Math 3322: Graph Theory

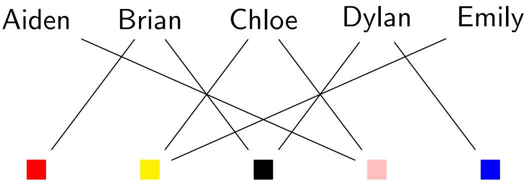
Chapters 8–10

Mikhail Lavrov (mlavrov@kennesaw.edu)

Spring 2021

Favorite colors

Five teenagers with attitude are chosen to defend the Earth from the attacks of an evil sorceress. Before they can take on their superpowers, they must each pick a color. But they're very picky about the colors they're willing to settle for:



What is the largest team of superheroes that can be formed?

Matchings in graphs

A **matching** in a graph is a set of independent edges: edges that share no endpoints. We are often especially interested in finding a **maximum matching** which has the largest size possible.

We write $\alpha'(G)$ for the size of a maximum matching in G; the $'$ indicates that it's the edge version of a problem (we'll look at the vertex version of the problem later).

We'll begin by looking at the maximum matching problem in bipartite graphs. Here, if G has bipartition (A, B), each edge in the matching has one endpoint in A and one in B, so the matching pairs up some vertices in A with adjacent vertices in B.

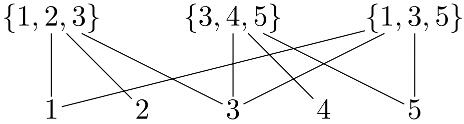
This can model problems about pairing one type of object to another: workers to tasks, rooms to occupants, meetings to days, etc.

Systems of distinct representatives

For some sets $S_1, S_2, \ldots, S_n$, a **system of distinct representatives** is a choice of elements $x_1 \in S_1, x_2 \in S_2, \ldots, x_n \in S_n$ that are all distinct: no two chosen elements are equal.

Example. Take $S_1 = \{1, 2, 3\}$, $S_2 = \{3, 4, 5\}$, and $S_3 = \{1, 3, 5\}$. Then $x_1 = 1, x_2 = 3, x_3 = 5$ is a system of distinct representatives.

This can be modeled as a matching problem in a bipartite graph! Put the sets $S_1, \ldots, S_n$ on one side, and their elements on the other.

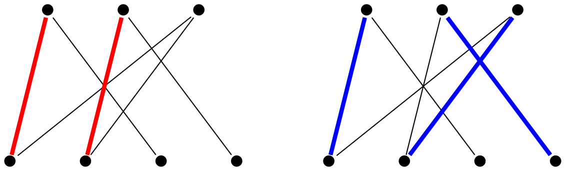


A system of distinct representatives is a matching of size n.

Greedy algorithms

A greedy algorithm is, informally, any strategy for solving a problem that makes decisions without thinking ahead.

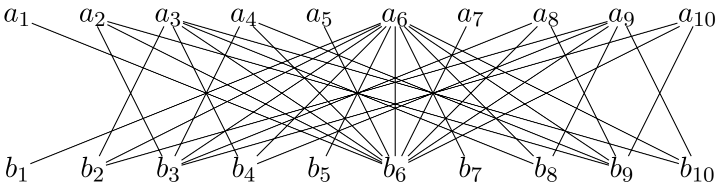
The bipartite matching problem has a natural greedy strategy. Let $A \cup B$ be the bipartition of G. Go through the vertices of A one at a time; for each $a \in A$, let b be the first unmatched neighbor of a, and add edge ab to the matching.



In red, we see the matching that the greedy algorithm finds. It is not always optimal! In blue, we see a maximum matching.

Multiples of six

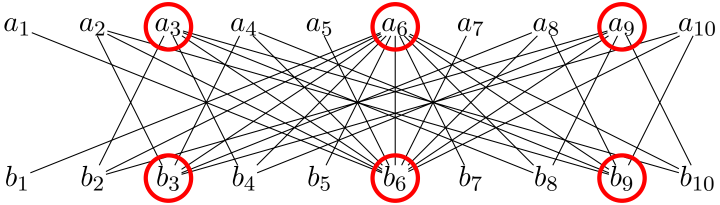
Consider the matching problem in the following bipartite graph: it has $A = \{a_1, \ldots, a_{10}\}$, $B = \{b_1, \ldots, b_{10}\}$, and an edge $a_i b_j$ whenever $i \cdot j$ is a multiple of 6.



A maximum matching in this graph (there are many) has size 6. Can you prove it?

Bottlenecks and vertex covers

If you play around with this problem for a while, you notice that multiples of 3 are the bottleneck. Every edge needs to include at least one of them; there are only 6, so once they run out, no other edge can be added to the matching.



We call such a set of vertices a **vertex cover**. Formally, a vertex cover in a graph is a set of vertices that includes at least one endpoint of every edge.

Vertex covers and matchings

We write $\beta(G)$ for the number of vertices in a **minimum vertex cover** of G: a vertex cover with as few vertices as possible.

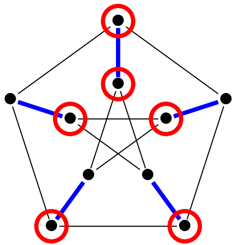
Theorem. If G is any graph, $M \subseteq E(G)$ is any matching, and $U \subseteq V(G)$ is any vertex cover, then $|M| \leq |U|$. In particular, $\alpha'(G) \leq \beta(G)$.

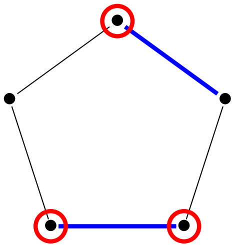
Proof. Every edge of M contains at least one vertex of U, and they can't share: two edges of M can't have the same vertex of U. So for the edges $e_1, e_2, \dots, e_k$ of M there must be just as many distinct vertices $v_1, v_2, \dots, v_k$ in U such that v_i is an endpoint of e_i. There may also be other vertices in U, so all we know is that $|M| \leq |U|$.

In particular, this holds when M is a maximum matching and U is a minimum vertex cover. In that case, $\alpha'(G) = |M| \leq |U| = \beta(G)$. □

Are these always equal?

Are $\alpha'(G)$ and $\beta(G)$ equal for any graph G? No! Here are two examples:





In the next lecture, we will see that they are equal for bipartite graphs. This makes the bipartite matching problem easier than the matching problem for general graphs.

Kőnig's theorem

Last time: In all graphs G, $\alpha'(G)$ (the size of a maximum matching) is at most $\beta(G)$ (the number of vertices in a minimum vertex cover).

Theorem (Kőnig). If G is bipartite, then $\alpha'(G) = \beta(G)$.

Proof strategy. Let (A, B) be the bipartition of G. We build a bigger graph H with a vertex s adjacent to all of A, and a vertex t adjacent to all of B, and apply Menger's theorem to H.

We will show:

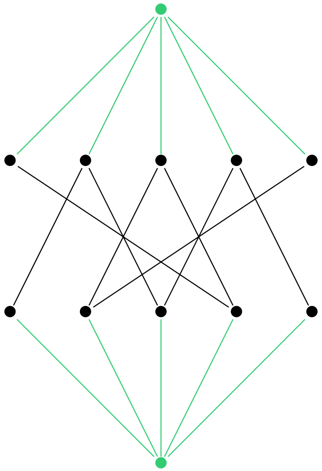
- $\alpha'(G)$ is the maximum number of internally disjoint $s - t$ paths;
- $\beta(G)$ is the minimum number of vertices in an $s - t$ cut.

From Menger's theorem, these are equal, so $\alpha'(G) = \beta(G)$. □

Matchings and $s - t$ paths

Our goal: show that matchings in G correspond to families of internally disjoint paths in H.

- In one direction: for every edge ab in a matching M, take the path (s, a, b, t) in H. These are internally disjoint because edges in M share no vertices.
- In the other direction: each path must use at least one edge of G, and taking one edge from each path gives a matching.

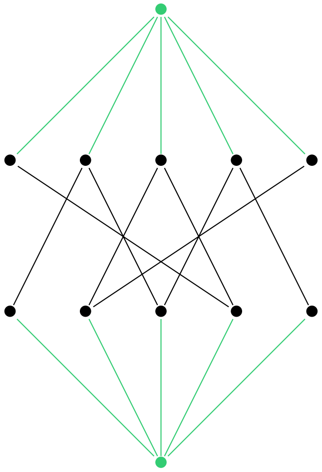


Vertex covers and $s - t$ cuts

Our goal: show that vertex covers in G correspond to $s - t$ cuts in H.

- In one direction: if U is a vertex cover, deleting U destroys all edges of G, disconnecting s from t.
- In the other direction: If U is a cut, then for every edge $ab \in E(G)$, either $a \in U$ or $b \in U$. Otherwise, (s, a, b, t) would be a path in $H - U$.

The previous slide shows $\alpha'(G) = \kappa(s, t)$. This slide shows $\beta(G) = \kappa(s, t)$. Therefore $\alpha'(G) = \beta(G)$. $\square$



Example of Kőnig's theorem

Corollary. Every k-regular bipartite graph has a perfect matching (a matching that covers every vertex).

Proof. Let G be a k-regular bipartite graph, and let $V(G) = A \cup B$ be the bipartition. to reassure ourselves, let's first prove $|A| = |B|$.

How many edges are in G? We get $k \cdot |A|$ by counting every edge out of every vertex in A (and that's all the edges). From B's point of view, we get $k \cdot |B|$. So $k \cdot |A| = k \cdot |B|$, or $|A| = |B|$.

Every vertex cover in G has at least $|A|$ vertices. That's because each vertex can cover at most k edges, and there are $k|A|$ edges.

By Kőnig's theorem, the maximum matching has at least $|A|$ edges, covering all of A (and all of B). This is a perfect matching! $\square$

Perfect matchings

There are two big theorems about matchings in bipartite graphs.

1. One is Kőnig's theorem. This gives a minimization problem (find $\beta(G)$) to balance the maximization problem (find $\alpha'(G)$).

   If we solve both, we know for sure that both our answers are correct.

