Math 442 (Graphs & Networks) Notes

January → April 2025

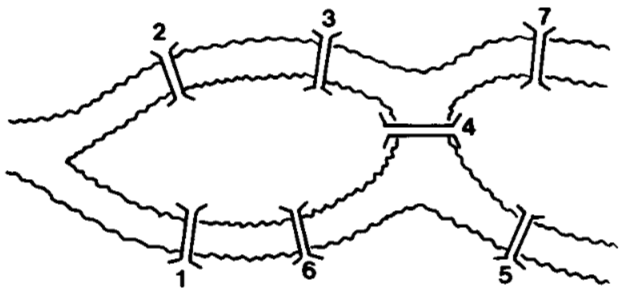
Contents

Lecture 1 (Feb. 25)

The content for today comes from Chapter 3, Circuits and Cycles.

We begin with the Königsberg bridge problem (now Kaliningrad), the oldest problem in graph theory.

Problem 1.1. *Can you walk across all 7 bridges without re-crossing any? Try it. You can start anywhere.*



It turns out this is impossible! Let's convert this problem to the language of graph theory. First, some definitions. In this chapter we are going to be dealing with:

- multigraphs: graphs with multiple edges allowed
- pseudographs: multigraphs with loops allowed

We have some common terms on multi/pseudographs.

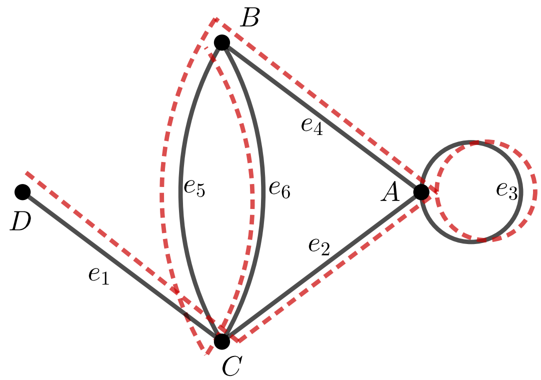
- Walk: A list of alternating vertices/edges $A_1e_1A_2e_2\ldots A_{n-1}e_{n-1}A_n$.
  - Ends on vertices
  - e_k connects A_k, A_{k+1}
  - $A_k \neq A_{k+1}$ unless e_k is a loop
  - Length of a walk is counted by the edges.
- Trail: a walk with no edge repeated.
- Path: a walk with no vertex repeated. Question: is this also a trial?

A closed trail is a 'circuit'. A closed path is a 'cycle'. Note that pseudographs can have cycles of lengths 1 or 2... these are called "loops" and "loons" respectively.

Eulerian circuits

An *Eulerian circuit (resp. trail)* in a pseudograph G is a circuit (resp trail) that contains every edge of G.

Example 1.2.



Eulerian trail $=$ $De_1Ce_2Ae_3Ae_4Be_5Ce_6B$

Theorem 1.3 (Euler, 1736)**.** *If a pseudograph G has an Eulerian circuit, then G is connected and the degree of every vertex is even.*

Proof. If the trail exists, we must pass through a given vertex A some number of times h, leaving no edge unpassed. The degree of A is then $2h$. □

The converse also true, proven over 100 years later.

Theorem 1.4 (Hierholzer)**.** *If a pseudograph G is connected and the degree of every vertex is even, then G has an Eulerian circuit.*

To prove this, we need a Lemma.

Lemma 1.5. *If every vertex in a pseudograph G has even degree, then any vertex with positive degree in G lies on a circuit in G.*

Proof. Let A be a vertex in G. Walk off of A through the graph. Every vertex we come to has even degree, so we can always leave by an edge we haven't used – each pass 'uses up' two edges incident to A. □

Now, we prove Theorem 1.4.

Proof. Let G be a connected pseudograph, degree of every vertex is even. Let C be a longest circuit in G. Assume for contradiction that G does not contain every edge of G.

Define a graph H, $v(H) = v(G)$, $e(H) = e(G) \backslash e(C)$. The degrees of vertices in H are even.

Claim: we can find a vertex A with non-0 degree in H. Assume we couldn't, and let A be any vertex in the circuit. There is at least some vertex B in H with positive degree. But this vertex is connected to A by a path in G, from connectivity. Since A is the cycle and B is not, somewhere along the path there is an edge connecting a vertex in C to a vertex outside of C.

By Lemma 1.5, A is part of some circuit C_1 in H. So we extend the circuit C by adding C_1. □

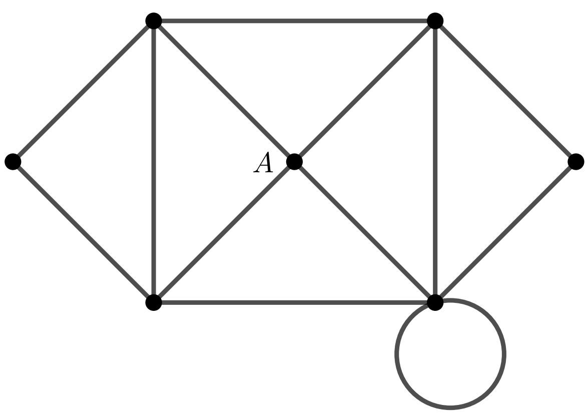
Algorithms, and Eulerian trails

The previous proof can easily be converted into an algorithm for finding an Eulerian circuit.

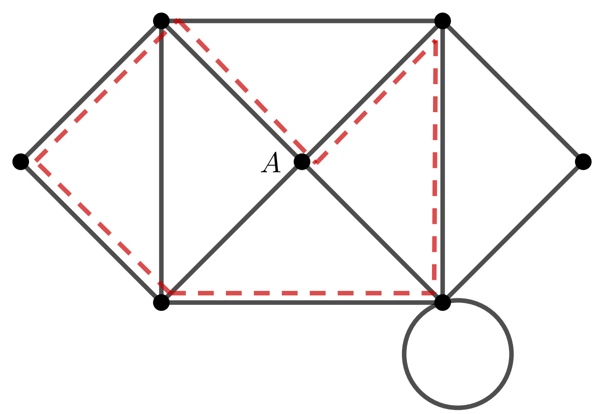
Hierholzer's steps:

1. Choose any starting vertex A.
2. Follow a trail until returning to A. Call this circuit C. (Why do we always get back to A?)
3. If there is a vertex A' on C with edges outside of C, define the graph H: $v(H) = v(G)$, $e(H) = e(G) \backslash e(C)$ and repeat (2) using this starting point and graph.
4. Append the new circuit to C, and repeat stage (3-4).

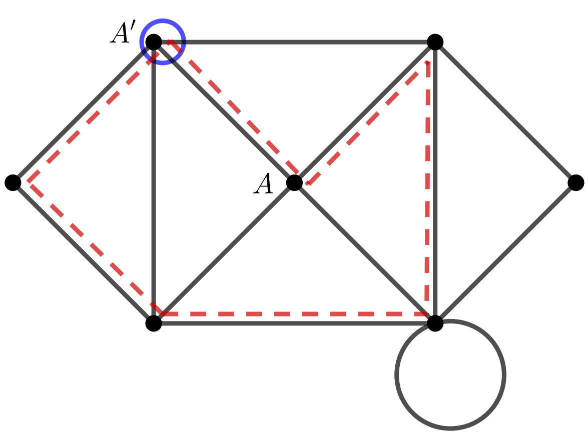
We can prove that this really terminates with an Eulerian circuit by following the logic of the proof. We now took a look at an example.



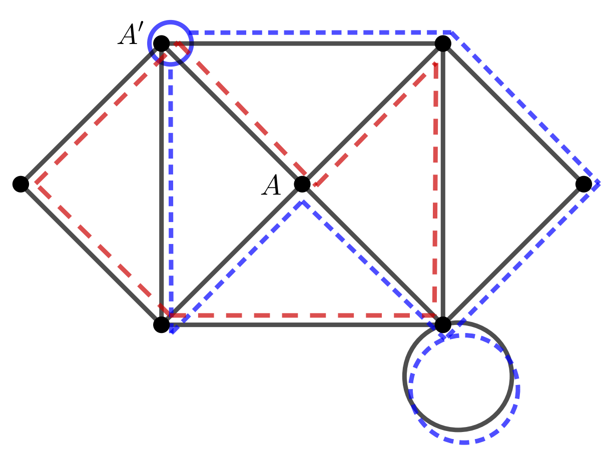
(1.) Choose any starting point



(2.) Follow a trail randomly until back at A



(3.) Choose a vert. A' with edges not in circuit



(4.) Find a new circuit out of remaining edges, append, and repeat 3-4 until all edges used.

Figure 1: Heirholzer's example

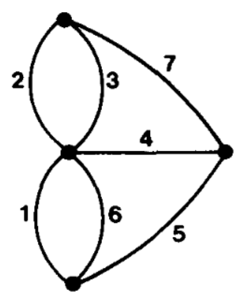
Lecture 2 (Feb. 27)

Theorem 2.1. *A pseudograph G has an Eulerian trail iff it is connected and has precisely 2 vertices of odd degree.*

Proof. $\implies$ G has an Eulerian trail, starting at A and ending at B. Adding an edge here creates an Eulerian circuit, so all vertices have even degree. So A and B must have had odd degree, and they are the only ones.

$\impliedby$ Let A and B have odd degree. Adding an edge here, everything has even degree, so there's a circuit. Removing that edge leaves a trail. $\square$

(Exercise) We can now solve the Königsberg bridge problem. To do so, write it in graph theory language: Does the following graph have an Eulerian trial?



Definition 2.2. *A* bridge *in a (pseudo)graph is an edge whose deletion disconnects the graph (i.e., increases the number of connected components).*

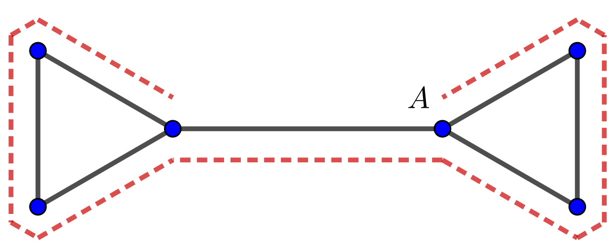
Another algorithm for finding Eulerian circuits and trails is called Fleury's algorithm. You can think of this as the 'don't burn bridges algorithm'. We run this algorithm on a pseudograph which is connected and has precisely 0 or 2 vertices of odd degree.

Fleury's Steps:

1. Choose a starting vertex A with odd degree (if one exists, else choose any).

2. Find an unused edge that is not a bridge, if there is one. Otherwise choose a bridge.

3. Move across to the other vertex, remove the edge (and add it to the trail), and repeat (2-3) until stuck.



Exercise: Where does this process terminate and why? Why does this use every edge? The first answer is the same as in the previous algorithm: its not possible to get stuck anywhere else because of the even degrees. For the second answer, if it doesn't use every edge, then we can argue at some point we must have used a bridge when there was another non-bridge edge available.

Question: what happens if you start in the wrong spot (an even vertex, when there are odd ones available)? What if there are more than 2 odd vertices?

