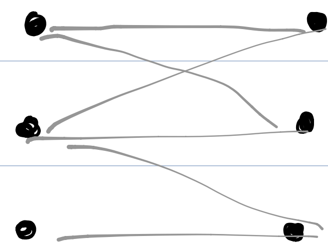
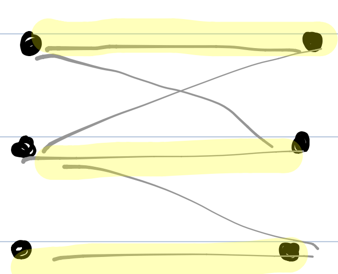
Maximum weight matching

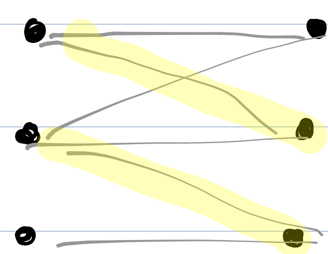
Bipartite graphs very useful for modeling:

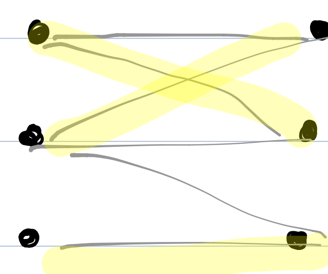
- jobs & machines
- employers, employees
- men women



Matching: set of edges with no common endpts

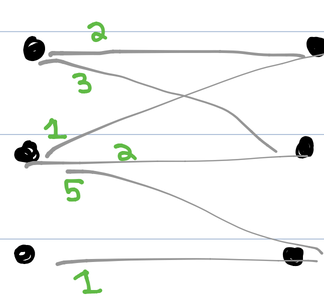






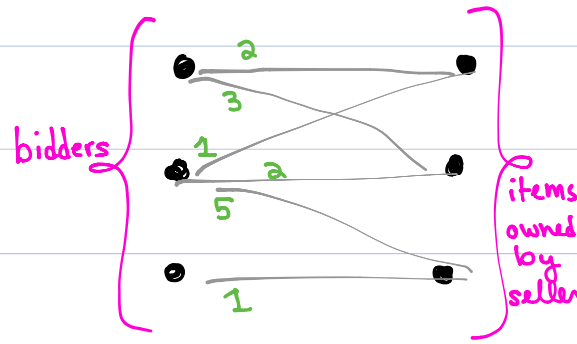
Total weight 5 8 5

Often ∃ weights on edges and we want to find max weighted matching



Ascending auction alg for max matching for integer weights

Fix bid increment $\delta = \frac{1}{n+1}$



$\vec{p} = (p_1, \dots, p_n)$ $\quad p_j$ price of item j

Initially all $p_j = 0$ and matching empty $M(i) = \emptyset \ \forall i$

As long as matching not perfect

one unmatched bidder i

v_{ij} = value of bidder i for item j

bids on some j in

$\underbrace{D_i(\vec{p})}_{\text{demand set.}} = \{ j \mid v_{ij} - p_j \geq v_{ik} - p_k \text{ and } v_{ij} \geq p_j \}$

and bids $p_j + \delta$ on it.

If j unmatched, then $M(i) := j$

else, say $M(\ell) = j$,

remove (ℓ, j) from matching

and add (i, j) [i.e. $M(i) := j$]

Theorem

Suppose v_{ij} integer. The "auction algorithm" terminates with a max wt matching and the final prices are almost "envy-free"

$$M(i)=j \Rightarrow \quad v_{ij}-p_j \geq v_{ik}-p_k-\delta \quad \forall k \qquad (*)$$

Proof

- from moment item is matched, it stays matched forever
- until item is matched, $p_j=0$

$\Rightarrow$ if i unmatched, $D_i(\vec{p}) \neq \emptyset$

Every step till termination $\Delta\left(\sum p_j\right)=\delta$

$$p_k \leq \max_{i,j} v_{ij} + \delta \Rightarrow \text{alg terminates}$$

$\Rightarrow$ matching perfect.

$(*)$ ✓

Let M^* be any other matching, M final matching output

by $(*)$ $\quad \sum_i \left(v_{iM(i)}-p_{M(i)}\right) \geq \sum_i \left(v_{iM^*(i)}-p_{M^*(i)}-\delta\right)$

$$\Rightarrow \sum_i v_{iM(i)} \geq \sum_i v_{iM^*(i)} - n\delta \qquad = \sum_i v_{iM^*(i)} - \frac{n}{n+1}$$

since wt of any p.m. is int $\Rightarrow$ M max wt matching

$\blacksquare$

How many times main loop executed? $\le \frac{n}{\delta} \max_{ij} v_{ij}$

$\max_{ij} v_{ij} = 1 \underset{\text{i.e. matching}}{\Longrightarrow} O(n^2)$ times.