = \det(L_{G \setminus e}[1]) + \det(L_{G/e}[1]).
\]
Combining this with the previous identities gives us
\[ \tau(G) = \det(L_G[1]). \]

7. Coloring and matching

7.1. Colorings.

7.2. Chromatic polynomials. If \( G \) is a graph, and \( k \geq 0 \) is a non-negative integer, a **proper \( k \)-coloring** is a way to label the vertices of \( G \) with the numbers (colors) \( \{1, \ldots, k\} \) so that two vertices that are connected by an edge have different labels. We are free to use colors multiple times and we don’t have to use all of them. Let \( \chi_G(k) \) (that letter is CHI) be the number of ways to properly color the vertices with \( k \) colors. The **chromatic number** of \( G \), denoted \( \chi(G) \), is the smallest \( k \) such that \( G \) has a proper \( k \)-coloring.

**Lemma 7.1.** Let \( x, y \) be two vertices of \( G \) with exactly one edge \( e \) between them. Then
\[ \chi_G(k) = \chi_{G \setminus e}(k) - \chi_{G/e}(k). \]

**Proof.** By the definitions, a proper \( k \)-coloring of \( G \) is the same thing as a proper \( k \)-coloring of \( G \setminus e \) where \( x \) and \( y \) get different labels. On the other hand, proper \( k \)-colorings of \( G \setminus e \) where \( x \) and \( y \) receive the same color are naturally in bijection with proper \( k \)-colorings of \( G/e \): if \( z \) is the result of contracting \( x \) and \( y \), make its color the common color of \( x \) and \( y \). The identity \( \chi_G(k) = \chi_{G \setminus e}(k) - \chi_{G/e}(k) \) is a translation of what we just said: proper \( k \)-colorings of \( G \) are the same thing as proper \( k \)-colorings of \( G \setminus e \) once we subtract off all of those that give \( x \) and \( y \) the same color. \( \square \)

The assumption about \( x \) and \( y \) having exactly one edge between them is a little bit annoying, but it’s easy to get around. Let \( G \) be a graph. Construct a simple graph \( \overline{G} \) whose vertices are the same as \( G \) and where \( x \) and \( y \) have an edge in \( \overline{G} \) if they have at least one edge in \( G \). In other words, multiple edges in \( G \) get replaced by a single edge in \( \overline{G} \).

**Lemma 7.2.** \( \chi_G(k) = \chi_{\overline{G}}(k) \).

**Proof.** The definition of proper \( k \)-coloring only involves labeling vertices and the conditions on them only depend on whether or not two vertices have the same color if they’re connected by an edge (but we don’t care how many edges). \( \square \)

Now we’re ready to prove the main result:

**Theorem 7.3.** If \( G \) is a graph with \( n \) vertices, then \( \chi_G(k) \) is a polynomial in \( k \) of degree \( n \) (more precisely, there is a unique polynomial of degree \( n \) whose values agree with \( \chi_G(k) \) at all non-negative integer inputs \( k \)).

**Proof.** By Lemma 7.2, it is enough to prove this for simple graphs \( G \). We proceed by induction on the number of edges. If there are no edges in \( G \), then any labeling of the vertices is a proper \( k \)-coloring, so \( \chi_G(k) = k^n \) which is certainly a polynomial of degree \( n \).

Now assume we’ve proved this for graphs with \( < m \) edges and let \( G \) be a graph with \( m \) edges. Let \( e \) be an edge of \( G \). Then \( G \setminus e \) and \( G/e \) both have \( < m \) edges. So \( \chi_{G \setminus e}(k) \) is a polynomial in \( k \) of degree \( n \) and \( G/e \) is a polynomial in \( k \) of degree \( n - 1 \). By Lemma 7.1, \( \chi_G(k) = \chi_{G \setminus e}(k) - \chi_{G/e}(k) \), so \( \chi_G(k) \) is a polynomial in \( k \) of degree \( n \). \( \square \)
For notation, we will write \( \chi_G(z) \) for this polynomial \( z \) is now a variable \) and we will use \( k \) to denote non-negative integers. This is the **chromatic polynomial** of \( G \). Then the chromatic number is the smallest positive integer \( k \) such that \( \chi_G(k) \neq 0 \).

Here are some easy properties:

**Proposition 7.4.**

1. \( G \) has at least one vertex if and only if \( \chi_G(0) = 0 \).
2. \( G \) has at least one edge if and only if \( \chi_G(1) = 0 \). (The converse is clearly true.)
3. If \( G \) has an odd length cycle, then \( \chi_G(2) = 0 \). (The converse is also true, as we will see when we discuss bipartite graphs.)