Decomposition into trails

Theorem 2.3 (Listing)**.** *If G is a connected pseudograph with precisely $2h$ vertices of odd degree, $h \neq 0$, G is decomposable into h trails and not fewer.*

Proof. (Exercise) Add h edges to make all vertices even degree. Note that each vertex only gets 1 additional added edge. Find the Eulerian circuit, then delete the edges. The circuit was chopped in h places, leaving h (non-empty!) trails. Each odd vertex must be the endpoint of a trail, so no fewer than h. $\square$

Problem 2.4. (Exercise, stated but not solved in class) *If a 3-regular graph is decomposed into trails, the average length is ≤ 3.*

Problem 2.5. (Exercise, discussed but not solved class) *Write down the modified Hierholzer's algorithm so that it finds Eulerian trails, when there are exactly 2 vertices of odd degree.*

Lecture 3 (Mar. 4, Review class)

Today is a class of review of the things we have learned so far. **About the quiz:** There will not be difficult logic or proofs. Graphs are are all simple, no pseudo/multi-graphs. In short, the most important things to study for the quiz are

- the basics, definitions, algorithms
- properties of common graphs:
  - Complete graphs
  - Bipartite graphs
  - Cycles
  - Trees
  - Paths
  - Regular graphs, etc!

Degree sequences

Theorem 3.1 (The Erdős–Gallai Theorem)**.** *A sequence of non-negative integers* $d_1,\ d_2, \ldots, d_n$ *arranged in non-increasing order* $(d_1 \geq d_2 \geq \cdots \geq d_n)$ *is the degree sequence of a simple, undirected graph if and only if:*

1. *The sum of the degrees is even:*
$$\sum_{i=1}^{n} d_i \text{ is even}$$
2. *The following inequality holds for all* $k = 1, 2, \ldots, n$*:*
$$\sum_{i=1}^{k} d_i \leq k(k-1) + \sum_{i=k+1}^{n} \min(d_i, k).$$

Havel-Hakimi steps: Another way to see if a sequence $d_1,\ d_2, \ldots, d_n$ is graphic, is by running the following algorithm.

1. sort the degrees to be decreasing.
2. delete d_1, and subtract 1 from the next d_1 elements.
3. repeat as necessary

Example 3.2. *Is (6, 5, 5, 4, 3, 2, 2, 1) graphic? Is (6, 5, 5, 4, 2, 2, 1, 1) ? Use the algorithm to figure this out, and reconstruct the graph if possible.*

Prüfer codes

Check the other notes for the algorithm. In class, we went through a couple examples. Draw the tree whose Prüfer code is $(1, 1, 1, 1, 6, 5)$.

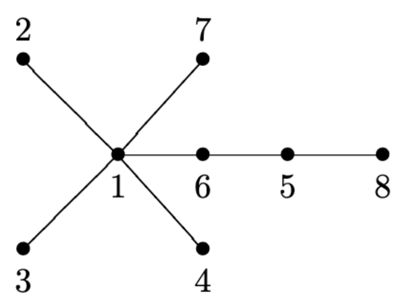


Figure 2: It should be this

Find the Prüfer code for the following graph.

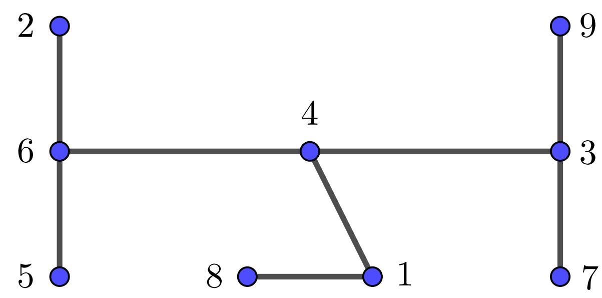


Figure 3: (It should be 6643143)

MSTs

In a graph with weighted edges, we are interested in finding a tree connecting all vertices of the lowest possible weight. We have two algorithms for doing this (the write-up of these algorithms is from the existing notes on the topic).

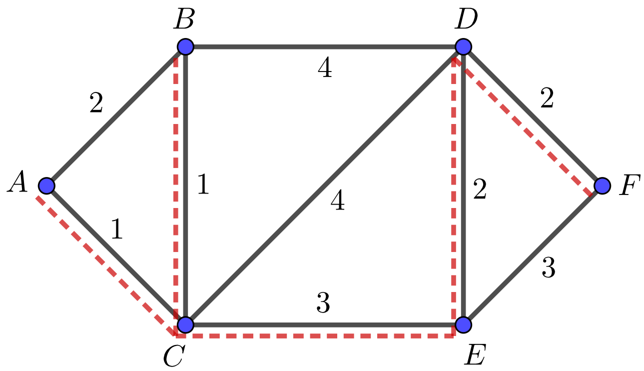
Kruskal's Algorithm

1. Sort all the edges of the graph in non-decreasing order by their weight.
2. Initialize an empty set (or forest) for your MST.
3. Iterate over the sorted edges:
   (a) Let an edge be (u, v) with weight w.
   (b) Check if u and v belong to the same connected component (i.e., whether including this edge forms a cycle).
   (c) If they are not in the same component, include this edge in the MST.
4. Continue until you have $n - 1$ edges in your MST (where n is the number of vertices in the graph).

Prim's Algorithm

1. Select any starting vertex v.
2. While there are vertices not yet in the MST:
   (a) From the set of edges that connect a vertex in the MST to a vertex outside the MST, choose the edge with the smallest weight.
   (b) Add this edge and the new vertex to the MST.

It would be helpful to try a couple examples finding the MST with both algorithms. We ran both on the following graph.

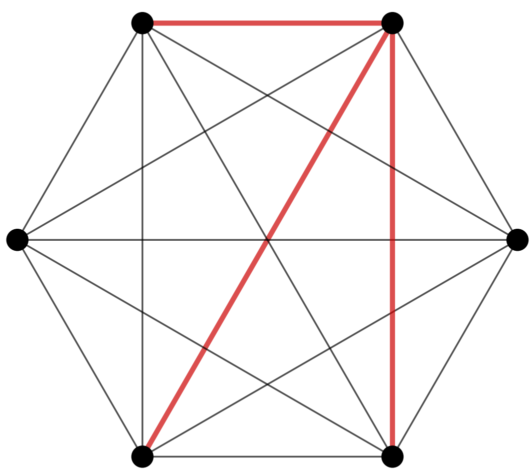


Colouring

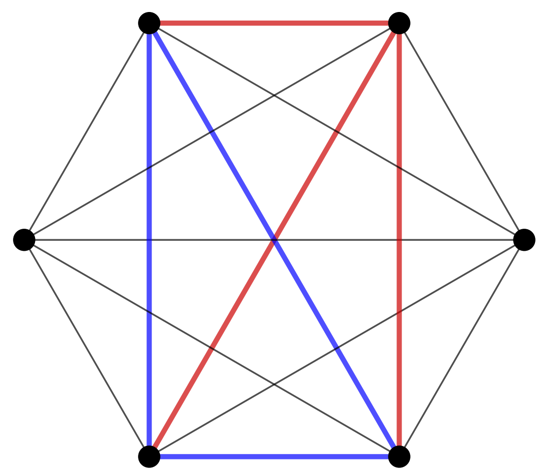
Theorem 3.3 (Brooks'). *$\chi(G) \leq \Delta(G)$ for a connected, simple graph G, unless G is a complete graph or an odd cycle. In that case, $\chi(G) = \Delta(G) + 1$.*

Theorem 3.4 (Vizing). *Let G be a graph. The edge colouring number $\chi_E(G)$ is either $\Delta(G)$ or $\Delta(G) + 1$.*

Example 3.5. *Every 2-colouring of K_6 has a monochromatic triangle.*



(1.) Any arbitrary vertex has 3 edges in one of the colours (say, red)

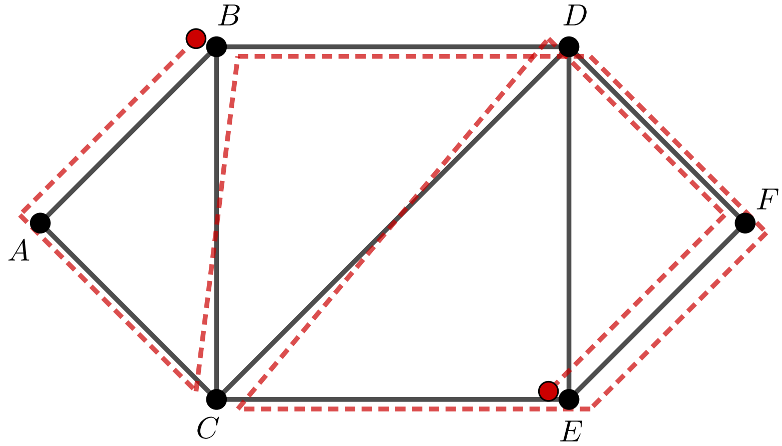


(2.) Then in order to avoid a red triangle, we end up creating a blue one.

Figure 4: Monochromatic triangle proof

Eulerian Trails and Circuits

We quickly went over Hierholzer's algorithm and Fleury's algorithm again. These algorithms are written in the previous sections so I won't restate them here.



Definition 3.6. *A graph is called "Eulerian" if it has an Euler trail or circuit.*

Definition 3.7. *A 2-factor is a spanning subgraph of degree 2 (one or more cycles).*

Theorem 3.8. *If a pseudograph is 4-regular, then G has a decomposition into two 2-factors.*

Proof. Note that the number of edges is even, $|e(G)| = \frac{4|v(E)|}{2} = 2|v(E)|$. The edges an Eulerian circuit, colour it alternatingly red-blue. Because the colours alternate, each vertex has exactly 2 blue and two red edges. So the subgraph consisting of the edges in one of the colours is a two-factor. $\square$

Lecture 4 (Mar. 11)

- "size" of a graph refers to number of edges, "order" refers to number of vertices.
- If $V \subseteq v(G)$, "subgraph induced by V" is the subgraph found by deleting all vertices in G outside of V.
- For a vertex x in a graph, $N(x)$ is the "neighbourhood" of x – the collection of vertices adjacent to x.

Turan's Theorem and Extremal Graph Theory

Problem 4.1. *Find a largest graph G with n vertices and chromatic number two.*

Solution. Since G has chromatic number two, the vertices are coloured by two colors, red and blue. If there is a red vertex that is not adjacent to a blue vertex, we can add this edge and increase the number of edges. Thus we know every blue vertex is adjacent to every red vertex, and we have a K_{m_1,m_2}. Suppose there are m_1 blue and m_2 red vertices, with $m_1 + m_2 = n$. We show that m_1 and m_2 are as close to equal as possible, $|m_1 - m_2| \leq 1$.

Write the difference $D = m_1 - m_2$ for a $D \geq 0$, we want to show $D = 0$ or 1 for the edges to be maximized. The number of edges is $m_1 m_2$, and $n = m_1 + m_2 = D + 2m_2$, so

$$\text{edges} = m_2(n - m_2) = m_2(m_2 + D) = \frac{1}{2}(n - D)\left(\frac{1}{2}(n - D) + D\right) = \frac{1}{4}(n^2 - D^2).$$

□

We can interpret the first equation, edges $= m_2(n - m_2)$, as a continuous equation edges$(x) = x(n - x)$ for n fixed. This is a downward parabola with roots 0 and n, so it's easy to see the max is $n/2$.