2. The other is Hall's theorem. This is all about answering the yes-or-no question: does G have a **perfect** matching?

   Hall's theorem gives a necessary and sufficient condition for a perfect matching to exist.

   More generally: if the bipartition is $V(G) = A \cup B$, Hall's theorem tells us when there is a matching that covers all of A.

Hall's theorem

What are some easy ways to know that a bipartite graph G has no matching that covers all of A?

- If there is a vertex $v \in A$ of degree 0, then we're definitely in trouble: that vertex can never be covered by a matching.
- If $|A| > |B|$, then we'll run out of vertices in B.

To generalize these, define $N(S)$ for a set $S \subseteq V(G)$ to be the set of all vertices adjacent to at least one element of S.

If $|N(S)| < |S|$ for some $S \subseteq A$, then we can't cover all of S: and therefore not all of A!

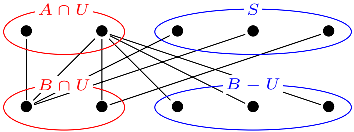
Theorem (Hall). If $|N(S)| \geq |S|$ for all $S \subseteq A$, then G has a matching of size $|A|$ (which covers all of A).

Proof of Hall's theorem

We will use the contrapositive. Suppose that G has no matching of size $|A|$; we'll find an S for which Hall's condition is violated.

By Kőnig's theorem, G has a vertex cover U with $|U| < |A|$.

Since U is a vertex cover, no vertex in $A - U$ can have a neighbor in $B - U$. So let $S = A - U$; all neighbors of S must be in $B \cap U$.

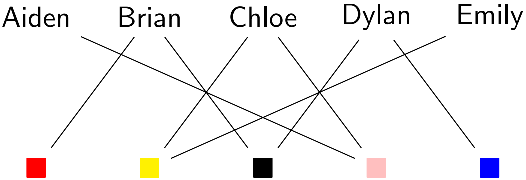


We know $|A \cap U| + |S| = |A|$ and $|A \cap U| + |B \cap U| = |U| < |A|$. From this, we can deduce that $|B \cap U| < |S|$.

Since $N(S) \subseteq B \cap U$, $|N(S)| < |S|$, so Hall's condition fails. $\square$

Example of Hall's condition

Let's return to the example we started discussing matchings with. Will all five teenagers with attitude be able to become superheroes?



No: this violates Hall's condition. Let $S = \{\text{Aiden}, \text{Chloe}, \text{Emily}\}$. Then $N(S) = \{\blacksquare, \blacksquare\}$. $|S| = 3$ but $|N(S)| = 2$.

We can do the same with vertex covers. $U = \{\text{Brian}, \text{Dylan}, \blacksquare, \blacksquare\}$ is a vertex cover, so there is no matching larger than $|U| = 4$.

Hall's theorem example

Claim. Every bipartite graph G with 99 vertices on each side, and minimum degree $\delta(G) \geq 50$, has a perfect matching.

Proof. We check Hall's condition for a set $S \subseteq A$. There are two cases:

1. If $|S| \leq 50$, even one vertex in S has at least 50 neighbors, so $|N(S)| \geq 50 \geq |S|$.
2. If $|S| > 50$, every vertex in B has at least 50 neighbors, and there are fewer than 50 vertices not in S. So every vertex in B has a neighbor in S, and $|N(S)| = 99 \geq |S|$.

Because Hall's condition holds for all S, we have a matching of size 99: a perfect matching. □

Edge covers and independent sets

We define two more optimization problems in a graph:

1. An **edge cover** in a graph is a set of edges whose endpoints include every vertex.

   (Note: this is impossible when isolated vertices exist.)

   We write $\beta'(G)$ for the size of a minimum edge cover in G.

2. An **independent set** in a graph is a set of vertices with no edges between any of them.

   We write $\alpha(G)$ for the number of vertices in a maximum independent set in G.

A summary of the four parameters

Here is a summary of the four parameters we have defined.

Parameter	Min/max	Of what?	Condition
$\alpha(G)$	$\max$	vertices	no two are adjacent
$\alpha'(G)$	$\max$	edges	no two are incident
$\beta(G)$	$\min$	vertices	that cover every edge
$\beta'(G)$	$\min$	edges	that cover every vertex

The corresponding sets are called **independent set**, **matching**, **vertex cover**, and **edge cover**.

If $L(G)$ is the line graph of G, then $\alpha'(G) = \alpha(L(G))$.

However, $\beta'(G)$ is not the same as $\beta(L(G))$; there's a similarity in "flavor text" but the problems are less closely related.

Gallai's identity #1

Theorem. For all n-vertex graphs G, $\alpha(G) + \beta(G) = n$.

Proof. Vertex covers and independent sets are closely related.

If $U \subseteq V(G)$ and $I = V(G) - U$, then

U is a vertex cover $\iff$ all edges of G have at least one endpoint in U

$\iff$ no edge of G has both endpoints in I

$\iff$ I is an independent set.

If U is a vertex cover of size $\beta(G)$, its complement is an independent set of size $n - \beta(G)$, so $\alpha(G) \geq n - \beta(G)$, or $\alpha(G) + \beta(G) \geq n$.

If I is an independent set of size $\alpha(G)$, its complement is a vertex cover of size $n - \alpha(G)$, so $\beta(G) \leq n - \alpha(G)$, or $\alpha(G) + \beta(G) \leq n$.

Therefore $\alpha(G) + \beta(G) = n$. $\square$

Gallai's identity #2

Theorem. For all n-vertex graphs G with no isolated vertices, $\alpha'(G) + \beta'(G) = n$.

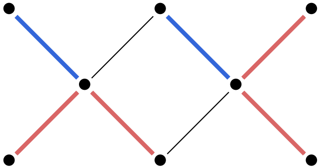
Pre-proof thoughts. The correspondence between matchings (for $\alpha'(G)$) and edge covers (for $\beta'(G)$) is not as nice. But we can still hope for a proof in the same way if we:

1. Prove that $\alpha'(G) \geq n - \beta'(G)$: given an edge cover of size $\beta'(G)$, find a matching of size $n - \beta'(G)$.
2. Prove that $\beta'(G) \leq n - \alpha'(G)$: given a matching of size $\alpha'(G)$, find an edge cover of size $n - \alpha'(G)$.

A useful fact: we can always cover k vertices with at most k edges.

From a matching to an edge cover

Suppose we have a matching with $k = \alpha'(G)$ edges. How can we find a good (small) edge cover?



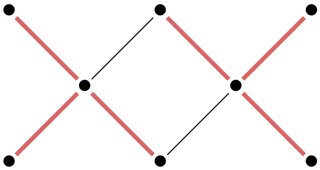
The **edges in the matching** are very efficient at covering: the k edges cover $2k$ vertices, which is the most we could hope for.

To cover the remaining $n - 2k$ vertices, we need at most one edge per vertex. That's $n - 2k$ **more edges**, for a total of $k + (n - 2k) = n - k$.

That's an edge cover of size $n - k$, so $\beta'(G) \le n - k = n - \alpha'(G)$.

From an edge cover to a matching

From an edge cover C of size $\ell = \beta'(G)$, how can we find a matching?



We can just greedily pick a matching of size $n - \ell$ from C.

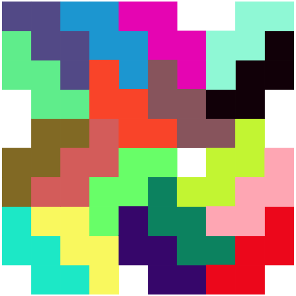
Number the edges of C as $e_1, \dots, e_\ell$. Each edge e_i covers either 1 or 2 more vertices not covered by $e_1, \dots, e_{i-1}$. If it covers 2 new vertices, it's safe to add to a matching.

To cover n vertices by ℓ edges, at least $n - \ell$ edges have to cover 2 new vertices. We get a matching of size $n - \ell$. So $\alpha'(G) \geq n - \beta'(G)$.

A tiling problem

Q1. You have an infinite supply of  shaped tiles. How many of them can you place on a 10×10 grid without overlap?

A1. We can fit at most 18 tiles:



Q2. How can we model this as a graph theory problem?

Tilings and independent sets

Define a graph G whose 256 vertices are ways to place a **single**  tile on the 10×10 grid: 8^2 places to center a tile, and 4 ways to rotate.

Two vertices are adjacent if those placements are incompatible: the tiles would overlap.

Then a tiling of the 10×10 grid is an independent set in this graph: a set of tile placements, no two of which overlap.

Disclaimer: Mathematica found an independent set of size 18 for me in seconds. It ran for several hours without confirming that no independent set of size 19 exists. (Probably I could have improved things by doing some clever symmetry breaking, IDK.)

About independent set problems

Some observations:

- The tiling problem seems like a toy example. But very minor variants of it are serious questions asked in the theory of error-correcting codes, where each tile represents a codeword.
- The problem "choose a maximum set of these things without any conflicts" has many other applications.
- With all that in mind, it's a real shame that—as you can guess from the disclaimer on the previous slide—the problem of finding a maximum independent set is very hard.

  Like the Hamiltonian cycle problem, we don't know of an efficient way to solve it for arbitrary graphs.

Bounds on independent sets

Theorem. Every n-vertex graph G with maximum degree $\Delta(G)$ has $\alpha(G) \geq \frac{n}{\Delta(G)+1}$.

Proof. This is the result we get with a greedy algorithm. Go through the vertices in an arbitrary order. Every time you get to a vertex v:

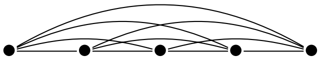
1. Add v to the independent set.
2. Delete v and all neighbors of v to prevent them from being considered.

With each step, we add 1 vertex to the independent set and remove at most $\Delta(G) + 1$ vertices from G. So we'll add $\frac{n}{\Delta(G)+1}$ vertices before we run out of vertices in G.

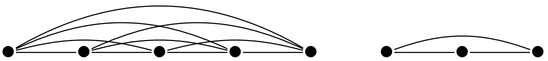
Theorem (Turán). $\alpha(G) \geq \frac{n}{d+1}$, where d is the average degree of G.

