Thursday, October 13

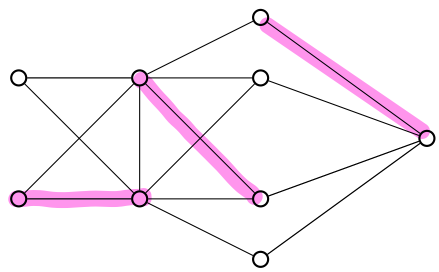
1. Welcome!
2. Thank you for topic ideas
   - ↳ if you didn't submit one, email me.
3. hw 5 is up, due before break
   - ↳ 1,2: Kruskals-y
   - ↳ 3,4: shortest path
   - ↳ 5: matching proof. (hint: check degree!)
4. questions?
5. shortest paths
6. matching
7. small work → none, get on the hw!

Math 494: Discrete Optimization 10/13

Another day of DiscOp, another graph problem to consider! This one needs a bit more theoretical setup, so we'll start by digging into some proofs before we build programs and algorithms. Our next few days are about *matchings*, which have applications all over the place, particularly with bipartite graphs. Let's get into it with a definition.

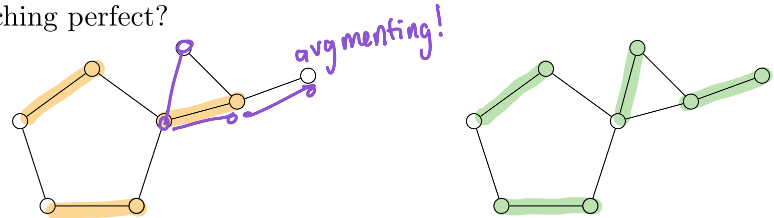
Definition (independent, matching) Two edges are *independent* if they do not share vertices. A *matching* is a set of independent edges.

Example: Find a matching of the graph below. Try to make your matching as big as you can!



Definition (maximal, maximum, perfect, saturate) A matching is *maximal* if it is not a subset of another matching, *maximum* if no other matching has more edges, and *perfect* if it contains every vertex of G. We'll say a perfect matching *saturates* G.

Example: For the graph below find a matching that's *maximal* and *maximum*. Is your maximum matching perfect?



Question: What sorts of questions can we ask about matchings?

how many maximum matchings?

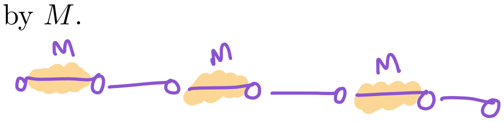
how many maximal?

minimize edges in maximal matching?

if edges have cost: min, max?

We're going to talk about several ways to make matchings in graphs, but one way will consider *augmenting paths.*

Definition: (alternating path, augmenting path) An M-*alternating* path is a path that alternates between edges in M and edges not in M. Such a path is an M-*augmenting* path if the endpoints are not saturated by M.

alternating: 

augmenting: 

Claim. *A matching is maximum if and only if it has no augmenting path.*

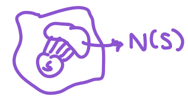
Intuitively...

Sure! if aug. path, we could make matching better. If no aug. path, can we be sure it's max?

Proof: Homework 5 Question 5.

Another condition we'll use is the idea of being "neighborly", particularly with bipartite graphs.

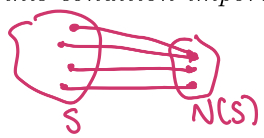
Definition: (neighborhood, neighborly, deficient) For a set of vertices $S \subseteq V$, the *neighborhood* of S, denoted $N(S)$, is the set of all other vertices that are adjacent to something in S. In set builder notation,



$$N(S) = \{v \in V \setminus S : v \sim s \text{ for some } s \in S\}$$

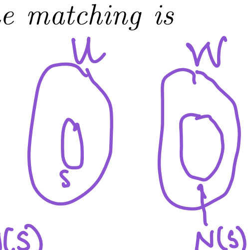
A set S is *neighborly*, for every $U \subseteq S$, $|N(U)| \geq |U|$. (Your book will call the opposite of neighborly *deficient*).

Why is this condition important?



if S is not neighborly, we can't find a matching that saturates S (uses every vertex from S)

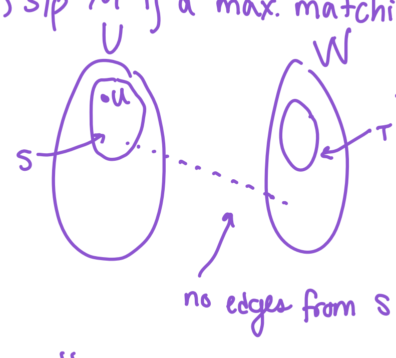
Theorem (3.12, Hall's)**.** *Let $G = (U \cup W, E)$ be a bipartite graph where $|U| \leq |W|$. Then G has a U saturating* matching *if and only if every $S \subseteq U$ is neighborly. (Note, if $|U| = |W|$, the matching is perfect).*



Proof:

($\Rightarrow$) Suppose G has a U saturating matching. Consider $S \subseteq U$. Note $|N(S)| \geq |S|$ because for every $s \in S$, $\exists!\, v \in N(S)$, (take the matched edge). Thus every s/set of U is neighborly.

($\Leftarrow$) s/p M is a max. matching that does not saturate U.



let $u \in U$ be unsaturated, and let S, T be the vertices that are reachable from u with alt. paths.

no edges from S to $W \setminus T$., because this builds an aug. path!

this means $N(S) = T$; but! $|S| = |T| + 1$ ← and u. so $|S| > |N(S)|$

each non-u vertex is matched w/ something in T

this contradicts S being neighborly.

So now that we have a handle on the graph theory behind matchings, let's get into the optimization. We'll consider two different types of matching problems (but there are way more that we could explore, or you could explore in a project).

- Finding the maximum matching in any graph.
- For weighted bipartite graphs with perfect matchings, finding the minimum weight perfect matching.

Let's begin by building linear programs for both.

Example: Write and dualize linear programs for both of these questions. Do the duals have interpretation?

Definition: (vertex cover) A set of vertices $W \subseteq V$ is a *vertex cover* if every edge of E is incident to one of the vertices of W.

Somehow these are related...? Let's talk it over next time! We'll also get into matching algorithms for both these questions.