Problem 4.2. *Find the largest graph G with n vertices and chromatic number k.*

Solution. We know every vertex is adjacent to every vertex of each other colour again, so we have a k-partite graph $K_{m_1,m_2,\ldots,m_k}$. It is complete, since if it wasn't, we could just add some more edges. The number of edges in this kind of graph is

$$\text{edges} = \sum_{i<j}^{k} m_i m_j = \frac{1}{2}\left(n^2 - \sum_{i=1}^{k} m_i^2\right).$$

The second equation for the number of edges comes from a "complimentary counting": The total number of pairs of vertices (ordered, and repetition allowed) is n^2. The total number of pairs of vertices (ordered, and repetition allowed) from within the i^{th} partition class is m_i^2.

Now, suppose there were two partition classes with $m_i \geq m_j + 2$. Consider a new graph made by transferring one element from the i^{th} to the j^{th} partition class. The new partition classes have sizes $m_1', m_2', \ldots, m_k'$, which are all the same values except $m_i' = (m_i - 1)$ and $m_j' = (m_j + 1)$ The difference in the number of edges is

$$\begin{aligned}
\text{edges} - \text{edges'} &= \frac{1}{2}\left(n^2 - \sum_{\ell=1}^{k} m_\ell^2\right) - \frac{1}{2}\left(n^2 - \sum_{\ell=1}^{k}(m_\ell')^2\right), \\
&= \frac{1}{2}\left((m_i - 1)^2 + (m_j + 1)^2 - m_i^2 - m_j^2\right), \\
&= 1 + m_j - m_i\,, \\
&\leq 1 + m_i - 2 - m_i\,, \\
&= -1.
\end{aligned} \tag{4.1}$$

edges' is larger, so $m_i \geq m_j + 2$ is not allowed in the max graph. So $|m_i - m_j| \leq 1$ for all i, j. □

To summarize, the result of this is a theorem:

Theorem 4.3. *The largest graph on n vertices with chromatic number k is $K_{m_1,m_2,\ldots,m_k}$, with $m = m_1 + m_2 + \ldots + m_k$ and $|m_i - m_j| \leq 1$.*

Now, a stronger version of this theorem. Certainly no graph with chromatic number k contains a subgraph isomorphic to K_{k+1}, but, for instance, a cycle of length five contains no K_3 and still has chromatic number three.

Theorem 4.4 (Turan)**.** *The largest graph with n vertices that contains no subgraph isomorphic to* K_{k+1} *is a complete k-partite graph* $K_{m_1,m_2,\dots,m_k}$*, with* $m = m_1 + m_2 + \dots + m_k$ *and* $|m_i - m_j| \leq 1$.

As a lemma, we first prove this in the case that $k = 2$.

Lemma 4.5 (Mantel's theorem)**.** *The largest graph with n vertices that contains no triangle is the complete bipartite graph* K_{m_1,m_2}*, with* $n = m_1 + m_2$*, and* $|m_1 - m_2| \leq 1$.

Proof. Let G be a graph with no Δ. Let V be the vertex set of G. We choose a vertex of largest degree $x \in v(G)$. Consider

$$N(x) := \{\text{vertices adjacent to } x\}.$$

No vertices in $N(x)$ are adjacent to each other, otherwise Δ exists.

We make a new graph on the same vertices as G, but with different edges. Let $W = N(x)$ in G, and $V = v(G) - N(x)$. Our new graph H is the complete bipartite graph $K_{|V|,|W|}$ on these vertices.

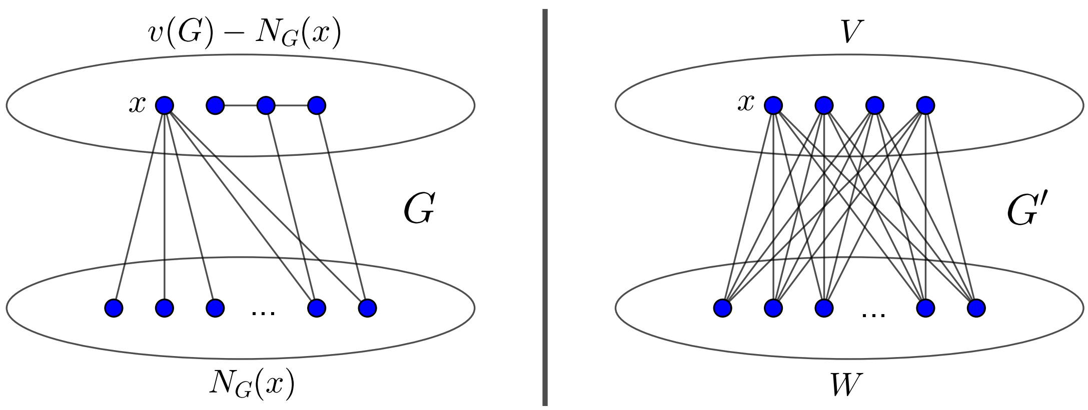


Figure 5: The graphs G vs. G' $(= H)$.

If z is any vertex in V, then

$$\deg_H(z) = \deg_H(x) = \deg_G(x) \geq \deg_G(z)$$

since x was maximal. If z is a vertex in W, then

$$\deg_H = n - |W| \geq \deg_G(z)$$

since no two vertices in W were adjacent in G. Since $\deg_G(z) \leq \deg_H(z)$ for all $z \in v(G)$, the number of edges $|e(H)| \geq |e(G)|$. H is a complete bipartite graph, which we can optimize like in the previous proofs. $\square$

We can now prove Turan's theorem.

Proof. We claim that if G is a graph on n vertices that contains no K_{k+1} then there is a k-partite graph H with the same vertex set as G such that

$$\deg_G(z) \le \deg_H(z)$$

for every vertex z of G.

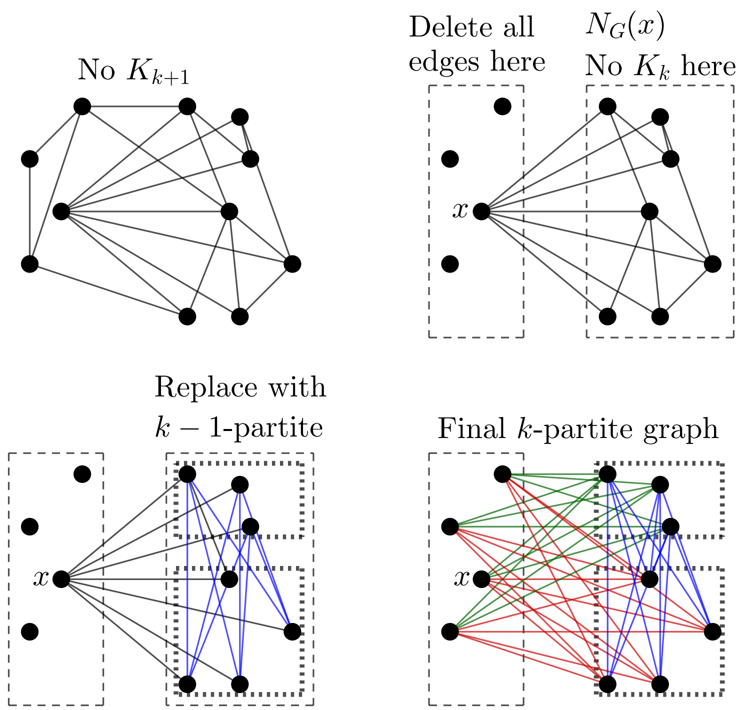
This claim is proven by induction on k. The base case 2 is the lemma. Suppose this holds for values $\le k$, and let G be an n vertex graph not containing K_{k+1}.

Let x be the vertex of max degree in G, and let G_0 be the subgraph induced by $N(x)$. There may be edges here this time, but there can be no K_k since x would be adjacent to all of it, making a K_{k+1}. Thus, there is a $(k-1)$-partite graph H_0 such that

$$\deg_{G_0}(z) \le \deg_{H_0}(z)$$

for all vertices $z \in N(x)$.

Disconnect all vertices in $V := v(G) - N(x)$, and connect them to all vertices in $W = v(H_0)$ to form H.



For vertices in V, we have $\deg_G(z) \le \deg_G(x)$ by maximality of x. For vertices in W, we have

$$\deg_G(z) \le \deg_{G_0}(z) + n - |W|,$$

since there are only $n - |W|$ elements outside W. Also, the induction hypothesis says

$$\deg_{G_0}(z) + n - |W| \le \deg_{H_0}(z) + n - |W|,$$

and finally,

$$\deg_{H_0}(z) + n - |W| = \deg_H(z)$$

since z is connected to everything outside. Therefore, $\deg_G(z) \leq \deg_H(z)$ for all vertices in G as claimed.

The claim now implies that any graph G with no K_{k+1} subgraph can be replaced with a complete k-partite graph with more edges. Optimizing the k-partite graph yields the theorem. □

The optimized k-partite graph is called the *Turan Graph.*

Lecture 5 (Mar. 13)

Summarizing the content from last time, we showed:

- The maximal K_{m_1,m_2} has $|m_1 - m_2| \leq 1$.
- The maximal $K_{m_1,\ldots,m_k}$ has $|m_i - m_j| \leq 1$ for all i, j.
- K_{m_1,m_2} has the max edges for Δ-free.
- $K_{m_1,\ldots,m_k}$ has max edges for K_{k+1}-free.
- For a graph G, if you can find a graph H on $v(G)$ so that $\deg_H(v_i) \geq \deg_G(v_i)$ for all vertices, H has more edges than G.

Today we look at some further extremal problems.

Lemma 5.1 (Cauchy-Schwartz inequality)**.** *For two sets of numbers* $\{a_1, \ldots a_k\}$, $\{b_1, \ldots b_k\}$,

$$\left(\sum_{i=1}^{k} a_i b_i\right)^2 \leq \left(\sum_{i=1}^{k} a_i^2\right)\left(\sum_{i=1}^{k} b_i^2\right).$$

An implication is: when summing a bunch of numbers squared, the result is always larger than their sum times the average value. One can understand this in terms of dot products: considering $U = (a_1, \ldots a_k)$, $V = (b_1, \ldots b_k)$ as vectors, the inequality is just $U \cdot V \leq |U||V|$.

Theorem 5.2 (Kővári-Sós-Turán, special case)**.** *If G_n contains no $K_{2,X}$, then it has at most $\sqrt{X} n^{3/2}$ edges.*

Proof. The proof works by counting "cherries" in two different ways, by their stems, and by the pairs of the berries themselves. A cherry is just a path of length 2, it feels right to call these paths cherries in the context of the proof.

(1.) A cherry (length 2 path)

(2.) A cherry in the verts. adjacent to v_1.

Figure 6: Cherries on a vertex v_1.