Why might a graph not have a perfect matching?

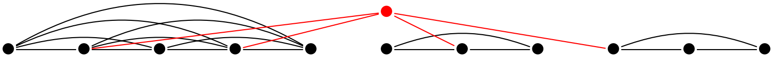
The stupidest reason for a graph to not have a perfect matching is an odd number of vertices. (You can't pair them all up!)



Even if the graph has an even number of vertices, this problem could be happening in each component:



For a more subtle problem, consider the graph below!

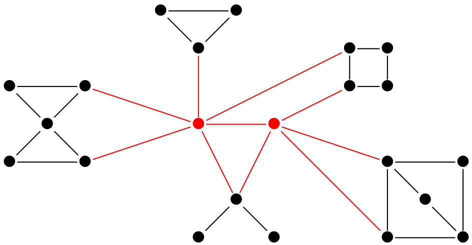


The red vertex can "rescue" any odd component—but not all three!

Tutte obstructions

Suppose that we have a graph with the following structure: a small set of vertices U such that $G - U$ has many odd connected components.

(Maybe also some even ones as well; those aren't a problem.)



Each odd component is a problem. Each vertex in U can fix one problem. If there are more than $|U|$ problems, they can't all be fixed.

Tutte's theorem

Theorem (Tutte). Suppose that G has no Tutte obstructions: for each $U \subseteq V(G)$, $G - U$ has at most $|U|$ odd components. Then G has a perfect matching.

We won't prove this theorem. So let's just make some observations about it.

- We have an idea of why this condition is necessary. The hard part is proving that it's sufficient.
- Taking $U = \varnothing$ is important: it checks whether G has any odd components (and whether G itself has an odd number of vertices).
- For bipartite graphs, Hall's theorem implies Tutte's. If there is a set S with $|N(S)| < |S|$, then take $U = N(S)$; every vertex in S is an odd component of $G - U$.

Applications of Tutte's theorem

Corollary (Petersen). Every 3-regular graph with no bridges has a perfect matching.

Proof. We check Tutte's condition for an arbitrary U. Let $G_1, \dots, G_k$ be the odd components of $G - U$.

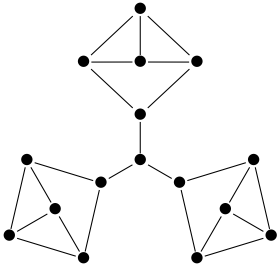
Each G_i must have at least 3 edges to U (for $\geq 3k$ such edges total).

(The number of such edges must be odd by summing the degrees in G_i, and we assumed that it cannot be 1.)

There are at most $3|U|$ edges from anywhere to U. So $3k \leq 3|U|$, or $k \leq |U|$. This proves that Tutte's condition holds. $\square$

What goes wrong with bridges

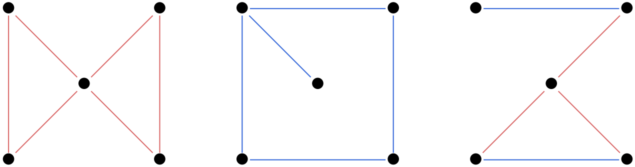
Here's why we needed the condition "with no bridges" on the previous slide:



This is a 3-regular graph, but it violates Tutte's condition: delete the middle vertex, and you get 3 odd components. No perfect matching!

Symmetric difference

If G and H are two graphs with the same vertex set V, their **symmetric difference** $G \triangle H$ is another graph with vertex set V. It contains all edges found in exactly one of G or H, but not both.



One way to think about $G \triangle H$ is this: which edges do we need to "toggle" (add or remove) to turn G into H (or H into G)?

Special case 1. If G and H share no edges, $G \triangle H = G \cup H$.

Special case 2. $G \triangle K_n$ is the complement $\overline{G}$.

Improving a matching

If we look for a matching by greedily adding on edges, we can get stuck in suboptimal matchings because we made a bad decision early on.

We'd like to correct this by thinking about more general steps to improve a matching that can fix past mistakes. To this end, we ask:

Question. What can the symmetric difference $M \triangle M'$ of two matchings M, M' look like?

(Here, we think of M and M' as subgraphs of the graph G.)

Partial answer. $M \triangle M'$ is a graph with maximum degree 2. Each vertex is incident to at most 1 edge of M, and at most 1 edge of M'.

In such a graph, all components are paths and cycles...

The symmetric difference of two matchings

Since $\Delta(M \triangle M') = 2$, there are only a few possibilities for each component of $M \triangle M'$.

1. Cycles with even length, alternating edges of M and M'.

2. Paths with even length, alternating edges of M and M'.

3. Paths with **odd** length, alternating edges of M and M'.

This is the only case which includes more edges from one matching than the other!

Augmenting paths

When M is a matching in a graph G, an M-**augmenting path** is a path $P = (v_0, v_1, \dots, v_{2k+1})$ in G such that

- P's endpoints v_0 and v_{2k+1} are uncovered by M.
- The edges $v_1v_2, v_3v_4, \dots, v_{2k-1}v_{2k}$ are all in M.

Treating both M and P as subgraphs of G, $M \triangle P$ is a bigger matching in G: it replaces the k edges $v_1v_2, \dots, v_{2k-1}v_{2k}$ by the $k+1$ edges $v_0v_1, v_2v_3, \dots, v_{2k}v_{2k+1}$.

If M is a matching and M' is a bigger matching, then $M \triangle M'$ will have at least $|M'| - |M|$ components which are M-augmenting paths.

So augmenting paths can always be used to make a matching bigger—if it's not already as big as possible.

Finding an augmenting path

Algorithms for finding matchings in graphs are all about efficiently searching for augmenting paths, over and over. This is:

- Relatively easy in bipartite graphs. If the bipartition is $A \cup B$ and we start at an uncovered vertex in A, we always alternate between $A \to B$ steps **outside** the matching, and $B \to A$ steps **in** the matching.

  (The Hopcroft–Karp algorithm is one way to make this happen.)

- Hard in general graphs. Here, it's possible to get to the same vertex in an even or an odd number of steps, and we don't know in advance which one will get us the augmenting path we want.

  (The blossom algorithm is a complicated way to solve this problem.)

Factorization

We begin with some definitions:

- A k-**factor** in a graph G is a k-regular spanning subgraph of G. ("Spanning" means that it includes all vertices of G.)
- In particular, a 1-factor is the same thing as a perfect matching.
- A 2-factor is a spanning subgraph whose components are cycles. (A Hamiltonian cycle is a connected 2-factor; but 2-factors are more general, and often easier to find.)
- Finally, a k-**factorization** of G is a decomposition of G into k-factors.

  ("Decomposition" means each edge of G is in exactly one of the k-factors.)

Some results

When do factorizations exist?

Theorem. Every regular bipartite graph has a 1-factorization.

Proof. We've already done the hard part: for $k \geq 1$, every k-regular bipartite graph has a perfect matching, or 1-factor. Take out that 1-factor, and you get a $(k-1)$-regular graph. Repeat until the graph is empty. □

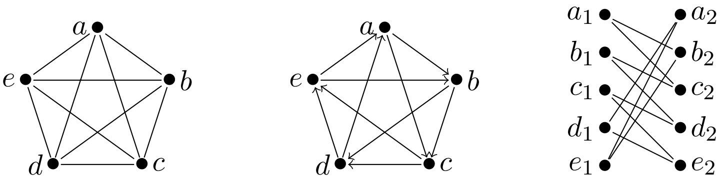
In general, regular graphs might fail to have matchings (we saw this last time). Even when they have matchings, they might not have 1-factorizations (this happens with the Petersen graph).

Theorem. Every $2k$-regular graph has a 2-factorization.

We'll work on proving this theorem today!

A transformation

We will transform our $2k$-regular graph into a directed graph and then into a bipartite graph.

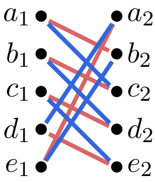


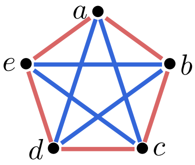
First, we create a directed graph where each vertex has indegree and outdegree k. To do this, take an Eulerian tour (of each component, if necessary) and orient each edge in the direction it's used.

Second, we turn this into a bipartite graph with two copies v_1, v_2 of every vertex v. Each directed edge (u, v) turns into an edge $u_1 v_2$.

Finding the factorization

We've gotten a k-regular bipartite graph, so it must have a 1-factorization (shown in **red** and **blue**).





Each edge u_1v_2 in the bipartite graph came from an edge (u, v) in the directed graph, which came from an edge uv in the original graph.

So each 1-factor in the bipartite graph gives us a set of edges in the original graph. That set of edges is a 2-factor, because each $v \in V$ is incident to one edge as v_1 and to a second edge as v_2. $\square$

Round-robin tournaments

How do we schedule a round-robin (everyone plays everyone else exactly once) tournament?

To finish all $\binom{n}{2}$ games as quickly as possible, we want to schedule multiple games at the same time. No person can play more than one game in the same round, of course.

In the ideal case (only possible when n is even), each round consists of exactly $\frac{n}{2}$ games, and everyone is playing.

This is a perfect matching in the complete graph K_n!

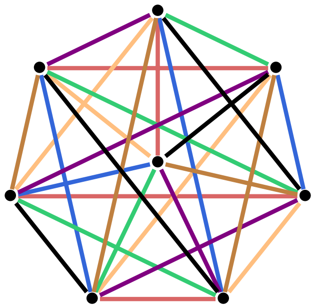
Each person has $n-1$ games to play, so the best possible schedule would consist of $n-1$ rounds with $\frac{n}{2}$ games in each round.

This is a 1-factorization of K_n!

The round-robin solution

Theorem. For all even n, K_n has a 1-factorization.

Proof by picture.



For odd n, leave out the center vertex; you'll have n rounds with $\frac{n-1}{2}$ games each, which is best possible.

Hamiltonian factorization

What if we want a Hamiltonian factorization of K_n: a 2-factorization in which every 2-factor is a Hamiltonian cycle? For which n is this possible?

