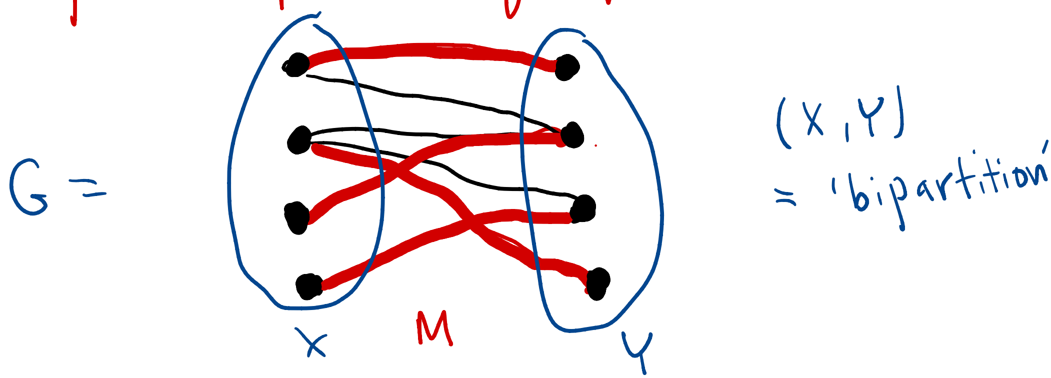
Math 4707: Finding a matching

3/15

Ch. 10 of LPV

Reminder: • HW #4 (Ch.'s 9+10) has been posted, due Wed. 3/24.

Recall that last class we were discussing matchings in bipartite graphs:



We wanted to find an X-saturating matching (one covering all vert's in X), or in general the biggest matching we could. This is a useful way to think about assignment problems.

We stated surprisingly simple nec.+suf. condition for X-saturating matching:

Thm ("Hall's Marriage Theorem")

$\exists$ an X-saturating matching in G

$\iff \forall A \subseteq X, \ \#N_G(A) \geq \#A$.

neighborhood

And actually we said more: call a matching w/ maximum # of edges a **maximum matching**.

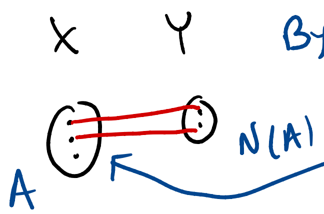
Thm If M is a maximum matching then

$$\#\text{unmatched vertices } x \in X \text{ in } M = \max_{A \subseteq X} \#A - \#N_G(A)$$

Pf of easy direction:

$\forall A \subseteq X$, $\#$ unmatched vertices $x \in X$ in $M \geq \#A - \#N_G(A)$

since



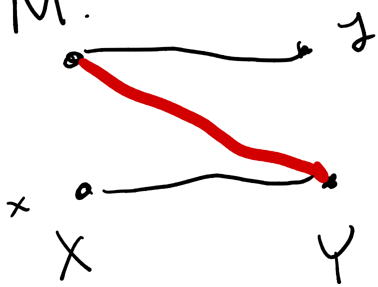
By pigeonhole, can only match $\#N(A)$ of these, so $\#A - \#N(A)$ go unmatched. □

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.

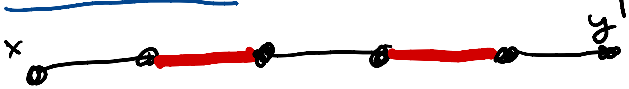
What do we mean by 'augment'?

Consider M:



← maximal but not maximum

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.

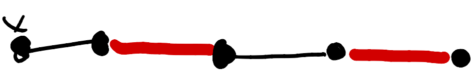
Thm Let M be a matching. Then:

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 $\exists A \subseteq X$ s.t. $\#$ unmatched vertices $x \in X$ in M $= \#A - \#N_G(A)$, which means M is a maximum matching.