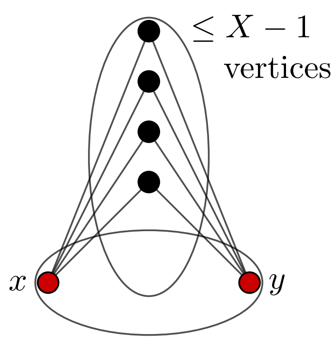
The number of cherries for a given vertex v_i in G_n is $\binom{d_i}{2} \geq \frac{1}{4}d_i^2$ (we are using that $\binom{d_i}{2} \geq \frac{1}{4}d_i^2$ only to simplify things a little). So the total number of cherries is

$$\#\text{cherries} = \sum_{i=1}^{n} \binom{d_i}{2} \geq \frac{1}{4}\sum_{i=1}^{n} d_i^2.$$

So, using Cauchy-Schwartz, we have

$$\#\text{cherries} \geq \frac{1}{4}\sum_{i=1}^{n} d_i^2 \geq \frac{1}{4}\cdot\frac{1}{n}\left(\sum_{i=1}^{n} d_i\right)^2 = \frac{1}{4}\cdot\frac{1}{n}\left(2|e(G)|\right)^2 = \frac{|e(G)|^2}{n}.$$

On the other hand, each pair of vertices have at most $X-1$ stems joining them, otherwise they would form $K_{2,X}$ with those stems.



Therefore,

$$\#\text{cherries} \leq (X-1)\binom{n}{2} \leq Xn^2.$$

Combining these bounds gives us $\frac{|e(G)|^2}{n} \leq \#\text{cherries} \leq Xn^2$, and simplifying gives us exactly $|e(G)| \leq \sqrt{X}n^{3/2}$. □

Theorem 5.3 (Mantel's theorem alternative statement)**.** *If G_n contains no triangle then it has at most $\frac{n^2}{4}$ edges.*

Proof. Let x and y be vertices in G which are joined by an edge. We see that $\deg(x)+\deg(y) \leq n$. This is because every vertex in the graph G is adjacent to at most one of u, v, so there are at most

$n-2$ edges connecting x and y to outside vertices, and the edge between them which adds one degree to each. Note now that

$$\sum_{x \in v(G)} d^2(x) = \sum_{(x,y)\in e(G)} (\deg(x) + \deg(y)) \leq |e(G)|n.$$

We can see the first equality here by considering how many times $\deg(x)$ for a particular vertex appears on the right hand side. For every edge x is in, we add $\deg(x)$ to the sum. So $\deg(x)$ appears in the sum $\deg(x)$ times, so its contribution is $\deg^2(x)$, like in the left hand side.

On the other hand, since $\sum_{x\in v(G)} d(x) = 2|e(G)|$, the Cauchy-Schwartz inequality implies that

$$\sum_{x\in v(G)} \deg^2(x) \geq \frac{1}{n}\left(\sum_{x\in v(G)} \deg(x)\right)^2 \geq \frac{4|e(G)|^2}{n}.$$

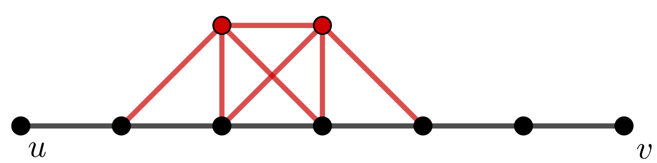
So,

$$\frac{4|e(G)|^2}{n} \leq |e(G)|n,$$

and the result follows. □

Problem 5.4. *Find a largest graph with* 9 *vertices and diameter* 6.

Solution. Start with a path of length 6 with end points u and v, which uses up 7 of the vertices. No other edges can be added between the vertices on the path. The two additional vertices can each be attached to at most three element of the path each, otherwise, there would be a shorter path from u to v. They can also be attached to each other, so there are a max possible 6+3+3+1=13 edges. The following figure shows we can realize that.



□

Problem 5.5. *Find a largest graph G satisfying*

1. $|v(G)| = 8$, *diameter four.*
2. $|v(G)| = 7$, *no cycle of length greater than three.*
3. $|v(G)| = 7$, *girth four.*
4. $|v(G)| = 9$, *exactly one triangle (3-cycle).*
5. $|v(G)| = 7$, *exactly one square (4-cycle).*
6. $|v(G)| = 8$, *girth 6.*

Lecture 6 (Mar. 18)

Problem 6.1. *Reviewing the two remaining questions from last class, find a largest graph G satisfying*

1. $|v(G)| = 7$, *no cycle of length greater than three.*

2. $|v(G)| = 9$, *exactly one triangle (3-cycle).*

Solution.

1. The max number of edges is 9! We can argue this in several steps. (This case-by-case is slightly more involved than the one we actually did in class.)

   - What if there are no cycles? Then its a tree, and the best possible is 6 edges.
   - Just one cycle? Then removing one edge from that cycle would leave you with a tree, so there's at most 7 edges.
   - What if there are two triangles that don't share a vertex? You can connect the remaining vertex to both with at most one edge, otherwise you'd make a C_4. Then you can also add a vertex connecting the triangles together, where the extra vertex is connected. This gives 9 edges.

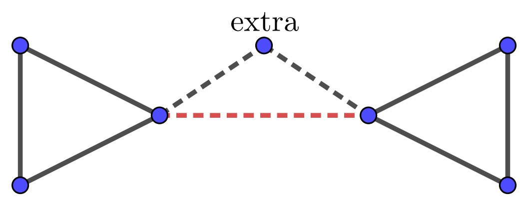
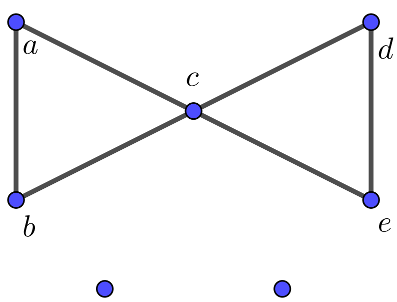
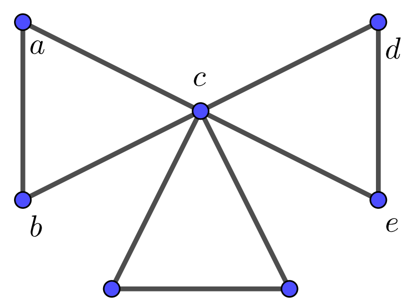


Figure 7: The result from 2 disjoint triangles

   - Lastly, what if we try two triangles joined at one vertex. No other edges can join vertices in a, b, c, d, e. The remaining two vertices, can each be joined to a, b, c, d, e in at most one spot, otherwise we would make a C_4. The remaining two could also be joined to each other, so there is a max of 9 edges possible. By connecting them both to c, we can realize this.

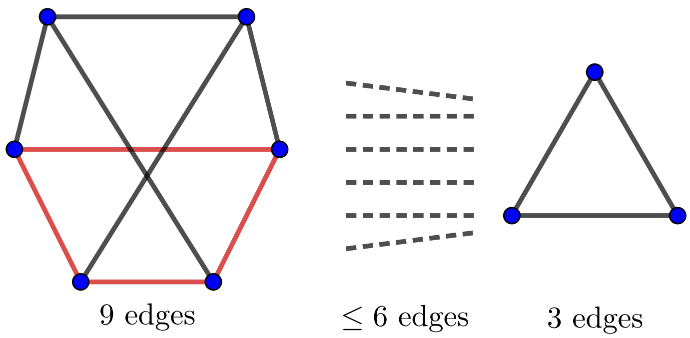


(1.) Two triangles joined at a vertex



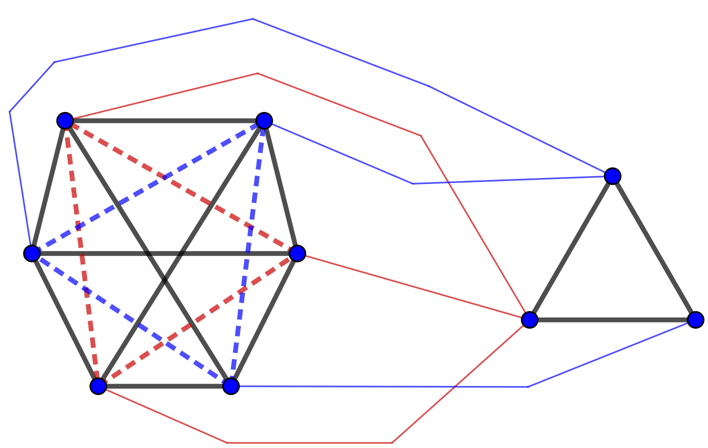
(2.) Nine edges

2. Consider that the graph with the triangle removed will be girth at least 4. We first find what is the maximum number of edges on a girth 4 graph with $|v(G)| = 6$. To do so, start with a cycle of length 4. The two other vertices can be attached to the cycle in at most 2 places each, and to each other. We show a configuration that realizes this below, with 9 edges.



Now, we have a triangle, and a girth 4-graph. Clearly, at most 6 edges connect the triangle to the girth 4 graph, since no two edges from the triangle can go to the same vertex. Therefore, the max number of edges possible is $9 + 6 + 3 = 18$. All we need to do is show a realization of this many edges now, and we'll be done. But it's worth thinking through the process of finding the connections.

One would be tempted to attach two edges to each vertex of the triangle, but this won't work. Every pair of edges attached to the same vertex on the triangle must be attached to an 'anti-edge' (just a pair of vertices with no edge between them) in the girth-4 component, otherwise we make a triangle. The anti-edges form two triangles. So, it's not possible to connect two edges from each vertex on the triangle to all vertices in the girth-4 component. We can achieve 18 edges by attaching 3 edges to one triangle vertex, 2 to the second, and 1 to the third.



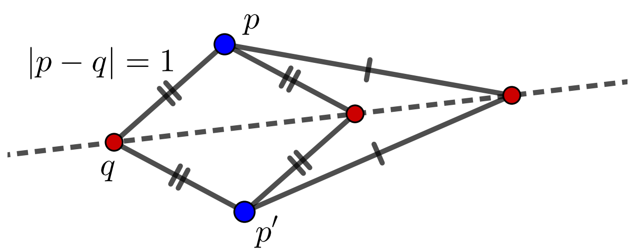
□

Theorem 6.2. *Define the* max unit distances *function, which is the maximal possible number of times a distance 1 can appear between a finite number n points in the plane.*

$$u(n) = \max_{P \subset \mathbb{R}^2,\ \text{number of points in } P = n} \# \{p, q \in P : |p - q| = 1\} \tag{6.1}$$

Then $u(n) \leq \sqrt{3} n^{3/2}$.

Proof. Given any set of points P in the plane, define their "unit distance graph". The vertices of the graph are the points in P, and there is an edge between $p, q \in P$ if and only if $|p - q| = 1$.



This graph has no $K_{2,3}$. The collection of points equidistant to p and p' above is their perpendicular bisector, and it's not possible that more than two points on this bisector are the same distance away from p and p'. $\square$

For example, we know $u(3) = 3$, $u(4) = 5$, $u(7) = 12$, and $u(9) = 18$. In fact, all values up to $n = 21$ are known and proven to be optimal, and the values we have found up to $n = 112$ are expected to be the best possible.