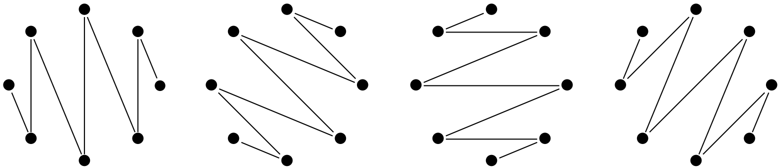
In order for any graph to have a 2-factorization, it must be $(2k)$-regular for some k. For a complete graph, this means $n = 2k + 1$: n must be odd.

Theorem. K_{2k+1} has a Hamiltonian factorization for all $k \geq 0$. Meanwhile, K_{2k} has a decomposition (not quite "factorization") into Hamiltonian **paths**.

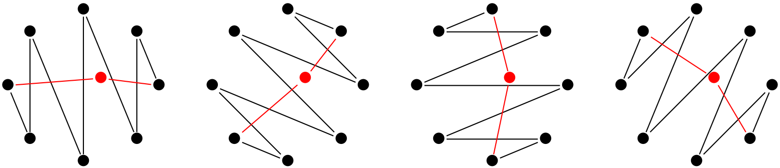
On the next slide, we'll show a construction for both statements.

Hamiltonian paths in K_{2k}

Here is another proof by picture:



And for the Hamiltonian factorization of K_{2k+1}:

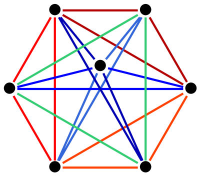


Triangle decomposition

In general, a **decomposition** of a graph G is a collection of subgraphs $G_1, \dots, G_k$ such that no two of them share any edges, and $G = G_1 \cup \dots \cup G_k$.

(This is a generalization of factorizations: a k-factorization is a decomposition into k-factors of G.)

We will consider triangle decompositions, or K_3 decompositions, in which every subgraph is isomorphic to K_3. For example:



Conditions for a decomposition

When does a K_3 decomposition exist?

- A K_3 decomposition is in particular a cycle decomposition; it only exists when all vertex degrees are even. We conclude that K_n only has one when n is odd.
- For another condition, let's count the triangles. There are $\binom{n}{2} = \frac{n(n-1)}{2}$ edges; if each triangle has 3 of them, that's $\frac{n(n-1)}{6}$ triangles.

  For this to be an integer, either n or $n-1$ must be divisible by 3.

Putting these together tells us that $n \equiv 1$ or $3 \pmod 6$.

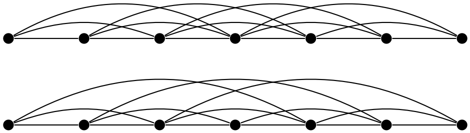
Theorem. This is sufficient: if $n \equiv 1$ or $3 \pmod 6$, then K_n has a triangle decomposition. *(We will not prove this.)*

Drawing graphs

Remember when I said that it doesn't matter how we represent a graph by a diagram? Well, now we're going to talk about how to represent a graph by a diagram.

Specifically: given a graph, we ask: can we draw this graph so that no edges cross?

Here's two examples:



One of these can be drawn without edge crossings, the other one can't. Which is which? As you can see, this is not an easy question!

Plane embeddings

We say that a **plane embedding** of a graph G is a way to draw G in the plane, representing vertices by points and edges by lines or curves, such that no edges cross.

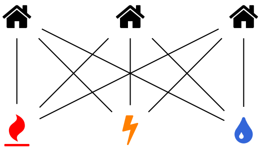
(Your textbook calls this a “plane graph”, which I think is misleading.)

A **planar graph** is a graph which has a plane embedding. Note:

- Being planar is a graph property: an isomorphism invariant of graphs. If $G \cong H$ and we have a plane embedding of G, then we can relabel it to get a plane embedding of H.
- Being drawn in the plane without crossings is **not** an isomorphism invariant.
- A graph can have many different plane embeddings.

The three utilities problem

The three utilities problem is a classic graph theory puzzle. We have three houses that need to be connected to the water, gas, and electricity companies.



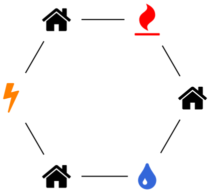
Can this be done without lines crossing or using a third dimension?

Formally: is the complete bipartite graph $K_{3,3}$ a planar graph?

We will see that this is impossible. $K_{3,3}$ will be our first example of a graph that we can prove does not have a plane embedding!

Why there is no solution

Begin by drawing a hexagon of six of the connections:



This hexagon **must** appear as a closed loop in any plane embedding, though it could be distorted or concave.

There are 3 edges remaining: one between each vertex and the opposite vertex. If 2 of them are inside the hexagon, they will cross. If 2 of them are outside the hexagon, they will cross. So there is no solution. $\square$

Turán's brick factory problem

Suppose we have $m > 3$ houses and $n > 3$ utilities. We've made the problem even harder, so of course there's no plane embedding. Can we say how many crossings there are?

This is still an open problem! A solution is known with

$$\left\lfloor \frac{n}{2} \right\rfloor \cdot \left\lfloor \frac{n-1}{2} \right\rfloor \cdot \left\lfloor \frac{m}{2} \right\rfloor \cdot \left\lfloor \frac{m-1}{2} \right\rfloor$$

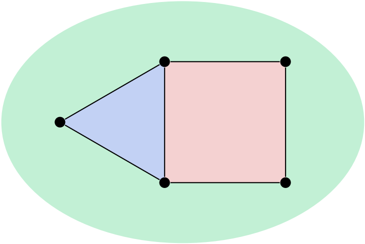
crossings. (Of the $\binom{mn}{2}$ pairs of edges, approximately $\frac{1}{8}\binom{mn}{2}$ cross.)

Already when $m = n = 9$, we don't know if this can be beaten.

Faces of a graph

When we give a plane embedding of a graph, its edges divide the plane into several regions, called the **faces** of a plane embedding.

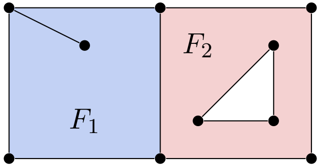
(One face is always the external face. Don't forget about it!)



In "nice and friendly" embeddings, the boundary of a face is a cycle in the graph. The **length** or **degree** of the face is the length of that cycle.

Misbehaving faces

Things don't always work out so nicely:



The boundary of face F_1 is not a cycle! In such cases, we think of the boundary as a closed walk; the length of the face is the length of the closed walk. Here, F_1 has length $\deg(F_1) = 6$.

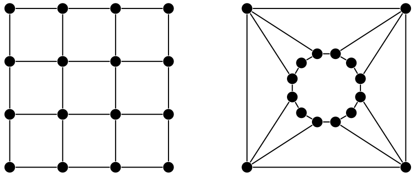
The boundary of face F_2 is not even connected! In this case, F_2 has several closed-walk boundaries; we say $\deg(F_2) = 7$, the sum of their lengths.

Matching up faces

For "well-behaved" planar graphs, when we draw multiple embeddings, there is a correspondence matching up equal-length faces in each one.

(Here, "well-behaved" is "3-connected", but don't worry about that.)

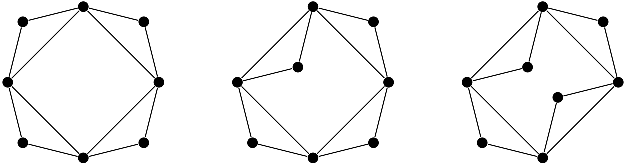
Sometimes, it's not an obvious correspondence:



Here, the external face of the first embedding has been turned into an internal face of the second embedding. (We can always take any embedding and "turn it inside out" to make any face the external face!)

Misbehaving faces II

Not all planar graphs have consistently behaving faces! Here are three embeddings of the same graph:

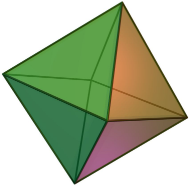


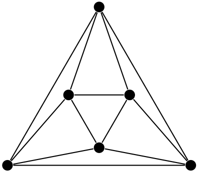
The faces have lengths $3, 3, 3, 3, 4, 8$ in the first embedding; $3, 3, 3, 3, 5, 7$ in the second; $3, 3, 3, 3, 6, 6$ in the third.

Moral: faces and face lengths are not isomorphism invariants! If you say "the faces of a graph", you are not being careful.

Planar graphs and 3D objects

3D shapes like the cube and octahedron have "skeleton graphs" which are planar graphs:





For intuition, you can imagine inflating a shape like the octahedron until you get a graph on the surface of a sphere. Then, poke a hole in the sphere and stretch it out into a flat shape.

This means that many things we'll learn about planar graphs are also helpful for understanding polyhedra.

Face lengths

Last time, we saw that the faces and face lengths of a planar graph depend on the embedding chosen. Is there anything that we can say about the faces that does not depend on the embedding?

Claim. Suppose that a graph G has a plane embedding with faces $F_1, F_2, \dots, F_k$. Then

$$\deg(F_1) + \deg(F_2) + \cdots + \deg(F_k) = 2|E(G)|.$$

Proof. Every edge in the graph either contributes $+1$ to the length of two faces, or $+2$ to the length of one face. $\square$

In particular, the sum of the lengths of the faces does not depend on the embedding of G!

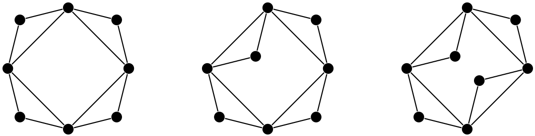
Euler's formula

Theorem (Euler's formula). Let G be a connected planar graph with n vertices and m edges, and suppose a plane embedding of G has k faces. Then

$$n - m + k = 2.$$

We'll use this formula for many things, but one thing it tells us is that the **number of faces** also does not depend on the embedding!

Here are embeddings of a graph with $n = 8$ vertices and $m = 12$ edges.

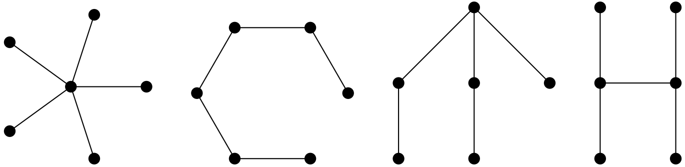


They all have $m - n + 2 = 6$ faces whose lengths sum to $2m = 24$.

