Lecture 9 - Bipartite Matching

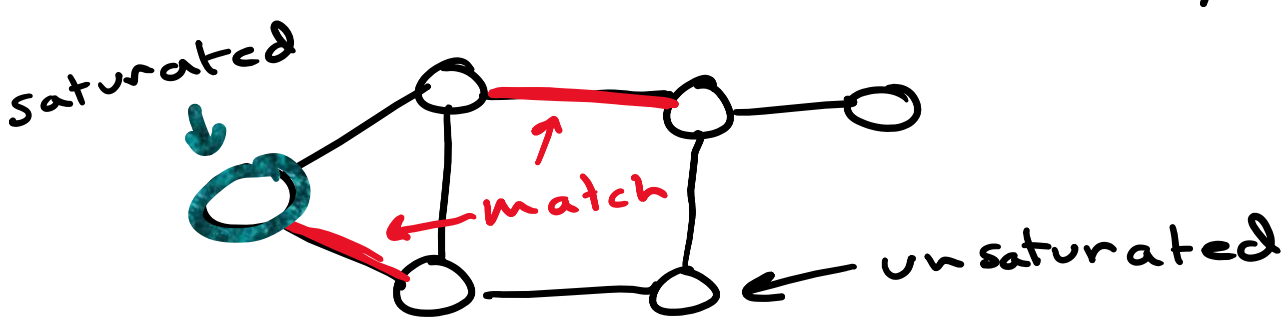
Monday, February 16, 2026 4:11 PM

Flights are becoming weirdly expensive, why?

- Same routes, same days, 50-100% more for certain routes.
- Gas prices low. Winter season. Economic uncertainty => cheaper flights
- Possible bias for Slota's routes?
- Airlines reducing flights -> lower supply, fixed demand => higher prices?
- Increased fixed costs (e.g., maintenance) due to tariffs?
- Price fixing? Failure of regulatory state?
- Other possibilities?

Matching

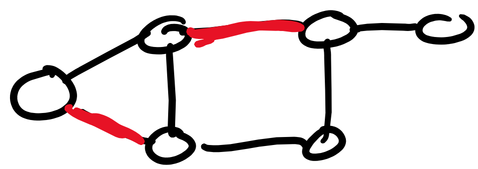
<u>**Match**</u> on G = a set of non-loop unique edges with no shared endpoints



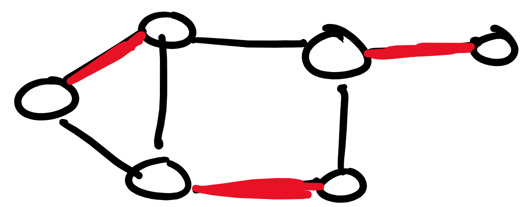
saturated vertex - vertex with an inciden matched edge

unsaturated vertex - vertex with no incident matched edge maximum match - largest match in terms of cardinality

maximal match - a match that can't be made larger



maximal



maximum / perfect

perfect match - a match that saturates all vertices

$$|M_p| = \frac{|V(G)|}{2}$$

perfect match

$M_p = \{e_i \ldots e_j\}$ unique

$e_k \in E(G)$

Does G have a perfect match?

Necessary Conditions:

→ $|V(G)|$ must be even

→ $\forall v \in V(G): d(v) \geq 1$

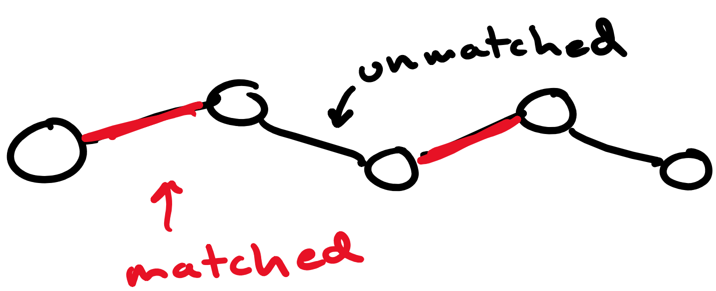
Sufficient Conditions:

→ Lot tougher to say

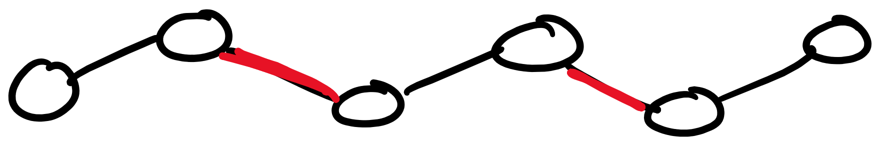
↳ $|\text{maximum match}| = \frac{|V|}{2}$