Cages

Problem 6.3. *What is the minimal graph with girth g?*

Problem 6.4. *What is the minimal 3-regular graph with girth g? This not something we can answer in general!*

Problem 6.5. *What is the minimal 3-regular graph with*

1. *girth 3? Answer is K_4.*
2. *girth 4? Answer is $K_{3,3}$.*
3. *girth 5? Answer is Petersen graph. We went through the argument showing this in class, it's the same one that's in the book.*
4. *girth ≥ 6? We saw the known examples in some slides, which have been uploaded separately.*

Ramsey's Theorem

Definition 6.6. *The* Ramsey number $R(s, t)$ *is the smallest integer with the property that any red-blue edge-coloring of* $K_{R(s,t)}$ *must contain a red* K_s *or a blue* K_t.

It's certainly true that $R(2, 2) = 2$. How about $R(2, X)$? We showed before that any red-blue edge colouring of K_6 has a monochromatic Δ, so $R(3, 3) \leq 6$.

Interesting questions:

1. Determine $R(s, t)$ exactly for small values s, t. We only know a handful of them. $R(4, 4) = 18$, but we don't know higher exactly: $43 \leq R(5, 5) \leq 46$, and $102 \leq R(6, 6) \leq 160$.

   "Erdős asks us to imagine an alien force, vastly more powerful than us, landing on Earth and demanding the value of $R(5, 5)$ or they will destroy our planet. In that case, he claims, we should marshal all our computers and all our mathematicians and attempt to find the value. But suppose, instead, that they ask for $R(6, 6)$. In that case, he believes, we should attempt to destroy the aliens." (Joel Spencer)

2. Determine the asymptotics of $R(s, s)$.

3. (Not discussed in class!) Determine the asymptotics of $R(X, t)$ for small fixed values of X. This is very hard as well. Just one year ago, it was shown that

$$\frac{t^3}{\log^4(t)} \leq r(4, t) \leq \frac{t^3}{\log^2(t)}.$$

   (the lower bound was last year, upper was 2001) which nearly solves this case.

Theorem 6.7 (Ramsey)**.** *For positive integers* s, t*, the Ramsey Number* $R(s, t)$ *is finite.*

Lecture 7 (Mar. 20)

We first reintroduced the definition of Ramsey Numbers (Definition 8.10), and the statement of theorem (Theorem 6.7). We now prove it!

Proof. We will use induction. The base cases are that $R(2, X) = X$ for any integer X. To see this, just note that in a complete graph on X vertices, colouring a single edge red would result in a red K_2. But if no edges are coloured red at all, we have a blue K_X. We also need to understand that $R(s, t) = R(t, s)$.

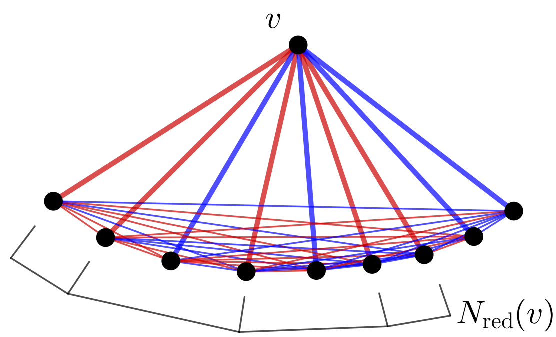
Next, for the induction step, we claim that $R(s, t) \leq R(s - 1, t) + R(s, t - 1)$.

Let $n = R(s - 1, t) + R(s, t - 1)$ and consider any red/blue colouring of the edges of K_n. Fix a vertex v, and let $N_{red}(v) = \{u \in N(v) : (u, v) \text{ is red}\}$, $N_{blue}(v) = \{u \in N(v) : (u, v) \text{ is blue}\}$.

Notice that

$$|N_{red}(v)| + |N_{blue}(v)| = n - 1 = R(s-1,t) + R(s,t-1) - 1$$

because the neighbourhoods contain all vertices, except for v. Consider two cases:



- There are at least $R(s-1,t)$ red edges connected to v, that is, $|N_{red}(v)| \geq R(s-1,t)$. The definition of $R(s-1,t)$ implies that $N_{red}(v)$ either contains a red copy of K_{s-1} or a blue copy of K_t. In the first case, adding v to the red K_{s-1} creates a red K_s, since all edges joining v to this set are red. In the second, we have a blue K_t.

- $|N_{red}(v)| \leq R(s-1,t) - 1$. In this case, we have the following:

  $$R(s-1,t) + R(s,t-1) - 1 = |N_{red}(v)| + |N_{blue}(v)| \leq R(s-1,t) - 1 + |N_{blue}(v)|.,$$

  and after cancelling,

  $$|N_{blue}(v)| \geq R(s,t-1).$$

  The definition of $R(s,t-1)$ implies $N_{blue}(v)$ has a red K_s or blue K_{t-1}, and we are done by adding v to the blue K_{t-1} in the latter case.

□

Overall, it is known that

$$c_1\sqrt{2}^s \leq R(s,s) \leq c_2 4^s.$$

You can show this upper bound using the recursive relationship from the proof. Now, it's interesting that our proof seems quite basic, so one would expect the 4^s could be improved. This problem resisted all attempts until very recently (2023), a huge breakthrough improved this to

$$c_1\sqrt{2}^s \leq R(s,s) \leq c_2 3.999^s.$$

The method has since been improved, and we have lowered that constant by a bit more.

The point of Ramsey numbers: The theorem explains that in a large enough system, it's impossible not to have very structured parts. This might be an intuitive phenomenon, but Ramsey's theorem is rigorous and quantifiable.

Another equivalent statement of the Ramsey numbers: *Every graph G_n has an independent set or a complete subgraph of size at least $m := \lfloor \log_4(n) \rfloor$.* To see this, consider all the edges currently in your graph as red edges. Add all the missing edges to the graph until it is complete, colouring the newly added edges blue. Because $n = 4^{\log_4 n} \geq 4^m$, we know this graph has a complete subgraph that's entirely red, or entirely blue. If its entirely red, this complete graph is made of edges that were in the original graph G_n, so G_n has a complete subgraph of size m. If the subgraph is entirely blue, then none of the edges at all was part of G_n, so the vertices form an independent set in G_n.

Another application: Every set of points $P \subset \mathbb{R}^2$ with $|P| = n$ and no three in a line, has a "long" increasing sequence or a decreasing sequence. To see this one, you create a complete graph where the vertices are the points. The edge between two points is coloured red if the segment joining the points has positive slope. Ramsey's theorem says that there is either a red K_m or a blue K_m, where m is around $\lfloor \log_4 n \rfloor$. A monochromatic K_m in such a graph is forced to be an entirely increasing sequence, or a decreasing one. (Not discussed in class: there is a simpler show a result like this, without Ramsey's theorem, that gets a much better quantitative bound!)

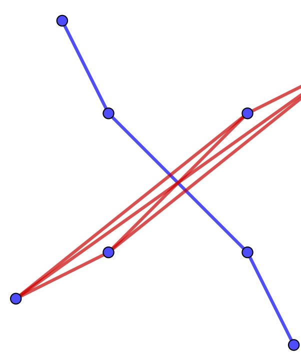


Figure 9: A decreasing sequence in blue, and an increasing K_4 in red

The Happy Ending Problem

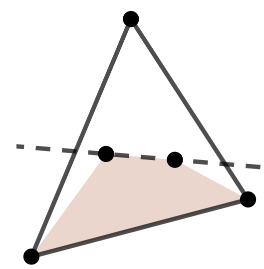
For this final beautiful application, we are going to define a couple things that will only be used in this section. The first:

Definition 7.1. *A point set $P \subset \mathbb{R}^2$ is called "convex" if the points can form a polygon, with all interior angles less than* 180 *degrees.*

This problem states that every set of points in the plane has convex subset. As a warmup:

Problem 7.2. *Show that any* 5 *points in the plane, no three in a line, has a convex subset of* 4 *points.*

To solve this, we first argue that the set must look like a triangle with two points inside, otherwise the 'outside points' would form a convex set of 4. Then, connect the two inside points with a line. This line hits two sides of the triangle, and the 2 points determining the third side are in convex position with those 2 interior points.



Now, the goal for the rest of this section is to solve the following problem.

Problem 7.3 (Happy Ending)**.** *Show that for any integer m, there is an n large enough so that for any set P of n points in the plane, no three in a line, there is a convex subset with $\geq m$ points.*

Doing the previous thing doesn't seem to work, as giving a colour to one pair of points doesn't seem to allow us to identify whether it's "in a convex set" or not. We would want something that would allow us to capture the structure of at least 4 points being in convex position at once...

Definition 7.4 (Hypergraph)**.** *A k-hypergraph is a collection of objects (vertices) and connections between k-tuples of these objects (edges).*

Notice that if $k = 2$, this is exactly the definition of a graph! Hypergraphs are not so different, but they are hard to draw, and many simple notions are much more complicated for them. Therefore we will only learn one single thing about them: that Ramsey's theorem holds for them as well.

K_n^k is the notation we'll use for the complete k-hypergraph on n vertices, which is n vertices with every k-tuple connected by an edge. For example, K_4^3 consists of 4 points, and each 3-tuple of points is joined by an edge.

Definition 7.5. *The k-*Ramsey number *$R^k(s,t)$ is the smallest integer with the property that any red-blue edge-coloring of $K^k_{R(s,t)}$ must contain a red K_s^k or a blue K_t^k.*

Theorem 7.6. *For positive integers s, t, k, the Ramsey Number $R^k(s,t)$ is finite.*

We can now solve Problem 7.3 in full. Consider the complete 4-hypergraph K_n^4, with an edge coloured red if its 4-tuple is convex, and blue if its 4-tuple is not convex. We choose n to be $n = R^4(m, 5)$. Then, by definition, the colouring we've defined either has a red K_m^4 or a blue K_5^4. In the second case, the colouring has a blue K_5^4, which means we have 5 points where every 4 of them is not convex. But we showed in problem 7.2 that this can never happen!

Because of that, the only possibility is that we are in the first case, where we have a red K_m^4, which means that we have m points where every single 4-tuple of them is convex. Because of this, the entire m points of this K_m^4 are in convex position!

Lecture 8 (Mar. 25)

On the midterm: Most things from the course are covered. The exception is Prufer codes! These will return on the final.

This review contains only things post-reading break, with one exception:

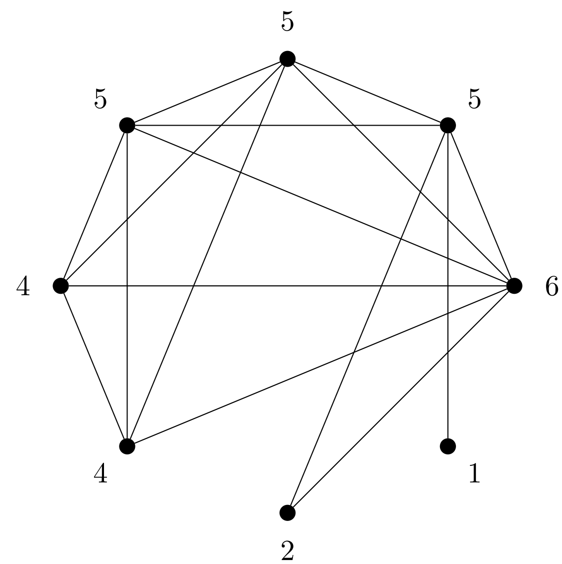
Midterm review: Degree sequences+Eulerian trails

The first thing to remember is how to construct sequences that are graphic. We only did the following one, it would be worth it to practice a couple more.