Planar graphs and trees

Euler's formula starts from a familiar formula: if a tree has n vertices and m edges, then $m = n - 1$.

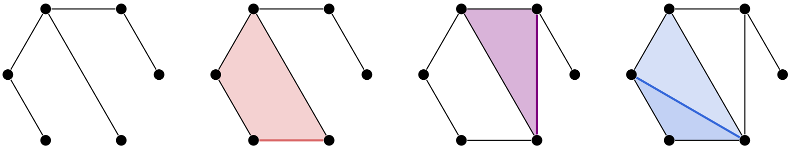
All trees are planar graphs! We could give a formal proof, but pretty much any drawing of a tree is a plane embedding unless you deliberately try to mess it up.



Trees have only one face (of length $2m$), so we confirm that $n - m + k = n - (n - 1) + 1 = 2$.

Induction on edges

Any connected graph G can be built by starting with a spanning tree and adding edges. So we'll do that in a plane embedding of G, and check that $n - m + k = 2$ continues to hold.



$n = 6$ $m = 5$
$k = 1$

$n = 6$ $m = 6$
$k = 2$

$n = 6$ $m = 7$
$k = 3$

$n = 6$ $m = 8$
$k = 4$

At each step, m and k both increase by 1, so $n - m + k$ does not change. $\square$

Edges in a planar graph

What is the maximum number of edges in an n-vertex planar graph?

Theorem. For $n \geq 3$, every n-vertex planar graph G has at most $3n - 6$ edges.

Proof. We can only have a face of length 2 or less if $G = K_1$ or $G = K_2$. Since $n \geq 3$, all our faces have length at least 3.

If there are m edges and k faces $F_1, \dots, F_k$, then

$$2m = \deg(F_1) + \deg(F_2) + \cdots + \deg(F_k) \geq 3 + 3 + \cdots + 3 = 3k.$$

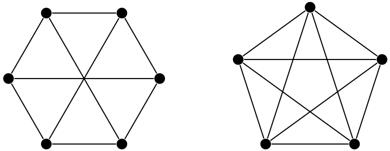
If G is connected, then $n - m + k = 2$, so $k = m - n + 2 \geq \frac{2}{3}m$. Solving for m, we get $m \leq 3n - 6$.

If G is not connected, we can add more edges to connect it; the same bound applies. $\square$

Another non-planar graph

So far, we've only seen one graph that we can prove is not planar: $K_{3,3}$.

But now, we can prove that lots of graphs are not planar. For example, K_5 is not planar, because it has $n = 5$ vertices and $10 > 3 \cdot 5 - 6$ edges.



However, $m \leq 3n - 6$ is only a necessary condition, not a sufficient one! (For example, $K_{3,3}$, which we know is not planar, has $m = 9$ edges and $n = 6$ vertices. $9 \leq 3 \cdot 6 - 6 = 12$.)

Triangulations

We know that $m \leq 3n - 6$ in any planar graph with n vertices and m edges. When is $m = 3n - 6$?

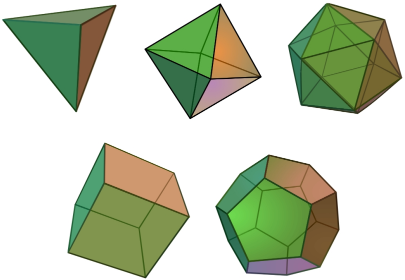
The inequality in our proof comes from the inequality $2m \geq 3k$, which came from knowing that all faces have length at least 3. So if the inequality becomes an equation, we need $2m = 3k$, which means we need all faces to have length exactly 3.

Such a planar graph is called a **maximal planar graph** or **triangulation**.

These are not hard to find! In any plane embedding, if a face has length more than 3, we can add an edge inside that face. So any planar graph can be extended to a maximal planar graph.

Platonic solids

The Platonic solids are the polyhedra whose faces are regular polygons, with the same number meeting at every corner.



Convex polyhedra are equivalent to planar graphs. Here, in particular, we have **regular** planar graphs in which every face has the same **length**.

Classifying the Platonic solids

Suppose that we have a plane embedding of a p-regular graph in which every face has length q. We must have $p > 2$ and $q > 2$.

Then we know that $m = \frac{1}{2}pn$ by the degree sum formula, and also that $m = \frac{1}{2}qk$ by the same formula for faces. Solve for everything in terms of n: $m = \frac{1}{2}pn$ and $k = \frac{2m}{q} = \frac{pn}{q}$. Finally, we have

$$n - m + k = 2 \implies n - \frac{1}{2}pn + \frac{pn}{q} = 2 \implies 1 - \frac{p}{2} + \frac{p}{q} = \frac{2}{n} > 0.$$

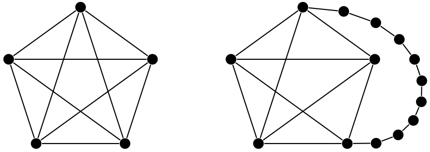
A more symmetric version of this is $\frac{1}{p} + \frac{1}{q} > \frac{1}{2}$.

This only has a limited number of solutions: (p, q) can be $(3, 3)$, $(3, 4)$, $(4, 3)$, $(3, 5)$, or $(5, 3)$.

More non-planar graphs

We know that graphs like K_5 with $m > 3n - 6$ can't be planar, because they are too dense: too many edges per vertex. But actually, there's a way to turn them into non-planar graphs with a relatively low number of edges.

Replace an edge vw by a path $(v, x_1, x_2, \ldots, x_\ell, w)$:



This adds ℓ more vertices and ℓ more edges. But it still can't be planar: any plane embedding of the new graph would give us a plane embedding of the old graph!

Subdivisions

Let G be a graph. A **subdivision of G** is a new graph H obtained from G by replacing as many edges we like by paths.

Equivalently, let's say that the operation "subdivide an edge vw of G" means "add a new vertex x, and replace edge vw by edges vx and xw". Then a subdivision of G is any graph we can get by doing this operation any number of times, to any edges we like.

If G is planar, then every subdivision of G is planar. If G is not planar, then none of its subdivisions are planar.

That's because going from a plane embedding of G to a plane embedding of a subdivision just means "draw some more dots". Going the other way is just erasing some dots.

Kuratowski's theorem

Observation. If G has a subgraph which is a subdivision of K_5 or a subdivision of $K_{3,3}$, then G is not planar.

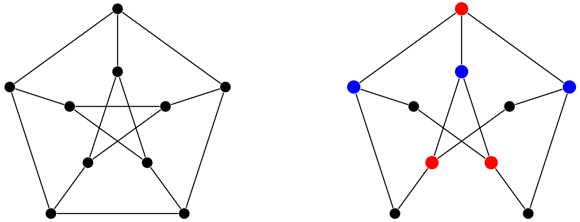
Theorem (Kuratowski). If G is not planar, then G has a subgraph which is a subdivision of K_5 or a subdivision of $K_{3,3}$.

In other words, this condition is both necessary and sufficient!

- To prove that a graph G is planar, the easiest way is to find a plane embedding of G.
- To prove that a graph G is not planar, we can use the observation: try to find a subdivision of $K_{3,3}$ or K_5 inside G.
- Kuratowski's theorem guarantees that this will always work: if G is not planar, then a proof of this type is possible.

Example: the Petersen graph

Let's use this test to show that the Petersen graph is not planar. Keep in mind that any subdivision of K_5 has degree-4 vertices. So in the Petersen graph, we should be looking for a subdivision of $K_{3,3}$.



On the right is one possible subdivision.

Another way to put it: we are looking for vertices $v_1, v_2, v_3, w_1, w_2, w_3$ with internally-disjoint $v_i - w_j$ paths for all i, j.

Dual graphs

You may have noticed the similarity between the degree sum formula and the face length formula:

$$\sum_{i=1}^{n} \deg(v_i) = 2m \qquad \sum_{i=1}^{k} \deg(F_i) = 2m.$$

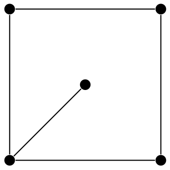
This is not a coincidence! The face length formula **is** the degree sum formula, but for a different graph. It's for the **dual graph** of G, whose vertices are the faces of G. Some warnings, though:

- In general, the dual graph depends on the plane embedding of G we chose.
- It's not always exactly a graph: it can have loops, and multiple edges between vertices.

The definition

Given a plane embedding of G, the dual graph G^* has vertices corresponding to faces of G. For each edge $e \in E(G)$, if the faces on either side of e in the embedding are F_i and F_j, G^* has an edge F_iF_j.

An example:

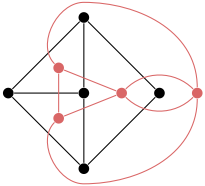


This graph G has an external face F_1 and an internal face F_2.

Four edges border F_1 and F_2, so G^* will have four copies of edge F_1F_2. One edge has F_2 on both sides, so G^* will have a loop at F_2.

Plane embeddings of dual graphs

If we're careful, then from the plane embedding of G, we get a plane embedding of G^*. Here is a nice example:

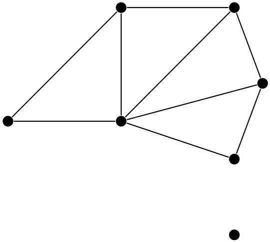


When G is connected, the vertices of G will correspond to faces of G^* (and in these plane embeddings of G and G^*, each vertex of G will be inside a face of G^*).

In that case, the dual of G^* will be G again.

Maps and planar graphs

A “dual-like” operation can be used to represent a map (in the ordinary, real-world sense) into a planar graph. We have a vertex for each region of the map, and an edge between adjacent regions.

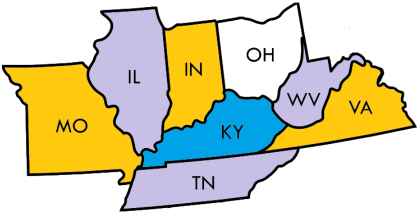


In the real world, maps sometimes “cheat”. When two disconnected regions are declared to be the same region, the result might not be a planar graph.