Alternating paths

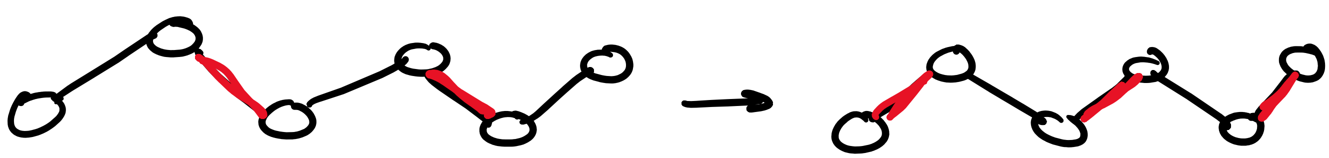
M-alternating path - given match M on graph G, an M-alt. path is a path subgraph that alternates between matched and unmatched edges



M-augmenting paths - an M-alt. path that starts and ends on path that starts and ends on unsaturated vertices



Note: along an M-aug. path, we can increase the size of M by swapping matched and unmatched edges along the M-aug. path



=> this implies that if G has an M-aug. path, then M is not maximum

Q: Does every non-maximum match have an M-aug. path?

have an M-aug. path?

A: yes → Berge

Berge: a match M on G is maximum iff there exists no M-aug. paths

Contrapositive

$P \iff Q \quad \neg P \iff \neg Q$

M is not maximum $\iff \exists$ M-aug. path

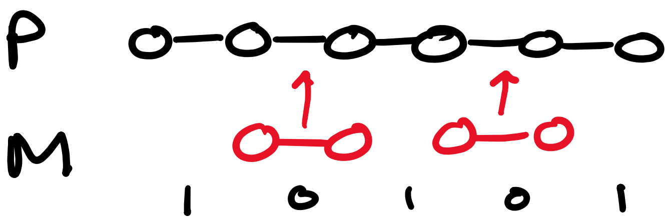
($\Leftarrow$) we already showed how to use an M-aug. path to increase the size of a match by 1

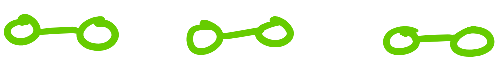
Note: To do so, we are taking a <u>symmetric difference</u>

of path P and match M

an exclusive OR → XOR

A	B	XOR
1	1	0
0	0	0
1	0	1
0	1	1



P△M  = M', our new larger match

(P⊗M)

(⇒) Consider M as our current match and M' as a larger match where $|M|<|M'|$

Consider $F = M' \triangle M$

even cycles

odd cycles? NO

even paths

odd paths

Consider odd paths in F

Consider odd paths in F

→ as $|M'| > |M|$ and the number of edges w.r.t. M and M' are equal in even cycles & paths

=> at least one odd path (M-aug. path) must exist □

Maximum Bipartite Matching

↳ maximum matching on bipartite graphs

We consider optimality with respect to $|X|$ w.l.o.g. where $|X| \le |Y|$ in $G_{X,Y}$ bipartite graph

X-saturating match

↳ match that saturates all $x \in X$

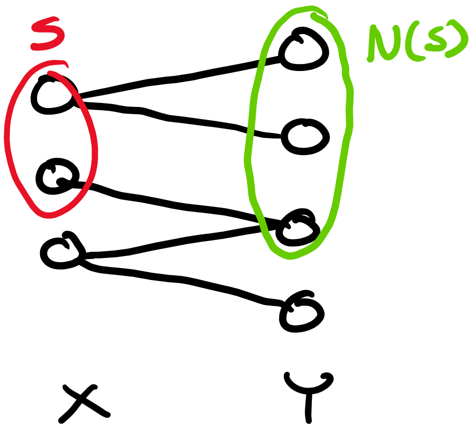
↳ obviously maximum

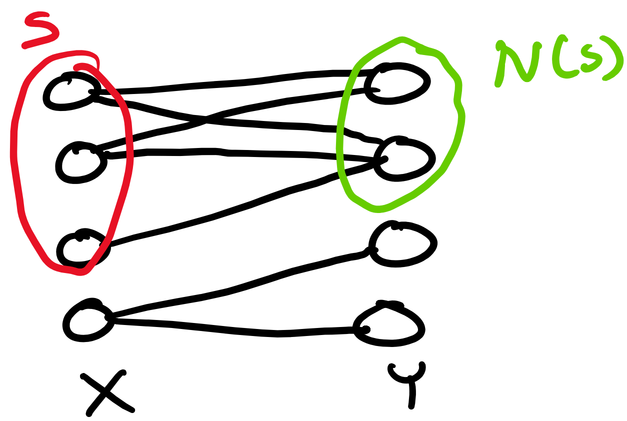
Hall: $\exists M$ that saturates X in bigraph $G_{X,Y}$ with $|X| \le |Y|$