Problem 8.1. *Reconstruct the graphic sequence* $(6, 5, 5, 5, 4, 4, 2, 1)$.

Solution.

$$
\begin{array}{r}
(6, 5, 5, 5, 4, 4, 2, 1) \\
(4, 4, 4, 3, 3, 1, 1) \\
(3, 3, 2, 2, 1, 1) \\
(2, 1, 1, 1, 1) \\
(0, 0, 1, 1) \\
(1, 1, 0, 0) \\
(0, 0, 0)
\end{array}
$$



□

Theorem 8.2. *A graph G has an Eulerian trail iff it is connected and has precisely 2 vertices of odd degree. G has an Eulerian circuit iff it is connected and all vertices are even degree.*

For example, the graph we constructed in the previous problem has 4 vertices of odd degree, so it does not have an Eulerian trail.

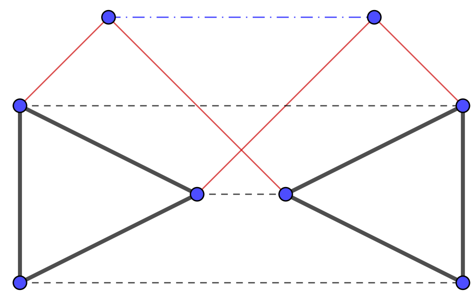
Midterm review: Extremal graphs

Problem 8.3. *How many edges at most on a graph with $|v(G)| = 8$ and exactly two triangles?*

Solution. This one ended up being complicated! We break into cases, based on whether the triangle subgraphs are disjoint, connected by a vertex, or connected by two vertices (and an edge).

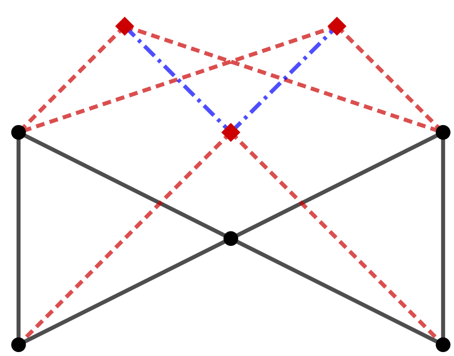
- The two triangles are disjoint. Then they use 6 of the vertices and make 6 edges.
  - There are at most 3 edges between the two triangles.
  - There is at most 1 edge joining each outside vertex to each triangle, so there's at most 4 edges joining outside the vertices to the triangles.
  - The two outside vertices may have an edge between them.

  This means this case has a maximum of 14 edges. We can realize that in the figure below.



- The two triangles share a single vertex.
  - There are no more edges between the triangles.
  - There is at most 1 edge joining each outside vertex to each triangle, so there's at most 6 edges joining outside the vertices to the triangles.
  - The three outside vertices may have at most two edges between them, otherwise they would make a triangle.

  This means this case has a maximum of 14 edges. We can realize that in the figure below.



- The two triangles share two vertices. Call this configuration of 4 points H, and call the other four vertices P.
  - There are no more edges between vertices in H.

– There is at most 2 edges joining each vertex of P to H, for a total of 8 edges. This is only possible by connecting to both of the only two disjoint vertices in H.

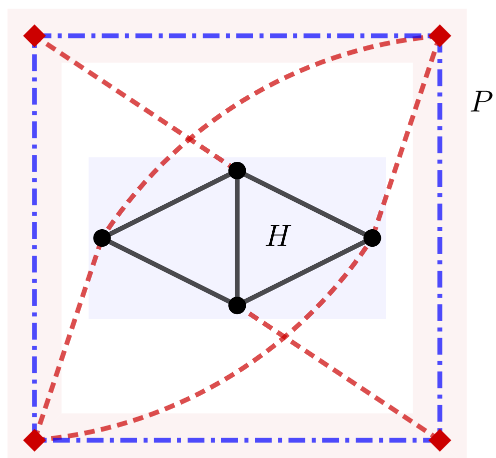
– The four vertices of P may have at most four edges between them in a C_4 pattern, otherwise they would make a triangle.

This is a maximum of 17 edges. However, note that if two vertices outside have an edge between them, its not possible to connect both of them to the two available spots in H.

– Connecting just two from P to H with two edges is is possible, using two opposite vertices in the C_4, which results in 15 edges.

– Connecting three vertices of P to the two spots in H results in in at most 3 edges outside H (by joining the final vertex to the three joined ones), and thus at most 15 edges as well.

– and connecting all 4 results in 0 edges possible outside H, resulting in only 13 edges.



This means this case and in this problem, any valid graph has a maximum of 15 edges. We realize that bound in the figure above. □

Definition 8.4. *The Turán graph on n vertices $T_{n,k}$ is the complete k-partite graph, with vertices evenly distributed.*

Theorem 8.5 (Turán's)**.** *The maximal graph G_n with no K_{k+1} is the Turán graph $T_{n,k}$.*

Problem 8.6. *How many edges does the Turán graph $T_{n,k}$ have? Assume k divides n if it helps.*

Solution. By counting the total number of pairs of points in the graph (repetitions allowed), and subtracting all the pairs within the same partite classes, we get

$$\#\text{edges} = \frac{1}{2}\left(n^2 - \sum_{i=1}^{k} m_i^2\right) = \frac{n^2}{2}\left(1 - \frac{1}{k}\right). \tag{8.1}$$

If k does not divide n, $\frac{n^2}{2}\left(1-\frac{1}{k}\right)$ is still nearly the number of edges. But for completeness, if we write $n = d + \frac{r}{k}$ for some $0 < r < k$, (so r is the remainder), we can work through some steps to show the precise value

$$\#\text{edges} \ = \frac{n^2 - r^2}{2}\left(1 - \frac{1}{k}\right) + \binom{r}{2}.$$

Often, we are just interested in the number of edges in a graph where n is large. So if k is some small constant number, then r is tiny compared to n so Equation (8.1) is a very good estimate. □

Theorem 8.7 (Mantel's)**.**

- *The maximal graph with no triangle is the balanced complete bipartite graph.*
- *A graph with no triangle G_n has at most $\frac{n^2}{4}$ edges.*

Theorem 8.8 (Kővári-Sós-Turán)**.** *If G_n contains no $K_{2,X}$, then it has at most $\sqrt{X}n^{3/2}$ edges.*

Problem 8.9. *Let $P \subset \mathbb{R}^2$ be a set of n distinct points, and let L be a set of n distinct lines in $\mathbb{R}^2$. Give an upper bound on the number of point line pairs (p, ℓ) where the point lies on the line.*

Solution. With any problem like this, the question will give us a collection of objects, and a relationship that those objects can either satisfy, or not. In this case, our objects are points and lines, and the relationship is the point lying on the line. Therefore, we consider a complete bipartite graph, with vertices representing both points and lines. We connect two vertices with an edge, if one vertex is a point, the other is a line, and the point lies on the line.

This graph cannot have a $K_{2,2}$. If it did, then we would have two points, each connected to two lines. This would correspond to two distinct points both contained in two distinct lines. But at most a single line goes through any pair of points, so this is impossible! Theoren 8.8 now implies there are at most $\sqrt{2}(2n)^{3/2}$ of these points-line pairs. □

Definition 8.10. *The* Ramsey number $R(s,t)$ *is the smallest integer with the property that any red-blue edge-coloring of $K_{R(s,t)}$ must contain a red K_s or a blue K_t subgraph.*

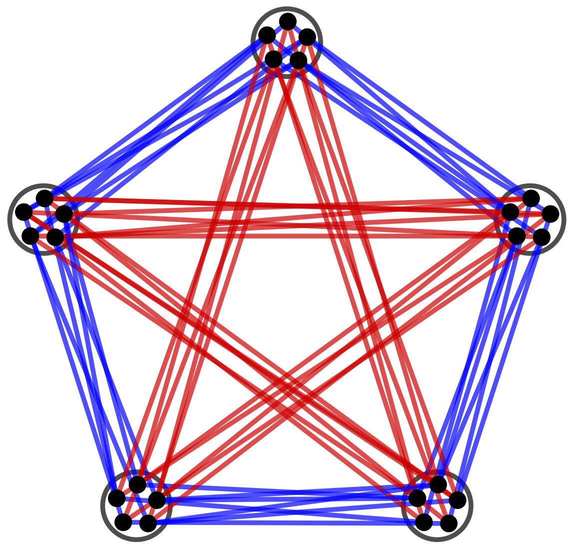
Problem 8.11. *(Stated in class but not shown) A graph G_n can be coloured red and blue, such that there is no monochromatic triangle. At most how many edges are in G_n? (The answer will depend on the number of vertices n)*

Solution. Because we want to avoid triangles, you might be tempted to apply Mantel's theorem. Indeed, considering the red edges alone, Mantel's theorem says there are at most $\frac{n^2}{4}$ red edges and the maximal graph is the complete bipartite graph. Then, the only place to add more edges is within the two partite classes. There are $n/2$ vertices in each, so Mantel's theorem says we can add at most $\frac{n^2}{16}$ blue edges to each of the two partite classes. Altogether, there are

$$\frac{n^2}{4} + \frac{n^2}{16} + \frac{n^2}{16} = \frac{3n^2}{8} = 0.375n^2$$

edges at most.

Perhaps surprisingly, this is not optimal. There are many more red edges than blue in that construction, so it would be better to even things out. What you must do to solve this problem is recall that any colouring of K_6 red and blue has a monochromatic triangle. So, our graph certainly cannot contain any K_6. Turán's theorem says the maximal number of edges in a graph avoiding K_6 is the Turán graph $T_{n,5}$. And we can demonstrate a colouring of $T_{n,5}$ which does avoid monochromatic triangles by extending the colouring for a K_5, as below.