Coloring maps

How many colors do we need to color the regions on a map, so that adjacent regions have different colors?

Some maps require four colors. For example:



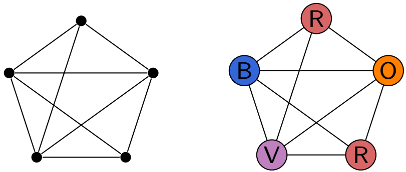
Whatever color Kentucky has, its 7 neighboring states can't be colored using only 2 more colors; they form an odd cycle.

Are four colors always enough? That's a much harder problem!

Vertex coloring

Generalizing this problem, a (vertex) coloring of G assigns every vertex of G a color; formally, it is a function $f : V(G) \to R$, where R is a set of colors.

A **proper (vertex) coloring** of G is a coloring that gives adjacent vertices different colors: $vw \in E(G) \implies f(v) \neq f(w)$.



We say that G is k-**colorable** if it has a proper coloring with k colors. The **chromatic number** $\chi(G)$ is the least k such that G is k-colorable.

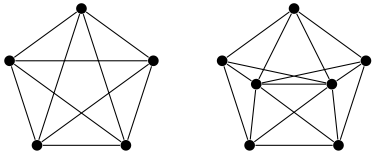
The four-color theorem

The general vertex coloring problem has many applications: register allocation in compilers, scheduling problems, etc.

Map coloring is equivalent to coloring a planar graph.

Theorem (Appel–Haken, 1976). Every planar graph is 4-colorable.

There is some intuition for this: the smallest graph that's not 4-colorable is K_5, and K_5 is not planar.

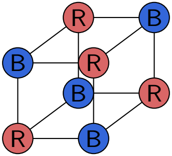


Not so fast! Many graphs require 5 colors without containing K_5...

Bipartite and 2-colorable graphs

Bipartite graphs are exactly the 2-colorable graphs. Two ways to say the same thing:

- "$V(G)$ can be partitioned into $A \cup B$ such that all edges have one endpoint in A and one endpoint in B"
- "The vertices of G can be colored red and blue such that all edges have one red endpoint and one blue endpoint."



Difficulty

We can check if a graph is bipartite (2-colorable) by trying to color it:

1. Color an arbitrary vertex red (because the colors are equivalent).
2. Color all its neighbors blue (because they cannot be red).
3. Color all of the blue vertices' neighbors red, and so on.

Eventually, we will have successfully colored everything, or this strategy will tell us to color a vertex blue when it has already been colored red (or red when it has already been colored blue).

If that happens, there is no 2-coloring, and the graph has an odd cycle.

Checking if a graph is k-colorable for $k \geq 3$ is very hard! There is no efficient algorithm known.

Bounds and coloring algorithms

If it's hard to determine the chromatic number of a graph, what can we do?

We can try to prove upper and lower bounds on the chromatic number of a graph.

Today, we will focus on upper bounds on the chromatic number. The best kind of upper bound is a constructive upper bound. Constructive upper bounds will give us an algorithm to color the graph with **some** number of colors, even though it may not be the best number.

For example, for any n-vertex graph G, its chromatic number $\chi(G)$ satisfies $\chi(G) \leq n$. This corresponds to the very simple algorithm "give each vertex its own color".

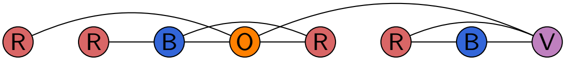
The greedy algorithm

The greedy algorithm is a slightly more intelligent coloring strategy.

Here, we assume that the vertices of a graph are ordered $v_1, v_2, \ldots, v_n$. The greedy algorithm colors v_1, then v_2, then v_3, and so on. It follows one simple rule:

- When coloring vertex v_i, use any available color[1] that was not used on any of v_i's neighbors that have been colored so far.

Here is an example. The colors we'll try are R B O V in order.



[1]For concreteness, "the first" color?

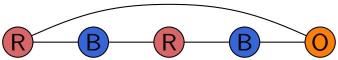
An improved bound

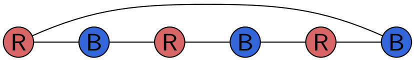
Theorem. Every graph G with maximum degree $\Delta(G)$ has chromatic number $\chi(G) \leq \Delta(G) + 1$.

Proof. This is how many colors the greedy algorithm uses, in the worst case.

If we have $\Delta(G) + 1$ colors available, then the greedy algorithm will never run out of colors. Each vertex we color has at most $\Delta(G)$ neighbors that have already been colored; even if they all get different colors, there is one color remaining. $\square$

This theorem says "$\chi(G) \leq 3$" when G is an odd cycle: that's good! But it also says "$\chi(G) \leq 3$" when G is an even cycle: less good.



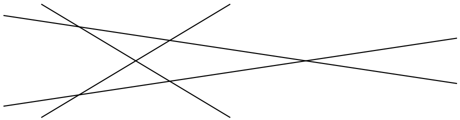


Greedy but smart

One way we can hope to improve this bound is to order the vertices carefully before we use the greedy algorithm, so that we don't see a high-degree vertex when all of its neighbors have been colored.

Here's an example, just to illustrate the strategy.

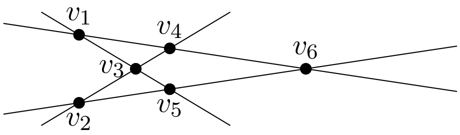
Theorem. Draw any number of lines in the plane, no three intersecting at a point.



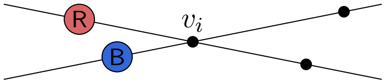
The intersection points between the lines can be colored with 3 colors, so that no line has consecutive intersection points of the same color.

Proving the theorem about lines

To prove the theorem about lines, order the points from left to right in the diagram, then use the greedy algorithm.



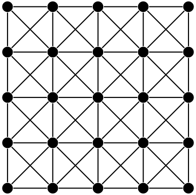
When we are coloring a vertex v_i, at most 2 of its neighbors will already have a color:



If 3 colors are available, one of them will always be free for v_i. $\square$

A motivational example

What is the chromatic number of the graph below?



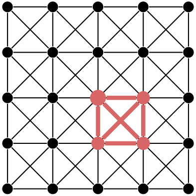
- The bound $\chi(G) \leq \Delta(G) + 1$ says at most 9 colors are needed.
- If we greedily color "in reading order", each vertex has at most 4 neighbors that precede it, so at most 5 colors are needed.
- But if you try this, you'll actually only use 4 colors.

Can you think of a lower bound that says $\chi(G) \geq 4$?

Lower bounds via cliques

In general: if H is a subgraph of G, then $\chi(G) \geq \chi(H)$. (To color G, you must color H.)

This is particularly useful when H is a complete graph or **clique**:



We've found a copy of K_4 inside the graph G. We know $\chi(K_4) = 4$, since we need to use a different color on each vertex. Therefore $\chi(G) \geq 4$ as well.

Clique number

In general, $\omega(G)$ is the number of vertices in the largest clique (complete subgraph) of G: the "clique number" of G.

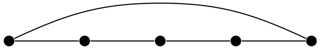
Some properties of $\omega(G)$:

- If G has n vertices, then $\omega(G) \le n$, with equality only if $G = K_n$.
- The clique number $\omega(G)$ is in a sense the opposite of the independence number $\alpha(G)$. Recall that $\alpha(G)$ is the size of the largest set of vertices with **no** edges between them.
- Actually, α and ω are related by $\alpha(G) = \omega(\overline{G})$ (and $\omega(G) = \alpha(\overline{G})$), because the complement of a clique is an independent set and vice versa.
- In general, we have a lower bound $\chi(G) \ge \omega(G)$.

How good is the clique bound?

In practice, for small examples, the clique number $\omega(G)$ gives a good lower bound on the chromatic number $\chi(G)$. It's especially good when $\omega(G)$ is relatively large.

It is not always tight. For example, any odd cycle C_{2k+1} has $\chi(C_{2k+1}) = 3$. But if $k \geq 2$, then $\omega(C_{2k+1}) = 2$.



Intuitively, a large clique is a "local property" of a graph: a small piece of the graph cannot be colored efficiently. But sometimes the chromatic number is large because of a "global property": any small piece can be colored efficiently, but they don't "fit together".

For large graphs, there is another flaw: $\omega(G)$ is hard to compute.

Triangle-free graphs

We know (from the example of an odd cycle such as C_5) that there are graphs for which $\chi(G) > \omega(G)$: the chromatic number is bigger than the clique number.

How much bigger can it get? Can we hope that $\chi(G)$ is always close to $\omega(G)$?

In fact, $\chi(G)$ and $\omega(G)$ can be very far apart. We'll see one way this can happen today, and another in the next lecture.

We say that a graph G is **triangle-free** if $\omega(G) \leq 2$: G contains no copies of K_3.

Theorem. There exist triangle-free graphs with arbitrarily large chromatic number.

The Mycielski construction

Let G be a graph with $V(G) = \{v_1, v_2, \ldots, v_n\}$. The **Mycielskian of** G is a graph $M(G)$ constructed from G as follows:

1. Start with a copy of G.
2. For each vertex $v_i \in V(G)$, add a "shadow vertex" u_i adjacent to all of v_i's neighbors in G.

   (There will never be edges between the shadow vertices.)
3. Finally, add a vertex w adjacent to all the shadow vertices $u_1, u_2, \ldots, u_n$.

If G is triangle-free, then so is $M(G)$. We will show that $\chi(M(G))$ is at least $\chi(G) + 1$.

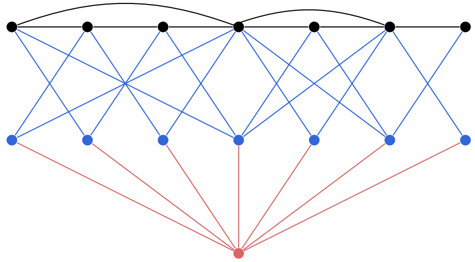
An example of a Mycielskian

Here is an example of how $M(G)$ is defined from a starting graph G. (This is not the G we'll want to look at in the long run.)