iff $\forall S \subseteq X : |N(S)| \ge |S|$

subsets

neighborhoods of all of S





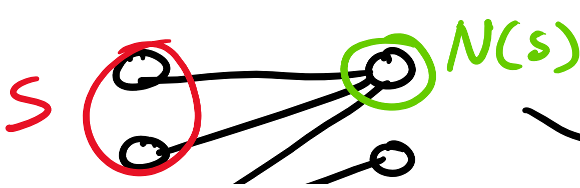
Contrapositive

No X-saturating M

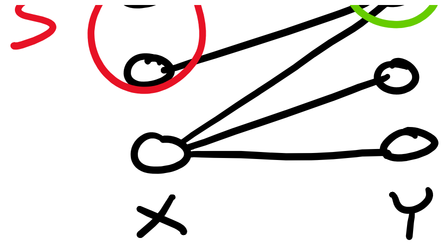
iff

$\exists S \subseteq X : |S| > |N(S)|$

($\Leftarrow$) we demonstrated it above



→ there are not



→ there are not enough 'partners' in Y for S to match with

(⇒) Consider some maximum M

Consider some $u \in X$, $u \notin V(M)$

unsaturated

Consider S = all $v \in X$ that are reachable via a u,v-M-alt. path

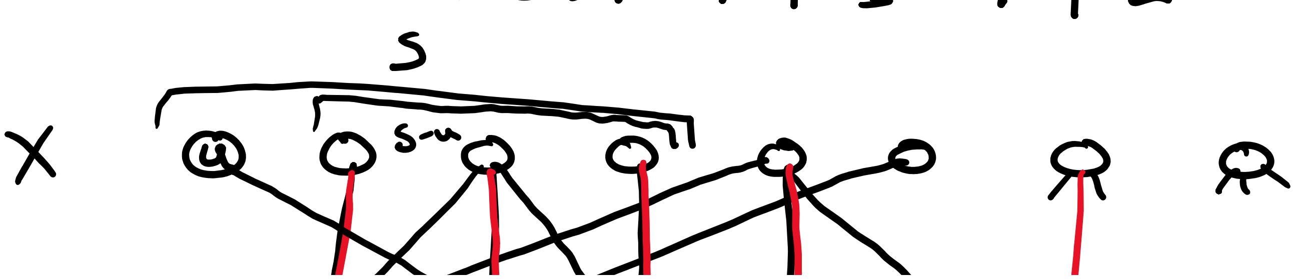
Note: $N(S)$ is fully saturated

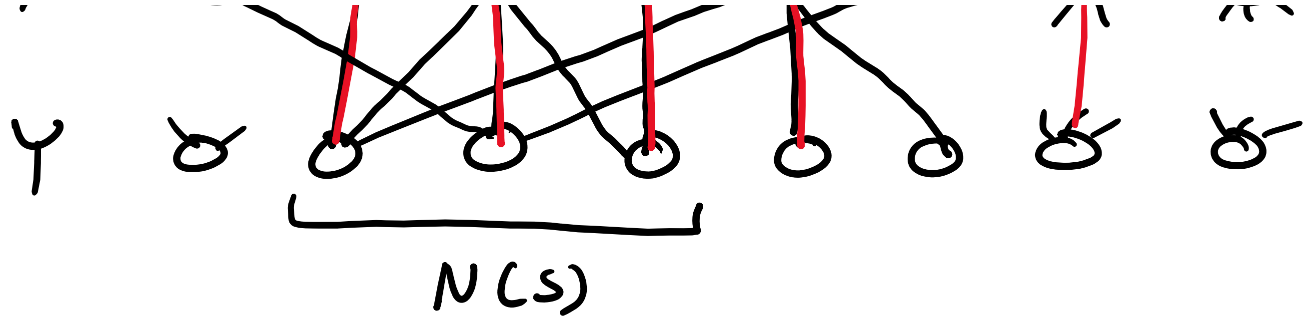
↳ we are traversing an M-alt. path

⇒ otherwise we have an M-aug. path contradiction

Finally: consider a bijection from $N(S) \leftrightarrow S - u$, we have

$$|N(S)| = |S| - 1 < |S| \quad \square$$





Q: How can we find a maximum match on a bipartite graph?