How about a property that determines if \( \chi_G(3) = 0 \)? This is an NP-complete problem, so there likely isn’t a simple criterion to determine this for a general graph.

**Example 7.5.** Let’s compute \( \chi_G(z) \) for the square:

\[
\begin{array}{c}
\text{2} \\
\text{1} \\
\text{3} \\
\text{4}
\end{array}
\]

\( G = 
\begin{array}{c}
\text{1} \\
\text{2} \\
\text{3} \\
\text{4}
\end{array}
\)

It will follow that \( \chi(G) = 2 \) (or you can figure that out by staring at the square). Here are some different approaches:

1. For the first way, we just use the definition. If we want to properly \( k \)-color \( G \), then 1 can be colored anything, so there are \( k \) choices for it. Now the color on 2 and 4 have to be different from the color assigned to 1, so there are \( k - 1 \) choices for each. There are two cases to consider: if the colors of 2 and 4 are the same, then the color for 3 has \( k - 1 \) choices. If they’re not the same, then the color for 3 has \( k - 2 \) choices. So the total number of colorings is: \( k(k - 1)^2 + k(k - 1)(k - 2)^2 \). (The first term counts the number of colorings where 2 and 4 have the same color and the second counts the number of colorings where 2 and 4 have different colors.) We can simplify it to get

\[
\chi_G(k) = k(k - 1)(k^2 - 3k + 3).
\]

2. For the second way, we’ll use deletion-contraction. Let \( e = \{1, 4\} \). Then

\[
\begin{array}{c}
\text{2} \\
\text{1} \\
\text{3} \\
\text{4}
\end{array}
\]

\( G \setminus e = 
\begin{array}{c}
\text{1} \\
\text{2} \\
\text{3} \\
\text{4}
\end{array}
\)

Its chromatic polynomial is simple to compute: for a proper \( k \)-coloring, 1 has \( k \) choices, 2 has \( k - 1 \) choices (any color different from the one given to 1), similarly 3 has \( k - 1 \) choices, and similarly, 4 has \( k - 1 \) choices. So

\[
\chi_{G \setminus e}(k) = k(k - 1)^3.
\]
The contraction by $e$ is

$$G/e = \begin{array}{c}
2 \\
\downarrow \\
5 \\
\uparrow \\
3
\end{array}$$

I called the new vertex 5. This is also easy to compute: for a proper $k$-coloring, 5 has $k$ choices, 2 has $k - 1$ choices, and 3 has $k - 2$ choices (any color different from the one given to 2 and 5 which are different from each other). So

$$\chi_{G/e}(k) = k(k - 1)(k - 2).$$

So using Lemma 7.1, we get

$$\chi_G(k) = \chi_{G\setminus e}(k) - \chi_{G/e}(k)$$

$$= k(k - 1)^3 - k(k - 1)(k - 2)$$

$$= k(k - 1)(k^2 - 3k + 3).$$

\[\square\]

**Example 7.6.** Here are some families of graphs where we can give explicit formulas for $\chi_G(z)$ and $\chi(G)$. I won’t explain how to get the derivation, you should see if you can figure out how to do it.

(1) The **complete graph** on $n$ vertices is denoted $K_n$ and is defined so that every pair of vertices has an edge between them. Then

$$\chi_{K_n}(z) = z(z - 1)(z - 2) \cdots (z - n + 1),$$

$$\chi(K_n) = n.$$

(2) The cycle $C_n$ of length $n$ has vertices $v_1, \ldots, v_n$ and edges $\{i, i+1\}$ for $i = 1, \ldots, n - 1$ and $\{1, n\}$. Then

$$\chi_{C_n}(z) = \begin{cases} (z - 1)^n + (z - 1) & \text{if } n \text{ is even} \\ (z - 1)^n - (z - 1) & \text{if } n \text{ is odd} \end{cases},$$

$$\chi(C_n) = \begin{cases} 2 & \text{if } n \text{ is even} \\ 1 & \text{if } n = 1 \\ 3 & \text{if } n \text{ is odd and } n \geq 3 \end{cases}.$$

(3) If $G$ is a tree with $n$ vertices, then

$$\chi_G(z) = z(z - 1)^{n-1},$$

$$\chi(G) = \begin{cases} 1 & \text{if } n = 1 \\ 2 & \text{if } n > 1 \end{cases}.$$

\[\square\]

7.3. Bipartite graphs.

7.4. Matchings.
8. Planarity

8.1. Definitions. A planar graph is, roughly speaking, a graph which can be drawn in the plane (for example, a piece of paper) in such a way that edges do not overlap each other. The basic instructive example is the complete graph $K_4$. Sometimes it is drawn as follows:

![Graph Diagram]

and the two diagonal edges overlap. However, here are two different ways to draw it so that none of the edges overlap:

![Alternative Graph Diagrams]

In the left drawing, one of edges is “curved”: this is allowed. In the right drawing, all of the edges are straight lines. It is a theorem that planar graphs can always be drawn so that the edges are all straight lines, but we won’t use this and it takes a bit of effort to prove this, so we won’t say anything more about it.