From Equation (8.1), this graph has $\frac{n^2}{2}(1 - \frac{1}{5}) = 0.4n^2$ edges. □

Lecture 9 (April 1st)

Max-Flow Min-Cut

Definition 9.1 (Directed graph)**.** *A simple directed graph ("digraph") D, consists of a set of vertices $v(D)$ and edges $e(D)$, where edges are* ***ordered*** *pairs $e = (u, v)$. We call*

- *u the tail of e,*
- *v the head of e, and*
- *u, v the ends of e.*

Definition 9.2 (Underlying graph and Orientation)**.** *If D is a digraph, the graph G with $v(G) = v(D)$ and $e(G) = e(D)$ is the* underlying (pseudo-)graph *of D. We call D an* orientation *of G.*

Definition 9.3 (Digraph degree and neighbourhood)**.**

- *The outneighbourhood $N_{out}(v)$, is the edges with tail v, and outdegree $d_{out}(v)$ is $|N_{out}(v)|$.*

- *The inneighbourhood $N_{in}(v)$, is the edges with head v, and indegree $d_{in}(v)$ is $|N_{in}(v)|$.*

Some common digraphs are:

- Directed Path (length n): A graph with $v(G) = \{v_1, \dots, v_{n+1}\}$ and edges may be numbered $\{e_1, \dots, e_n\}$ so that $e_i = (v_i, v_{i+1})$.
- Directed Cycle (length n): A graph with $v(G) = \{v_1, \dots, v_n\}$ and edges may be numbered $\{e_1, \dots, e_n\}$ so that $e_i = (v_i, v_{(i+1 \mod n)})$.
- Rooted tree: connected, and all vertices have $d_{in}(v) = 1$, except one (the root), which has $d_{in}(v) = 0$.
- Tournament: A digraph whose underlying graph is complete

Theorem 9.4 (Handshaking)**.** *For every digraph D,*

$$\sum_{v \in v(D)} d_{out}(v) = \sum_{v \in v(D)} d_{in}(v) = |e(D)|. \tag{9.1}$$

Definition 9.5 (Walks)**.** *A directed walk of length n in a digraph D is a sequence $v_0, v_1, v_2, \dots, v_n$ so that $v_i \in v(D)$ and $(v_{i-1}, v_i) \in e(D)$. If*

- *$v_0 = v_n$ the walk is closed,*
- *v_i are all distinct it's a directed path,*
- *(v_{i-1}, v_i) are all distinct it's a directed trail.*

Definition 9.6 (Connected)**.**

- *A digraph D is weakly connected if the underlying graph is connected.*
- *Two vertices $u, v \in v(D)$ are connected if there is a directed path from u to v.*
- *D strongly connected if every $u, v \in v(D)$ are connected.*

Definition 9.7 (Network)**.** *A network $N(D, s, t, c)$ consists of*

- *A directed graph D.*
- *A source vertex $s \in v(D)$ and a sink $t \in v(D)$. $N_{in}(s)$ is empty, $N_{out}(t)$ is empty (some references don't require this).*
- *A capacity function: a mapping $c : e(D) \mapsto \mathbb{R}^+$ denoted by $c(u, v)$ for $(u, v) \in e(D)$. c is the max volume that can pass through an edge per unit of time.*

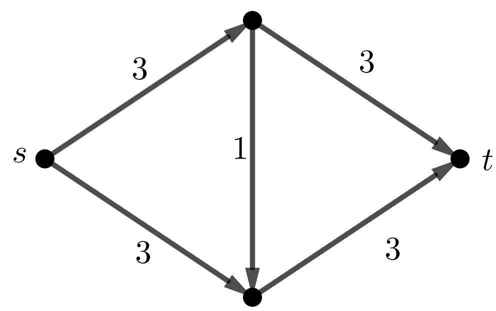


Figure 10: An example of drawing a network

Definition 9.8 (Flow)**.** *A flow in a network is a mapping $f : e(D) \to \mathbb{R}^+$ denoted by $f(u,v)$, subject to the following two constraints:*

- *Capacity Constraint: For every edge $(u,v) \in e(D)$, $f(u,v) \leq c(u,v)$.*
- *Conservation of Flows: For each vertex v apart from s and t (the source and sink, respectively), the following equality holds:*

$$\sum_{u \in N_{in}(v)} f(u,v) = \sum_{w \in N_{out}(v)} f(v,w).$$

The value *of the flow $|f| \geq 0$ is the total flow reaching the sink,*

$$|f| = \sum_{u \in N_{in}(t)} f(u,t).$$

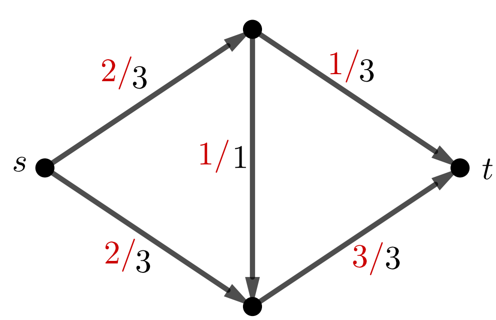


Figure 11: An example of communicating flow on the network from Fig. 10

The capacity constraint then says that the volume flowing through each edge per unit time is less than or equal to the maximum capacity of the edge, and the conservation constraint says that the amount that flows into each vertex equals the amount flowing out of each vertex, apart from the source and sink vertices.

Typically, the source will not be the head of any vertex, and the sink will not be the tail of any vertex. We could modify this definition slightly to allow for multiple sources and sinks.

Proposition 9.9. $|f| = \sum_{u \in N_{out}(s)} f(s,u)$.

Proof. This is a consequence of the conservation of flows:

$$0 = \sum_{(u,v)\in e(D)} f(u,v) - f(u,v) = \sum_{u \in v(D)} \left(\sum_{v \in N_{out}(u)} f(u,v) - \sum_{v \in N_{in}(t)} f(w,u) \right),$$

where we reorganized the edges into the vertices' neighbourhoods. The only contributions that don't cancel are

$$0 = \sum_{u \in N_{out}(s)} f(s,u) - \sum_{u \in N_{in}(t)} f(u,t) = \sum_{u \in N_{out}(s)} f(s,u) - |f|.$$

□

Problem 9.10 (Max flow)**.** *Given a network $N(D,s,t,c)$, find a flow with maximal $|f|$. That is, route as much flow as possible from s to t.*

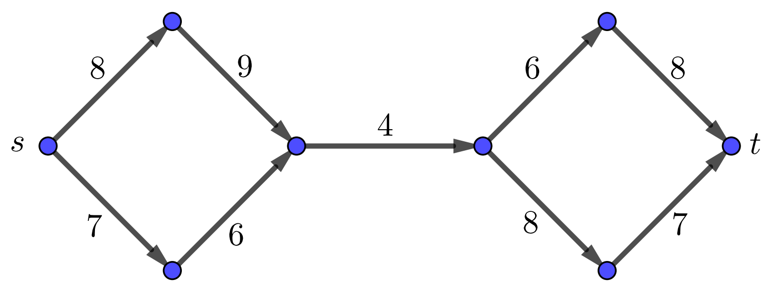


Figure 12: An example where it's easy to tell the max flow possible

Definition 9.11 (Residual network)**.** *Given a network $N(D,s,t,c)$ and a flow on it, f. The residual network $N_f(D,s,t,c_f)$ is a network on the same digraph, with $c_f(u,v) = c(u,v) - f(u,v)$. Further, if $(u,v) \in e(D)$, the new capacity function also records the "reverse capacity" $c_f(v,u) = f(u,v)$ for every edge in the path.*

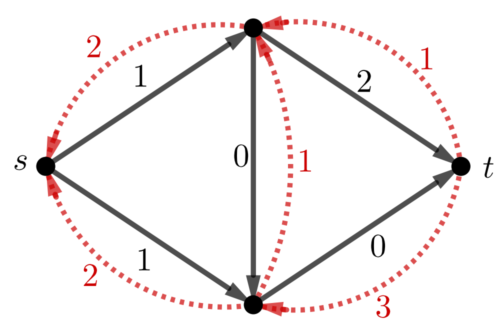


Figure 13: The residual network of Figure 11

Theorem 9.12 (Ford-Fulkerson Algorithm)**.** *Given a network $N(D, s, t, c)$ Consider the following algorithm:*

1. *Create a flow f, all edges 0 flow.*
2. *Update the residual network $N_f(D, s, t, c_f)$.*
3. *If there is a path P in the underlying graph (**not a directed path**) from s to t where $c_f(u, v) > 0$ for every $(u, v) \in P$:*
   - *find $b_f(P) = \min_{(u,v)\in e(P)} c_f(u, v)$*
   - *For each edge (u, v) in the path:*
     - *if $(u, v) \in e(D)$: set $f(u, v) = f(u, v) + b_f(P)$*
     - *otherwise, $(v, u) \in e(D)$, so: set $f(v, u) = f(v, u) - b_f(P)$*
   - *Go back back to step 2.*

This algorithm finds the max flow, if it terminates.

Lecture 10 (April 3rd)

First we slow down, and think about how we could find a good flow most simply.

Naive algorithm: Given a Network $N(D, s, t, c)$,

1. Initialize an empty flow
2. If there is a directed path from s to t with $c(u, v) - f(u, v) > 0$ for all edges:
   - Find $b_f(P) = \min_{(u,v)\in e(P)} (c(u, v) - f(u, v))$
   - For $(u, v) \in e(P)$: set $f(u, v) = f(u, v) + b_f(P)$.

Why does this algorithm not work so well? It can get stuck with sub-optimal flows!

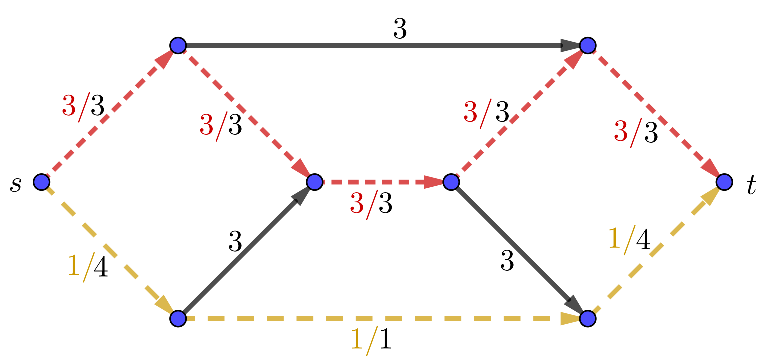


Figure 14: A network with two naively chosen directed paths, no more can be found

We then gave the full algorithm (Theorem 9.12) again. Also, from the step: "If there is a path P in the underlying graph from s to t where $c_f(u,v) > 0$ for every $(u,v) \in e(P)$", we defined those paths with this property to be *augmenting paths*.

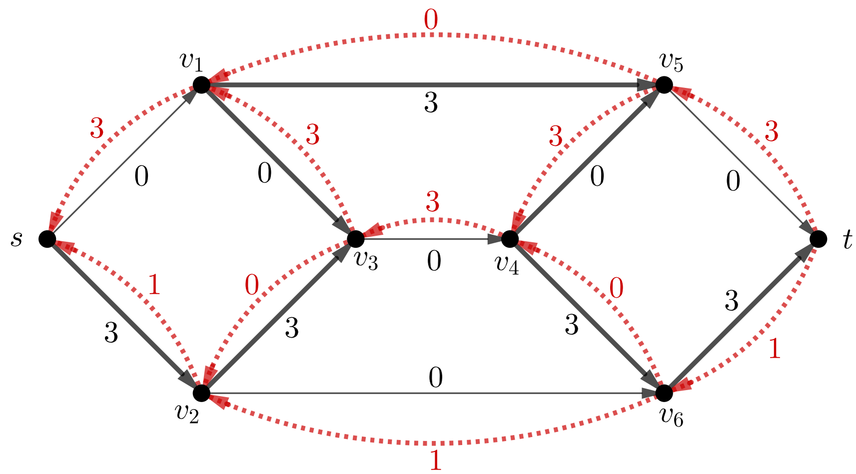


Figure 15: The residual network of the suboptimal graph above

Figure 15 also has a (bolded) valid augmenting path. We can see what this path will do to the flow along its edge in the next step of the algorithm by organizing its edges in a table:

Edge	(s,v_2)	(v_2,v_3)	(v_3,v_1)	(v_1,v_5)	(v_5,v_4)	(v_4,v_6)	(v_6,t)
ordered edge $\in D$?	Yes	Yes	No	Yes	No	Yes	Yes
flow adjustment	$+3$	$+3$	-3	$+3$	-3	$+3$	$+3$

Exercise questions – After each step (each new augmented path added) in the Ford-Fulkerson algorithm:

- Justify that the flow always increases.
- Justify the capacity constraint is still satisfied.
- Justify that conservation of flows is still satisfied.