↳ A: Find M-augmenting paths

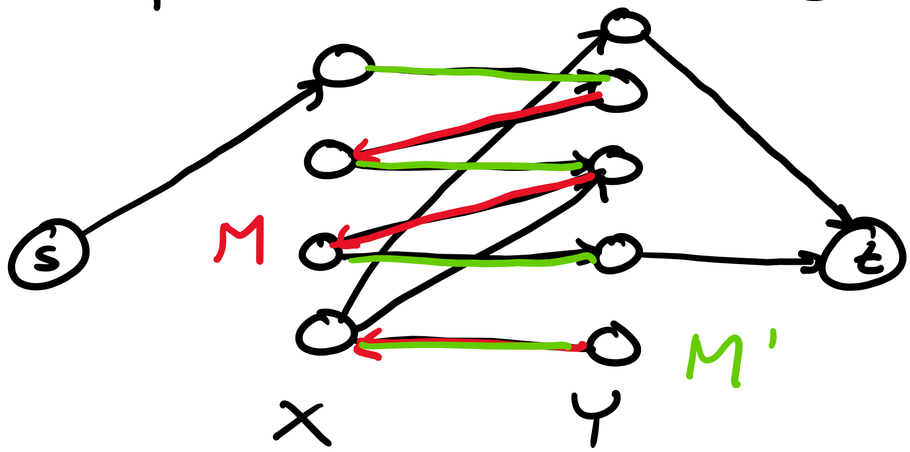
Maximum bipartite matching ALGO.

↳ Iteratively find M-aug. paths to increase M

To do so:

- add vertex s with directed edges to all unsaturated $u \in X$
- add vertex t with directed edges from all unsaturated $v \in Y$
- direct all matched (u, v) edges $u \in X, v \in Y$ to point from $v \to u$
- direct all unmatched (u, v) to

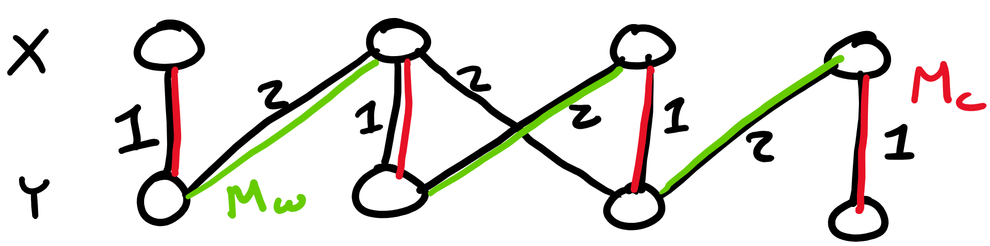
- direct all unmatched (u,v) to point from $u \to v$



Mo' matchin'

Maximum weight matching

↳ we optimize the sum of edge weights in our match instead of just $|M|$ cardinality



$|M_c| = 4$ $\qquad$ $|M_w| = 3$

$\underbrace{\sum_{e \in M_c} w(e) = 4}$ $\qquad$ $\underbrace{\sum_{e \in M_w} w(e) = 6}$

$$\underbrace{\qquad e \in M_c \qquad}_{\text{max cardinality}} \qquad \underbrace{\qquad e \in M_w \qquad}_{\text{max weight}}$$

Note: max cardinality and max weight matchings are separate (sp?)

stable matching

↳ optmize a match for given preferences

Classic "Marriage Problem"

- n men and n women
- each specifics a preference order of potential partners

Goal: create a stable match

unstable pair: man x and woman y are not matched but both prefer match (x, y) over their current partners

stable match: a match with

<u>stable match</u>: a match with no unstable pairs

Gale-Shapley proposal ALGO:

Input: preferences for all n men and n women

while not done:

  Each man proposes to their highest preference partner that has not yet rejected them

  If each woman gets exactly 1 proposal → done

  Else women with 2+ proposals reject all but their highest preference

Men (x, y, z, w)	Women (a, b, c, d)
x: a > b > c > d	a: z > x > y > w
y: ~~a~~ > ~~c~~ > b > d	b: y > w > x > z
z: ~~c~~ > d > a > b	c: w > x > y > z
w: c > b > a > d	d: x > y > z > w

Iterate:

$(x, a)\ (y, a)\ (z, c)\ (w, c)$

reject reject

$(x, a)\ (y, c)\ (z, d)\ (w, c)$

reject

$(x, a)\ (y, b)\ (z, d)\ (w, c)$ ✓