The original graph G:

The shadow vertices:

The final vertex w:

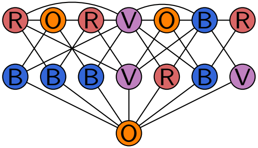


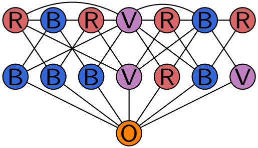
Even though $M(G)$ has some new triangles using a shadow vertex, those triangles are "shadows" of a triangle in G. If G were triangle-free, $M(G)$ would be as well.

The key step of the proof

Lemma. $M(G)$ has chromatic number at least $\chi(G) + 1$.

We'll actually show the contrapositive: however we color $M(G)$, we can color G with one fewer color. Take a proper coloring of $M(G)$...





...and modify it. For each vertex inside the copy of G, if it has the same color as w, change it to the color of its shadow vertex.

The result is a coloring of $M(G)$ in which no other vertex has w's color. In particular, the copy of G inside $M(G)$ uses one fewer color! $\square$

The induction

To finish our proof, define a sequence of graphs.

1. Let $G_2 = K_2$. This is triangle-free and has chromatic number 2.
2. Let $G_3 = M(G_2)$. This is triangle-free and has chromatic number 3. (In fact, G_3 is the 5-cycle C_5.)
3. Let $G_4 = M(G_3)$. This is triangle-free and has chromatic number 4. (G_4 is an 11-vertex graph known as the Grötzsch graph.)
4. And so on, letting $G_n = M(G_{n-1})$ for all n.

The result is that G_n is a triangle-free graph with $3 \cdot 2^{n-2} - 1$ vertices such that $\chi(G_n) = n$. $\square$

(We only proved that $\chi(G_n) \geq n$, but it's possible to properly color $M(G)$ with $\chi(G) + 1$ colors.)

Clique and independence number

Recall the definitions of the independence number $\alpha(G)$ and the clique number $\omega(G)$:

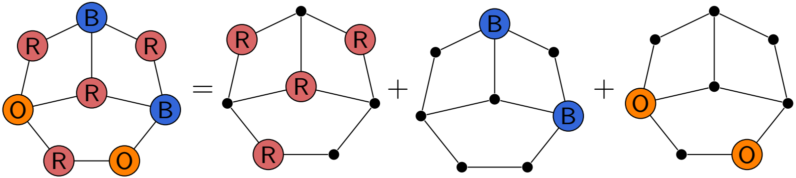
- $\alpha(G)$ is the maximum size of a set of vertices with **no** edges between them.
- $\omega(G)$ is the maximum size of a set of vertices with **all** edges between them.

We have a lower bound on chromatic number from cliques: $\chi(G) \geq \omega(G)$. That's because a k-vertex complete graph needs at least k colors.

Surprisingly, we also get a lower bound on chromatic number from independent sets!

Independence number and coloring

In any proper coloring of a graph, each color class is an independent set:



By definition, each independent set has at most $\alpha(G)$ vertices.

So if we are supposed to cover all n vertices of an n-vertex graph with sets of size at most $\alpha(G)$, we need at least $\frac{n}{\alpha(G)}$ sets: $\chi(G) \geq \frac{n}{\alpha(G)}$.

This bound is not always tight. In the graph above, $n = 8$ and $\alpha(G) = 4$. However, you can't cover G with two 4-vertex independent sets: all such sets need the center vertex.

Comparing the two bounds

Of the two bounds

$$\chi(G) \geq \omega(G) \qquad \chi(G) \geq \frac{n}{\alpha(G)}$$

which one is better? It's situational, of course, but we can prove that

Theorem. The second bound is much better for almost all graphs.

To prove this, we'll first need to be able to think about what "almost all graphs" means. We will think of it this way:

1. Pick a vertex set $\{v_1, v_2, \ldots, v_n\}$.
2. For each pair $\{v_i, v_j\}$, flip a coin to decide if edge $v_i v_j$ exists.

This gives us a **random** n**-vertex graph**. Every graph on these n vertices is equally likely to be the result.

Cliques in the random graph

How big are cliques in the random graph?

As an estimate, let's consider a graph with $n = 1\,000\,000$ vertices, and look for 40-vertex cliques inside it.

- There are $\binom{1\,000\,000}{40} \approx 1.22 \times 10^{192}$ possible 40-vertex sets that could be cliques.
- But each one is very unlikely to be a clique: all $\binom{40}{2} = 780$ edges exist only with probability $2^{-780} \approx 1.57 \times 10^{-235}$.
- Multiplying these together gives an approximately 1.93×10^{-43} upper bound on the probability that any 40-vertex clique appears.

So the lower bound $\chi(G) \geq \omega(G)$ can't even tell us that $\chi(G) \geq 40$ for this $1\,000\,000$-vertex graph.

Independent sets beat cliques!

How big can independent sets get in the same random graph?

Each pair of vertices is equally likely to have an edge or not. Therefore having a 40-vertex independent set is just as unlikely as having a 40-vertex clique. (Same number of possible 40-vertex sets; same probability of 2^{-780} for each of them to work.)

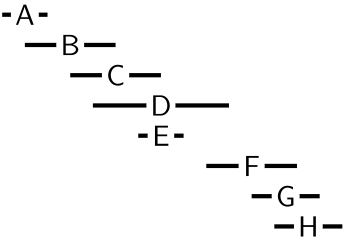
Therefore almost all graphs G on $1\,000\,000$ vertices have:

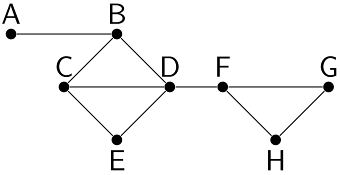
- no 40-vertex clique;
- no 40-vertex independent set;
- chromatic number $\chi(G) \geq \frac{n}{\alpha(G)} \geq \frac{1\,000\,000}{40} = 25\,000$.

(It is possible—but much harder—to prove that this lower bound is actually very close to the truth.)

Chromatic numbers and the scheduling problem

Suppose the math department schedules several events for the same day. They have different starting and ending times, shown below:





We can represent overlap between the events by a graph.

Only 3 rooms are free for these events. Can we 3-color the graph? That would let us assign the events to rooms without double-booking.

Interval graphs

An **interval graph** is a graph defined by a set of intervals $\{[a_1, b_1], \dots, [a_n, b_n]\}$.

It has a vertex v_i for each interval $[a_i, b_i]$, and an edge $v_i v_j$ whenever the intervals $[a_i, b_i]$ and $[a_j, b_j]$ overlap: $[a_i, b_i] \cap [a_j, b_j] \neq \varnothing$.

Coloring interval graphs has many applications, such as the scheduling problem on the previous slide, or the register allocation problem mentioned earlier.

Theorem. If G is an interval graph, then $\chi(G) = \omega(G)$.

In other words, if there are never more than k events happening at the same time, we can always schedule them in k rooms.

Greedy coloring an interval graph

Proof. To color an interval graph, order the intervals by their starting point; then, color them greedily in this order.

Suppose we're coloring vertex v_i, corresponding to interval $[a_i, b_i]$. Which **neighbors** of v_i have already been colored?

It's exactly the ones corresponding to intervals containing the point a_i. (Their starting point is before a_i, but they intersect $[a_i, b_i]$, so their endpoint must be after a_i.)

The intervals containing a_i (including $[a_i, b_i]$ itself) correspond to a clique in G.

So there must be at most $\omega(G)$ of them: v_i, and at most $\omega(G) - 1$ previously-colored neighbors. They will all be different colors—but we have $\omega(G)$ colors, so we have a color left to use on v_i. $\square$

A glimpse of chordal graphs

There's a larger class of graphs called **chordal graphs** with the same property.

Chordal graphs are graphs whose vertices can be ordered $v_1, v_2, \ldots, v_n$ such that the neighbors of v_i in $\{v_1, v_2, \ldots, v_{i-1}\}$ form a clique.

(This is exactly what ordering the intervals by starting point does!)

As a result, if G is a chordal graph, and we can find such an ordering (which turns out to be easy to do), then we can greedily color G with only $\omega(G)$ colors—and get $\chi(G) = \omega(G)$.

We've seen several cases where $\omega(G)$ is much smaller than $\chi(G)$. But chordal graphs come up in many applications, and they're a reason why this bound is still often useful for graphs that come up in practice.

Coloring planar graphs

Map coloring is where the study of vertex coloring began: with coloring planar graphs.

As mentioned before, the four color theorem says that every planar graph has chromatic number at most 4.

However, proving this is hard. There have been several proofs, but they consider hundreds of cases, which are in practice checked by computer.

We will:

1. Prove some weaker bounds on the chromatic number of planar graphs.
2. Prove the four-color theorem for some special types of planar graphs.

What we know about planar graphs

We will need the following theorem, which we proved earlier:

Theorem. Every planar graph on $n \geq 3$ vertices has at most $3n - 6$ edges.

Together with the degree sum formula, we get the following inequality for such graphs: if $V(G) = \{v_1, v_2, \ldots, v_n\}$, then

$$\sum_{i=1}^{n} \deg(v_i) \leq 6n - 12 \implies \frac{1}{n} \sum_{i=1}^{n} \deg(v_i) \leq 6 - \frac{12}{n} < 6.$$

This says the average degree is less than 6, so we conclude:

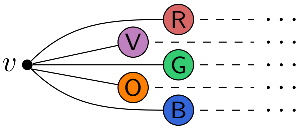
Corollary. Every planar graph has a vertex v with $\deg(v) < 6$.

The six color theorem

Theorem. Every planar graph G has $\chi(G) \leq 6$.

Proof. We will prove this theorem by induction on n, the number of vertices in G. Base case: if $n = 1$, then $\chi(G) = 1 \leq 6$.

Suppose all $(n-1)$-vertex planar graphs are 6-colorable; let G be an n-vertex graph. By the corollary, G has a vertex v with $\deg(v) < 6$; by the induction hypothesis, $G - v$ has a proper coloring with ≤ 6 colors.