It’s easy enough to see that if $G$ is the simple graph associated to $G$, then $G$ is planar if and only if $G$ is planar.

When you draw a graph in the plane, you have separated the plane into different regions. Another way to say this: if you delete the graph from the plane, the regions are the different connected pieces. These are usually called faces. In the right drawing of $K_4$ above, the faces are the 3 inside triangles and then there is 1 “outside” face, so 4 in total. You can also see there are 4 faces in the left drawing. We’ll see shortly that the number of faces only depends on the isomorphism type of the graph.

Once you draw a graph $G$ in the plane, there is something called the dual graph $G^*$: the vertices are the faces of $G$, and two faces are connected by an edge if they share an edge. This notion is useful for translating the problem of coloring maps into problems of coloring graphs (the countries are faces), but otherwise we won’t do much with this definition.

8.2. Some equations and inequalities.

**Theorem 8.1 (Euler).** Let $G$ be a connected planar graph with $n$ vertices, $m$ edges, and $f$ faces. Then $n - m + f = 2$.

This is Theorem 12.2 of Bóna so we don’t prove this again here.

**Corollary 8.2.** Let $G$ be a planar graph with $c$ connected components, $n$ vertices, $m$ edges, and $f$ faces. Then $n - m + f = c + 1$.

**Proof.** We’ll do induction on $c$, the case $c = 1$ being Theorem 8.1. Let $G$ be a graph with $c$ connected components $G_1, \ldots, G_c$. Say $G_c$ has $n'$ vertices, $m'$ edges, and $f'$ faces. Let $H$ be the result of removing $G_c$. Then $H$ is still planar and has $n - n'$ vertices, $m - m'$ edges, but $f - (f' - 1)$ faces: we subtract $f'$ - 1 because $G_c$ and $H$ have the same “outside” face. By induction: $n - n' - (m - m') + f - (f' - 1) = c$. By Theorem 8.1, we also know that $n' - m' + f' = 2$. Adding these two equations gives $n - m + f = c + 1$. □
A simple planar graph with $n$ vertices can’t have arbitrarily many edges. We can give a bound in terms of $n$, but can do much better if we use the girth. For $n \geq 3$, let $C_n$ be the cycle graph with $n$ vertices (so it has $n$ vertices $v_1, \ldots, v_n$ and $n$ edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_n, v_1\}$).

**Definition 8.3.** A simple graph $G$ has **girth** at least $g$ if it does not contain a subgraph isomorphic to $C_3, C_4, \ldots, C_{g-1}$. (If $g = 3$, this condition is vacuous, so every graph has girth $\geq 3$.) We say that $G$ has girth equal to $g$ if it has girth at least $g$ and also contains a subgraph isomorphic to $C_g$. If $G$ has no cycles, then it has infinite girth. □

If $H$ is a subgraph of $G$, then the girth of $H$ is at least as big as the girth of $G$.

**Theorem 8.4.** Let $G$ be a simple planar graph with $n$ vertices, $m$ edges, and finite girth $g$. Then

\[ m \leq \frac{g}{g-2}(n-2) \]

 proves. Let $c$ be the number of connected components of $G$. Since $G$ has finite girth $g$, the boundary of every face of $G$ has $\geq g$ edges (the boundary of the outside face is the boundary of $G$). Each edge is on the boundary of at most 2 faces, so we conclude that $2m \geq gf$. By Corollary 8.2,

\[ f = c + 1 - n + m \geq 2 - n + m, \]

so $2m \geq g(2 - n + m)$. Rearranging terms we get $g(n - 2) \geq (g - 2)m$. Now divide by $g - 2$ (note that $g - 2 \geq 1$ because $g \geq 3$ by the way we defined girth). □

If $G$ has infinite girth, then it’s a forest, and we already know that $m \leq n - 1$.