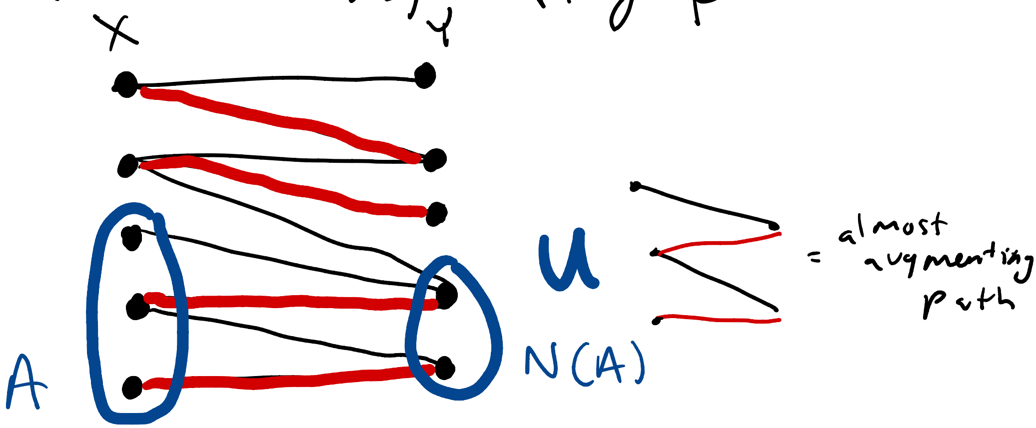
pf: a): we have already explained.

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.

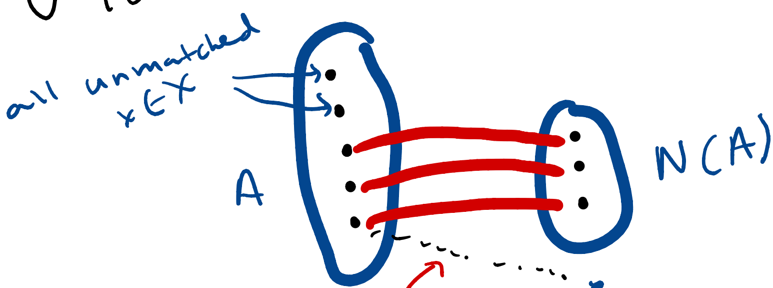
e.g.



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

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

i.e., U looks like:



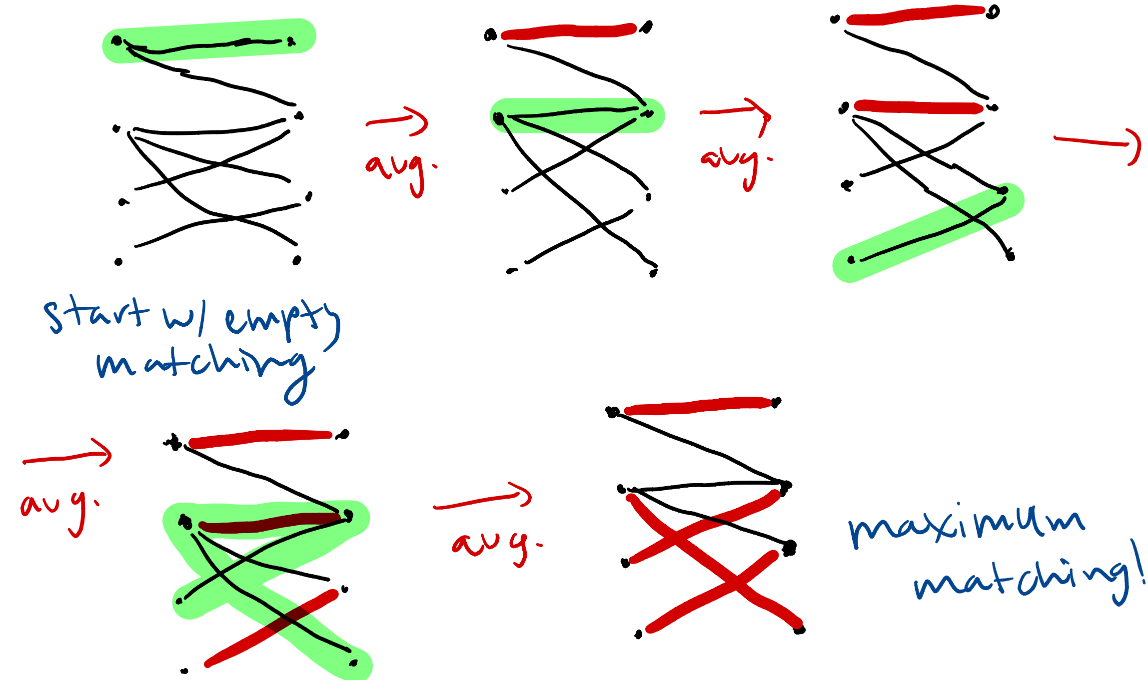
<u>Otherwise</u>: could extend almost augmenting path to a full augmenting path, but we assumed we didn't have any of these. So indeed for this A we have

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

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

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.

We will discuss this next class...

One last thing to think abt.:

How do you know you're done in the maximum matching alg.?

After we take a 5 min break, we can practice finding a max. matching on today's worksheet in breakout groups...