Extend it to a proper coloring of G by picking a color for v. Since v has at most 5 neighbors, at least one color is available to color it. $\square$

Triangle-free planar graphs

Remember how general triangle-free graphs could have arbitrarily large chromatic number?

Theorem. Every triangle-free planar graph G has $\chi(G) \leq 4$.

Proof. For this, we have to go back and dig into where we got the bound $m \leq 3n - 6$ for general planar graphs.

We said: if there are m edges and k faces $F_1, \dots, F_k$, then

$$2m = \deg(F_1) + \deg(F_2) + \dots + \deg(F_k) \geq 3 + 3 + \dots + 3 = 3k.$$

If G is triangle-free, then $\deg(F_i) \geq 4$ for all i, and $2m \geq 4k$.

This leads to $m \leq 2n - 4$ for triangle-free planar graphs. In this case, we can improve our corollary to show that there is a vertex v such that $\deg(v) < 4$, and prove the new theorem. □

Greedy coloring heuristics

Suppose you want to find a pretty good coloring of a graph, without too much effort. What might you do?

One answer is to use the greedy algorithm, with some heuristic for choosing the order of vertices. A few common heuristics:

- **Random:** just randomly permute the vertices. This is reasonable if you want to avoid worst-case behavior.
- **Largest-First:** Color the vertices in decreasing order of degree. This deals with high-degree vertices early, before we color too many of their neighbors.
- **Smallest-Last:** What we did for planar graphs! Let v be the smallest degree vertex, color $G - v$ (recursively), then color v.

Some history

Since the four color problem was posed in 1852, lots of people attempted proofs. Peter Tait published one in 1880 (which turned out not to work).

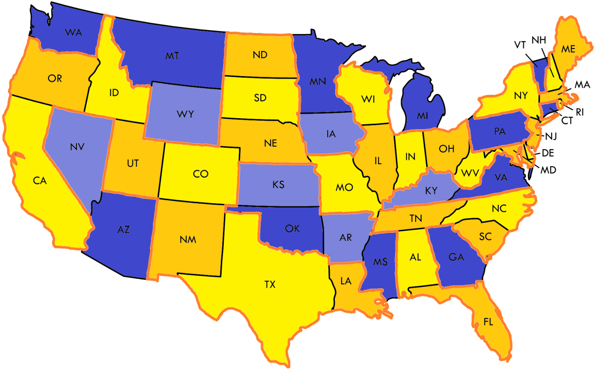
The idea of Tait's proof was to color a map by drawing a Hamiltonian cycle in the dual graph. (We will see on the next slide how this helps.)

Tait thought that for reasonably well-behaved maps, such a cycle would always exist.

It wasn't until 1891 that anyone even noticed this assumption. In 1946, a counterexample was found, ruling out this approach to the four-color theorem.

How Tait would color the United States

Given a Hamiltonian cycle in the dual graph (shown in orange)...



...we can 2-color the regions inside the cycle, and then 2-color the regions outside.

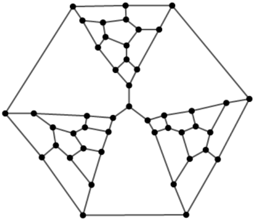
Tait's conjecture

Tait's proof could be rescued with the following conjecture:

Conjecture. Every 3-regular, 3-connected planar graph is Hamiltonian.

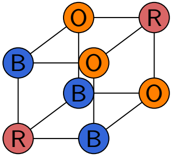
(The assumptions are not true for the dual graph of every map, but they're true in all the "hard" cases.)

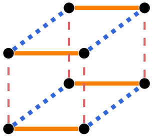
In 1946, the first counterexample was found (now known as the Tutte graph):



Proper edge colorings

Previously: a proper (vertex) coloring of a graph G is an assignment of "colors" to each vertex, so that adjacent vertices get different colors.



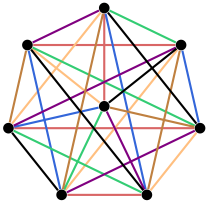


A proper **edge coloring** of a graph G is an assignment of colors to each edge, so that edges that share an endpoint get different colors.

The **edge chromatic number** $\chi'(G)$ is the least number of colors needed to properly edge color G.

Edge coloring and 1-factorizations

We've seen that some regular graphs have 1-factorizations:



A 1-factorization of a k-regular graph G is in particular an edge coloring of G with k colors.

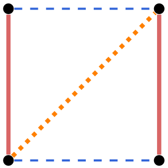
An edge coloring is more general. The edges of one color must be a matching (to avoid sharing endpoints) but they don't have to be a perfect matching.

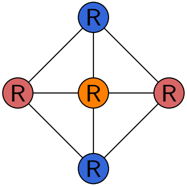
Edge coloring and line graphs

We've seen this with several other graph properties before. The edge chromatic number of G is the ordinary chromatic number of its line graph! In symbols:

$$\chi'(G) = \chi(L(G)).$$

Here is an edge coloring of G, and the corresponding coloring of $L(G)$:





Basic properties

The first thing we can say about the edge chromatic number is that it's at least the maximum degree.

Fact. All graphs G have $\chi'(G) \geq \Delta(G)$.

Proof. This is really the first lower bound we proved about ordinary vertex coloring, in disguise. It is $\chi(G) \geq \omega(G)$, applied to the line graph.

If you look all $\Delta(G)$ edges incident on the same maximum-degree vertex, they all have to be different colors. So at least $\Delta(G)$ colors are needed. $\square$

The obvious upper bound is not very good. Applying $\chi(G) \leq \Delta(G) + 1$ to the line graph gets us $\chi'(G) \leq 2\Delta(G) - 1$.

Vizing's theorem

It is hard to prove, but it turns out that much more is true!

Theorem (Vizing). All graphs G either have $\chi'(G) = \Delta(G)$ or $\chi'(G) = \Delta(G) + 1$.

So when we're edge-coloring graphs, we just have to figure out which of the two cases we're in.

- From our results on 1-factorizations: $\chi'(G) = \Delta(G)$ for every regular bipartite graph.
- More is true: **all** bipartite graphs have $\chi'(G) = \Delta(G)$, and we can prove this in much the same way.
- For complete graphs: $\chi'(K_n) = \Delta(K_n) = n - 1$ when n is even, and $\chi'(K_n) = \Delta(K_n) + 1 = n$ when n is odd.

Acquaintances and strangers

Fact. Suppose that there are 6 people at a party, and any two of them are either acquaintances or strangers. Then we can find either

- 3 people such that any two of them are acquainted, or
- 3 people such that any two of them are strangers.

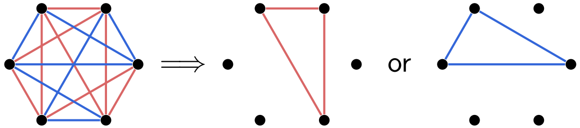
We could say: let $V(G)$ be the people, and put an edge between acquaintances. If $|V(G)| = 6$, then either $\omega(G) \geq 3$, or $\alpha(G) \geq 3$.

Since we're coloring edges today, let's say that we have a K_6 whose edges are colored red or blue. Then we can find either:

- 3 vertices such that all 3 edges between them are red, or
- 3 vertices such that all 3 edges between them are blue.

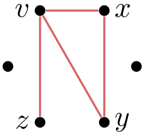
Monochromatic triangles

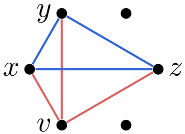
Fact. If we 2-color the edges of K_6, at least one of these happens:



Proof. Let v be any vertex of K_6. Out of the 5 edges out of v, at least 3 (call them vx, vy, vz) have the same color; say, red.

If one of xy, xz, yz is red (say, xy) then v, x, y induce a red triangle.





Otherwise, xy, xz, yz are all blue: x, y, z induce a blue triangle. □

One more step

Old fact. Any red-blue coloring of K_6 has a red K_3 or blue K_3.

New fact. Any red-blue coloring of K_{10} has a red K_3 or blue K_4.

Proof. As before, look at the 9 edges out of a vertex v. Either at least 4 are red, or at least 6 are blue (because $3 + 5 < 9$).

- **If edges vx_1, vx_2, vx_3, vx_4 are red:** as before, either some $x_i x_j$ is red, creating a red K_3, or all edges between x_1, x_2, x_3, x_4 are blue, creating a blue K_4.

- **If edges $vy_1, vy_2, \dots, vy_6$ are blue:** by our previous result, the edges just between $y_1, \dots, y_6$ form either a red K_3 or a blue K_3.

  If it is red, great. If y_i, y_j, y_k form a blue K_3, then v, y_i, y_j, y_k form a blue K_4. □

Ramsey's theorem

Theorem (Ramsey). For any $s, t \in \mathbb{N}$ we can pick a large enough n such that any red-blue coloring of K_n has a red K_s or blue K_t.

Proof sketch. We do the same thing as we did before, inducting on s and t. Pick a vertex v, and either there are many red edges out of v, or many blue edges.

In the first case, we find a red K_{s-1} or blue K_t among the red-edge neighbors of v. With v, the red K_{s-1} becomes a red K_s.

In the second case, we find a red K_s or blue K_{t-1} among the blue-edge neighbors of v. With v, the blue K_{t-1} becomes a blue K_t. $\square$

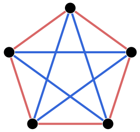
This generalizes to more than 2 colors, too! The same argument applies with one case per color.

How large is n?

Let $R(s,t)$ be the minimum value of n such that any red-blue coloring of K_n has a red K_s or blue K_t. How large is $R(s,t)$?

Our proof on the previous slide gives $R(s,t) \leq R(s-1,t) + R(s,t-1)$. This gives $R(s,t) < \binom{s+t-2}{s-1}$ by induction.

Lower bound: $R(3,3)$ is exactly 6. Proof by picture!



Lower bound: $R(40,40) > 1\,000\,000$. We saw this last week: if you randomly color $K_{1000000}$, you almost never get a red K_{40} or blue K_{40}.

In general, we know $2^{k/2} < R(k,k) < 2^{2k}$.