**Corollary 8.5.** Let $G$ be a simple planar graph with $n$ vertices and $m$ edges. If $n \geq 3$, then $m \leq 3n - 6$.

 proves. If $G$ is a forest, then $m \leq n - 1 \leq 3n - 6$ (the second inequality holds because $n \geq 3$). Otherwise, $G$ has finite girth $g \geq 3$ so $m \leq \frac{g}{g-2}(n-2)$ by Theorem 8.4. But $\frac{g}{g-2} \leq 3$, so we can simplify that to $m \leq 3(n - 2)$. □

The following result will be useful when we study colorings of planar graphs:

**Corollary 8.6.** If $G$ is a simple planar graph, then it has a vertex with degree $\leq 5$.

 proves. Suppose not. Then every vertex of $G$ has degree $\geq 6$ (in particular, $G$ has at least 7 vertices). Let $m$ be the number of edges and $n$ be the number of vertices. By the handshake lemma, $2m = \sum_v \deg(v) \geq 6n$ where the sum is over all vertices $v$. In particular, $m \geq 3n$, which contradicts Corollary 8.5 since $n \geq 7$. □

### 8.3. Obstructions to planarity.

Recall that $K_n$ is the complete graph on $n$ vertices: every pair of vertices has an edge between them. Also recall the complete bipartite graph $K_{n,m}$. It has vertices $\{x_1, \ldots, x_n, y_1, \ldots, y_m\}$ and the edges $\{x_i, y_j\}$ for all $i = 1, \ldots, n$ and $j = 1, \ldots, m$.

Here are two important examples of non-planar graphs:

**Example 8.7.** $K_5$ is not planar: it has girth 3, 5 vertices, and 10 edges. If it were planar, then Theorem 8.4 implies $10 \leq 3(5 - 2)$ which is false. □

**Example 8.8.** $K_{3,3}$ is not planar: it has girth 4 (it has no odd cycle because it is bipartite and $x_1, y_1, x_2, y_2$ is a 4-cycle), 6 vertices, and 9 edges. If it were planar, then Theorem 8.4 implies $9 \leq 2(6 - 2)$ which is false. Corollary 8.5 only gives a bound of 9 on the number of edges, which isn’t good enough to show that $K_{3,3}$ is not planar. □
Given a graph \( G \) and an edge \( e = \{x, y\} \), the **subdivision** of \( G \) along \( e \) is a new graph obtained as follows: add a new vertex \( z \), and edges \( \{x, z\} \) and \( \{y, z\} \) and remove \( e \) (pictorially, we’ve replaced \( e \) with two edges. Here’s an example where we’re subdividing the left edge:

Starting with any graph \( G \), we can subdivide edges all we like (including the new ones), the resulting set of graphs are called subdivisions of \( G \).

It’s clear that if \( G \) isn’t planar, then neither is any subdivision of it (all we really did is add vertices along edges). Furthermore, if \( G \) contains a subgraph which isn’t planar, then \( G \) also can’t be planar.

Combining what we know, if \( G \) contains a subgraph which is isomorphic to a subdivision of \( K_5 \) or \( K_{3,3} \), then it isn’t planar. The converse is also true, but we won’t prove it:

**Theorem 8.9** (Kuratowski, 1930). A simple graph \( G \) is planar if and only if it does not have a subgraph which is isomorphic to a subdivision of \( K_5 \) or \( K_{3,3} \).

There is a variant which is also convenient for testing planarity. Given a graph \( G \) and an edge \( e \), we can delete it to get \( G \setminus e \) or contract it to get \( G/e \). We will call both of these **graph minors** of \( G \), and more generally, a graph minor of \( G \) is any graph which can be obtained from \( G \) by repeatedly deleting edges and vertices, and also contracting edges.

A moment’s thought tells us that if \( G \) has a graph minor which is non-planar, then \( G \) also can’t be planar. So we conclude that if \( G \) has a graph minor isomorphic to \( K_5 \) or \( K_{3,3} \), then \( G \) is not planar. Again, the converse is also true, but we won’t prove it:

**Theorem 8.10** (Wagner, 1937). A simple graph \( G \) is planar if and only if it does not have a graph minor which is isomorphic to \( K_5 \) or \( K_{3,3} \).

8.4. The 5-color theorem.

9. **Pigeon-hole principle**

9.1. **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 9.1** (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 \( n > k \).

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

**Example 9.2.** • 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:

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

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:

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

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 $\sqrt{(1/3)^2 + (1/3)^2} = \sqrt{2}/3 \approx 0.4714$.

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.

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

**Theorem 9.3** (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 $n > rm$. \hfill \square

Example 9.4. • 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:

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

10. Ramsey theory

10.1. Ramsey’s theorem for graphs.

10.2. Ramsey’s theorem for hypergraphs.

10.3. Turán’s theorem.

10.4. Lower bounds on Ramsey numbers.