Math 4990: Matchings

11/24

2nd half of Ch. 11 of Bóna

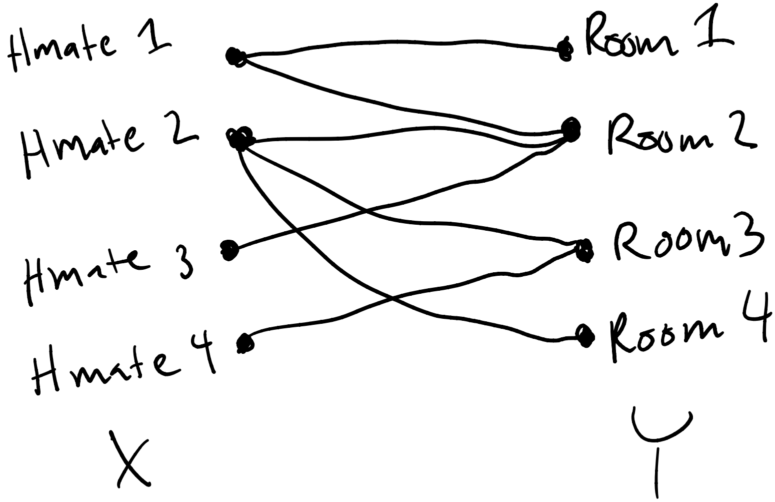
Reminders: • Midterm #2 will be graded + returned soon, if not already.

• HW #5 (the last one!) has been posted, is due in a week on 12/1.

Consider the following "real world problem": a group of people are moving into a house together and they need to decide how to allocate rooms to the housemates. Each housemate has certain rooms they would consider acceptable to live in, and other rooms not acceptable.

Q: How can we allocate rooms to housemates so that every housemate gets an acceptable room?

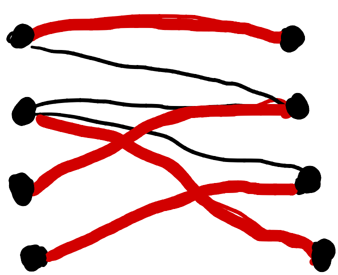
It's helpful to represent the information of which housemates find which rooms acceptable in the form of a bipartite graph:



We have a set X of vertices representing the housemates, a set Y representing the rooms, and an edge from $x \in X$ to $y \in Y$ means housemate x finds room y acceptable.

What is a valid assignment of housemates to rooms in this graph theory language?

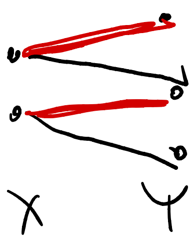
It's a subset of edges with each vertex contained in exactly one edge of the subset:



Def'n A matching in a graph G is a subset of edges s.t. every vertex is in at most one of those edges. It is a perfect matching if every vertex is in exactly one of the edges.

We'll focus on matchings in bipartite graphs. From now on today let G be a bipartite graph w/ bipartition (X, Y).

Def'n A perfect matching of X into Y is a matching that includes every vertex of X (but not necessarily of Y); e.g.,



Maybe have more rooms than housemates, but that's ok as long as every hmate gets a room.

Main Q: When does a perfect matching of X into Y exist? And how to find it?

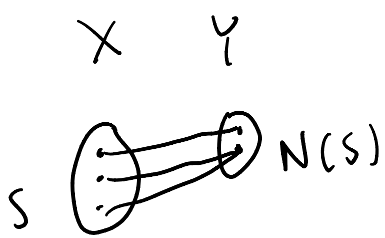
Observation: Definitely need $|Y| \geq |X|$, i.e., there has to be at least as many rooms as housemates. Similarly each $x \in X$ has to be adjacent to atleast one $y \in Y$, i.e. every housemate has to find some room acceptable.

Continuing this reasoning leads to...

Def'n For a subset S of vertices, its neighborhood, $N_G(S)$ or $N(S)$, is the set of all vertices adjacent to some $s \in S$.

Prop. If a perfect matching X into Y exists, then $|N(S)| \geq |S|$ $\forall S \subseteq X$.

Pf: If there's some $S \subseteq X$ w/ $|S| > |N(S)|$, then:



No hope to match all the things in S!

$\boxtimes$

The surprising fact is that the converse of this proposition is also true:

Thm ("Hall's Marriage Theorem")

$\exists$ a perfect matching of X into Y
$\iff \forall S \subseteq X, \ |N(S)| \geq |S|$.