Definition 10.1 (Cut)**.** *An s-t cut (S,T) is a partition of $v(D)$ such that $s \in S$ and $t \in T$. That is, an s-t cut is a division of the vertices of the network into two parts, with the source in one part and the sink in the other.*

For any partition S, T (vertex sets with $S \cap T = \emptyset$ and $S \cup T = v(D)$), we use the following notation for the cut set (see Figure 16; there may be edges from T to S):

$$X(S,T) := \{(u,v) \in e(D) \ : \ u \in S, v \in T\}.$$

If (S,T) is an s-t cut, $X(S,T)$ is the set of edges that connect the source part of the cut to the sink part. Thus, if all the edges in $X(S,T)$ are removed, no positive flow is possible, because there is no path in the resulting graph from the source to the sink. Define

$$C(S,T) = \sum_{(u,v)\in X(S,T)} c(u,v).$$

The "capacity" of an s-t cut is $C(S,T)$, the total capacity of edges from S to T.

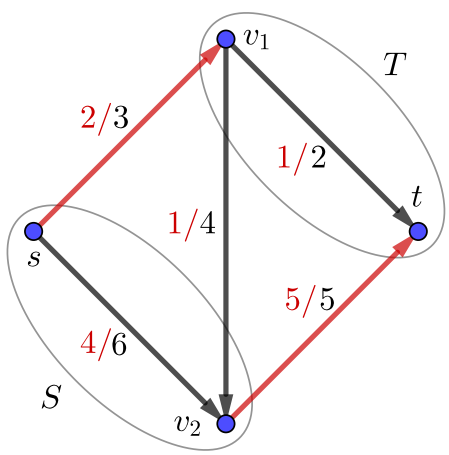
Problem 10.2 (Min Cut)**.** *Minimize $C(S,T)$, that is, determine S and T such that the capacity of the s-t cut is minimal.*

Even though min-cut doesn't mention anything about flows directly, these problems are closely related.

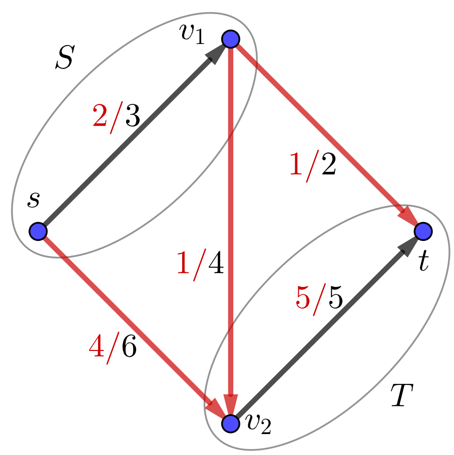
Theorem 10.3 (Max-flow min-cut)**.** *For any network, the solutions to max flow and min cut are the same value.*

Towards this, let's consider a cut on a network that has a flow, and we'll prove two lemmas to help in the proof. Let

$$F(S,T) = \sum_{(u,v)\in X(S,T)} f(u,v).$$



$X(S,T) = \{(s,v_1),(v_2,t)\}$
$C(S,T) = 8$
$F(S,T) = 7$



$X(S,T) = \{(s,v_2),(v_1,v_2),(v_1,t)\}$
$C(S,T) = 12$
$F(S,T) = 6$

Figure 16: Example of two s-t cuts on a network, with cut capacities and flows

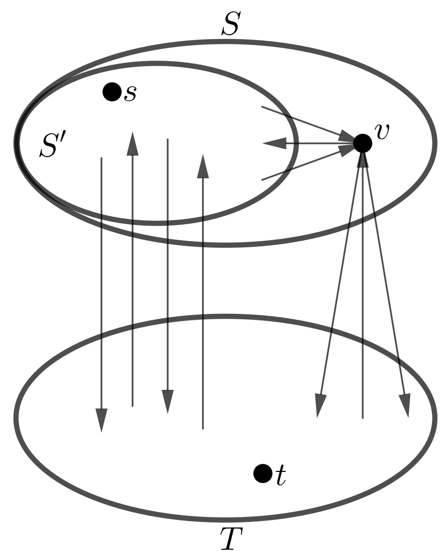
Lemma 10.4. *For a network $N(D,s,t,c)$, flow f, and an s-t cut (S,T), we have $|f| = F(S,T) - F(T,S)$.*

Proof. By induction on the size of S. If $|S| = 1$, the source is the only vertex in S. Then

$$F(S,T) - F(T,S) = F(S,T) = \sum_{(u,v)\in X(\{s\},D-\{s\}} f(u,v) = \sum_{v\in N_{out}(s)} f(s,v) = |f|$$

by proposition 9.9.

Next, suppose it's true for all $|S| < k$, and now let $|S| = k$. Choose any vertex $v \in S$ other than the source, and let $S' = S - \{v\}$.



From the induction hypothesis, we have:

$$|f| = F(S', T + \{v\}) - F(T + \{v\}, S') = (F(S',T) + F(S',v)) - (F(T,S') + F(v,S')) . \quad (10.1)$$

From conservation of flows,

$$0 = F(S' + T, v) - F(v, S' + T) = (F(S',v) + F(T,v)) - (F(v,S') + F(v,T)) , \quad (10.2)$$

and subtracting equation (10.2) from (10.1), what what we get is

$$|f| = (F(S',T) + F(v,T)) - (F(T,S') + F(T,v)) = F(S,T) - F(T,S).$$

□

In particular, a useful corollary follows.

Lemma 10.5. *For a network $N(D,s,t,c)$, flow f, and a cut (S,T), we have $|f| \le C(S,T)$ with equality iff $F(T,S) = 0$ and $f(u,v) = c(u,v)$ for each $(u,v) \in X(S,T)$.*

Proof. By Lemma 10.4,

$$|f| = F(S,T) - F(T,S) \le F(S,T) \le C(S,T).$$

Equality only holds if and only if both of those inequalities become equality, which is equivalent to the conditions that $F(T,S) = 0$ and $f(u,v) = c(u,v)$ for each $(u,v) \in X(S,T)$. □

Lecture 11 (April 8th)

Proof of 10.3. Let f be a flow on a network. We prove theorem 10.3 by showing the following statements are equivalent:

1. f is a maximum flow.

2. There is an s-t cut with $C(S,T) = |f|$.

3. There are no augmenting paths left in the network N with flow f.

2. $\implies$ 1.: From Lem 10.5, all flows have $|f| \le C(S,T)$, so a flow with $|f| = C(S,T)$ is max.

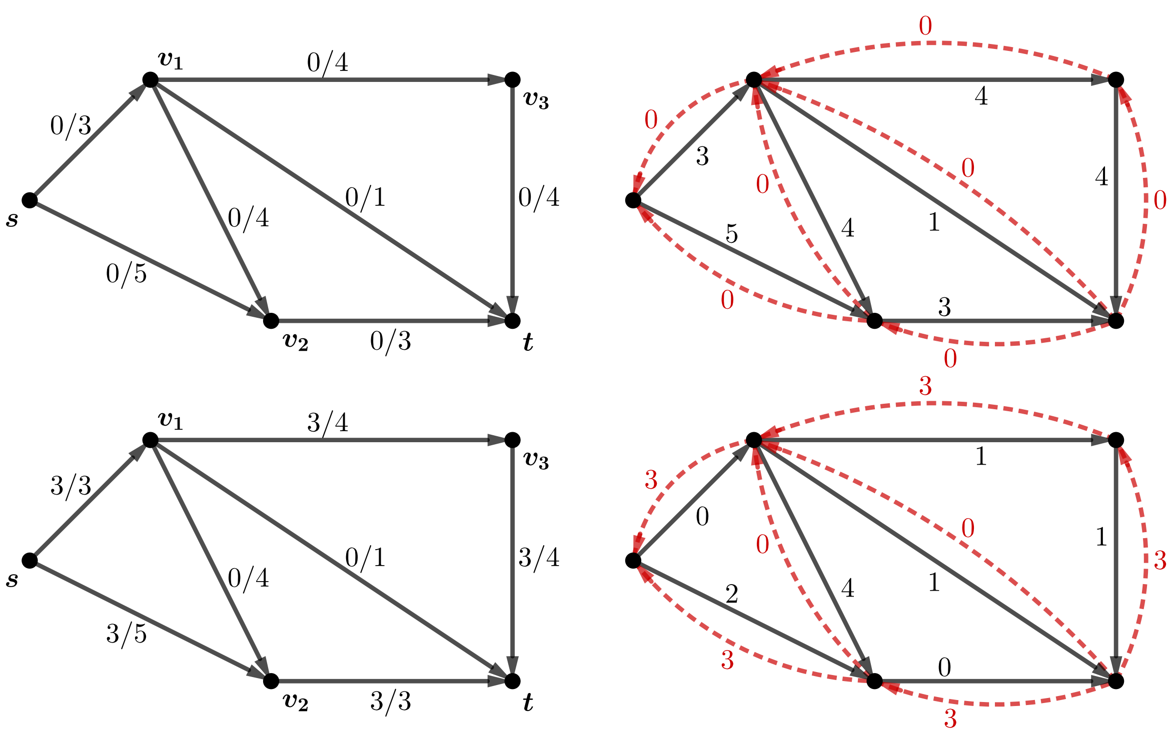
1. $\implies$ 3.: If there was an augmenting path, we could use the algorithm to increase the flow.

3. $\implies$ 2.: There is no augmenting path from s to t, but let S be the set of vertices "reachable" from s, i.e., the vertices v where an augmenting path from s to v exists. Let T be all the other vertices. By definition of S, it's not possible that

- $F(T,S) \neq 0$, otherwise there would be a reverse edge that we could augment further along, or

- $f(u,v) \neq c(u,v)$ for an edge in $X(S,T)$, otherwise we could just straightforwardly extend an augmenting path along that edge.

$\square$



How can we be certain this cannot be improved? We find a min cut! $S = \{s, v_2\}$ is a cut set where $F(T,S) = 0$ and $f(u,v) = c(u,v)$ for each $(u,v) \in X(S,T)$. Therefore, $|f|$ is the max flow, by the (2) to (1) part of the proof of the theorem.