Actually, we can say a bit more. Let's call a matching with the most edges among any matching a **maximum matching**.

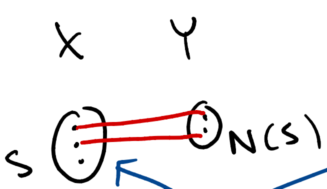
Thm If M is a maximum matching then

$$\#\text{unmatched vertices } x \in X \text{ in } M = \max_{S \subseteq X} |S| - |N(S)|.$$

Pf of easy direction:

$\forall S \subseteq X$, $\#$ unmatched vertices $x \in X$ in $M \geq |S| - |N(S)|$

since



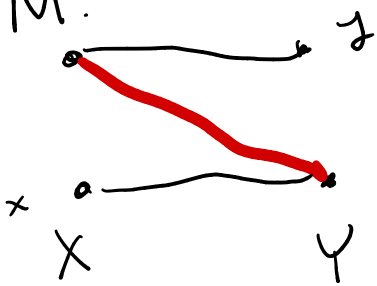
Can match at most $|N(S)|$ of these, so $|S| - |N(S)|$ go unmatched. $\square$

What about the hard direction? Lets not just prove it, but also give an algorithm which finds a maximum matching.

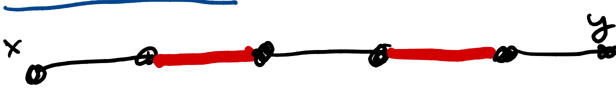
Idea behind algorithm: start w/ any matching, and if it's not maximum, augment it until it is.

What do we mean by 'augment'?

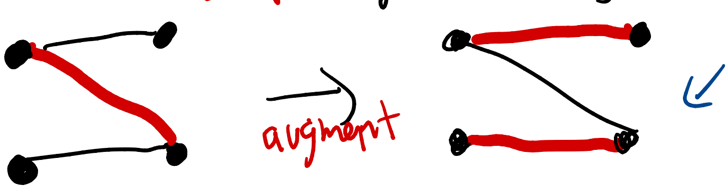
Consider M:



We can find path from $x \in X$ to $y \in Y$ s.t. x, y both unmatched in M, and path alternates



between non-edges + edges of M. Call this an augmenting path. Can flip edges along augment. path:



bigger matching!

So the way our algorithm will work is:

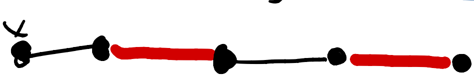
- We repeatedly augment along augmenting paths as long as we can;
- We stop when we have no augmenting paths.

Thm Let M be a matching. Then:

a) If M has an augmenting path, then we can augment along it to get a matching M' w/ more edges.

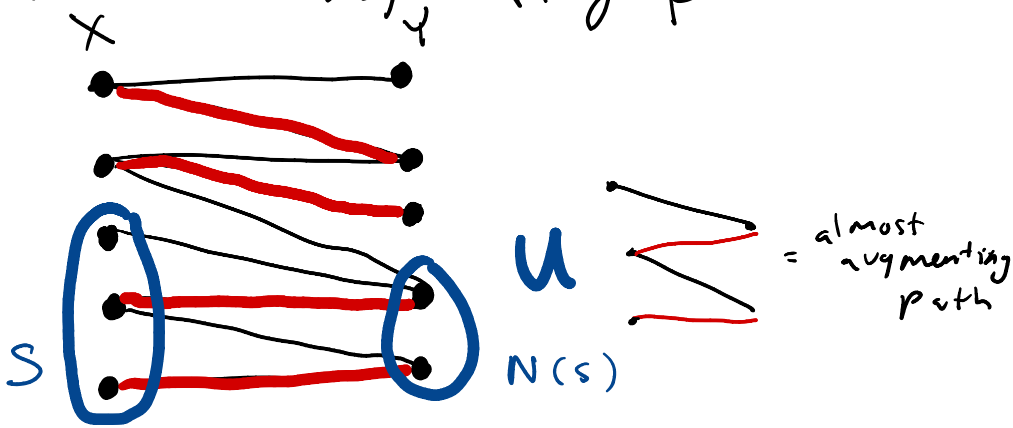
b) If M has no augmenting paths, then $\exists\, S \subseteq X$ s.t. $\#\,\substack{\text{unmatched} \\ \text{vertices } x \in X \\ \text{in } M} = |S| - |N(S)|$, which means M is a maximum matching.

pf: a): we have already explained.

b): Suppose M has no augmenting paths. Let's call a path P an almost augmenting path if it:
- starts at any unmatched $x \in X$,
- alternates  between non-edges and edges in M.

Consider set $U :=$ all vertices reachable by a an almost augmenting path.

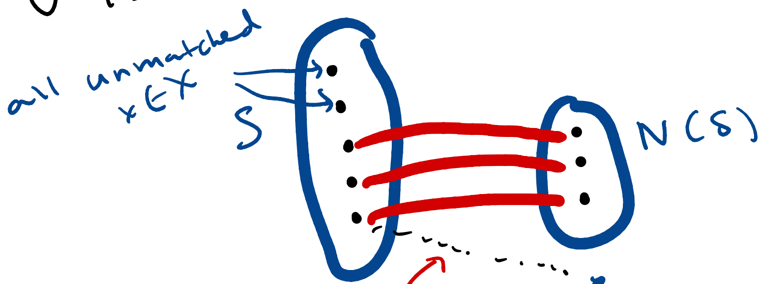
e.g.



Let $S := U \cap X$. <u>Claim</u>: $N(S) = U \cap Y$,

and consists of $y \in Y$ s.t. y matched to some $x \in S$.

i.e., U looks like:



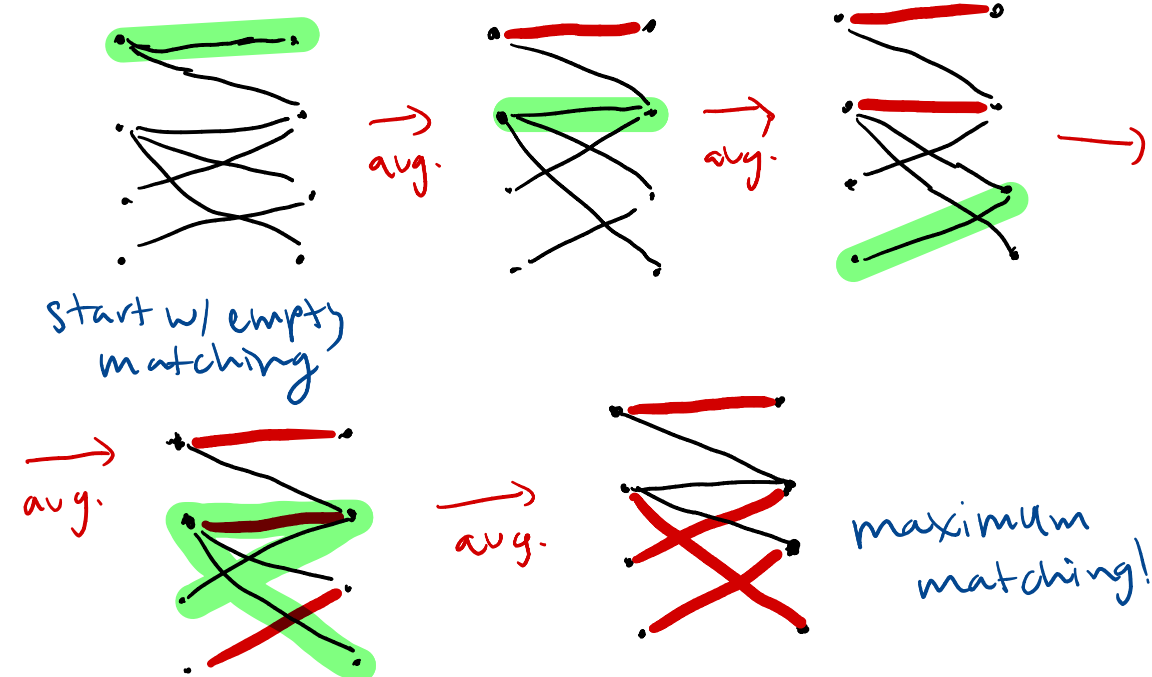
<u>Otherwise</u>: could extend almost augmenting path to a full augmenting path, but we assumed we didn't have any of these. So indeed for this S we have

$\#$ unmatched $x \in X$ in $M = |S| - |N(S)|$,

and since $\#$unmatched $\geq \max(|S| - |N(S)|)$,

this means our matching is maximum ☒

Example of augmentation algorithm:



<u>Remark</u>: this algorithm is a special case of the Ford-Fulkerson algorithm for finding a maximum flow in a network with edge capacities.

Now let's take a break...

and when we come back let's work on Matching problems on the worksheet in breakout groups!