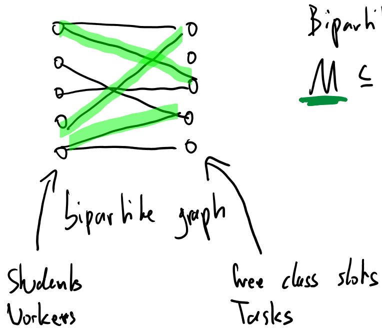
11 DP on tree & Bipartite Matching

Thursday, February 13, 2025 13:52



Bipartite Matching

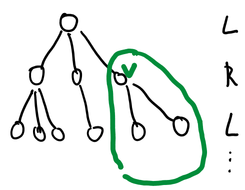
$\underline{M} \subseteq E$ such that each vertex has at most 1 edge from M incident to it.

bipartite graph represents 2 groups
edges represent qualification or interest
matching represents an assignment.

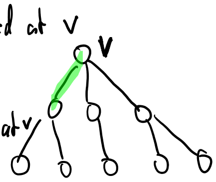
Def: Maximum Matching is a matching that is as large as possible

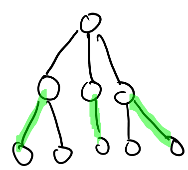
---

Setting G is a tree
(special case of bip. graph)



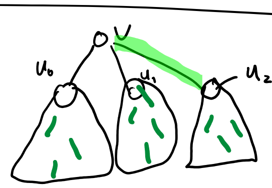
$M[v]$ = size of max matching in subtree rooted at v
if v is matched.





$U[v]$ = size of max matching in st. rooted at v
if v is not matched

$$U[v] = \sum_{u \text{ child of } v} \max(U[u], M[u])$$



$$M[v] = \sum_{u \text{ child of } v} \max(U[u], M[u]) + 1 + \max_{u \text{ child of } v} \Big(U[u] - \max(U[u], M[u])\Big)$$

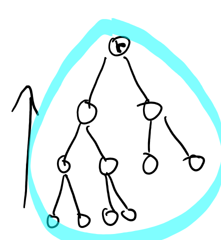
$U[\text{leaf}] = 0$

$M[\text{leaf}] = -\infty$

Algo 1 (Assume we know leaves and the children for each vertex)

Input (T)

for leaf v of T
$M[v] = -\infty$
$U[v] = 0$

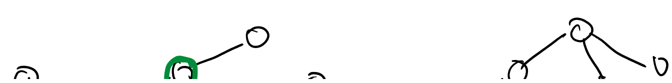


$\rightarrow$ for v in the tree, from bottom to top

$$U[v] = \sum_{\text{child } u \text{ of } v} \max(M[u], U[u])$$

$$M[v] = \underbrace{\sum_{\text{child } u \text{ of } v} \max(M[u], U[u])}_{U[v]} + 1 + \max_{\text{child } u \text{ of } v} \Big(U[u] - \max(U[u], M[u])\Big)$$

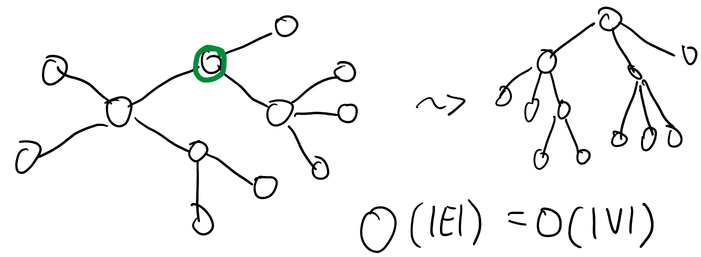
$$\Big[\sum_{v \in V} O(\deg(v)) = O(|E|) = O(|V|)$$
because tree



or v

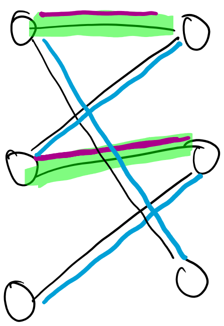
U[v]

return max(U[r], M[r]) // r is root



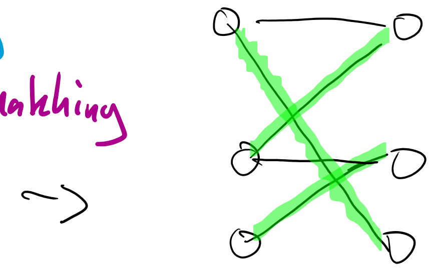
$O(|E|) = O(|V|)$

2:48



add to matching

remove from matching



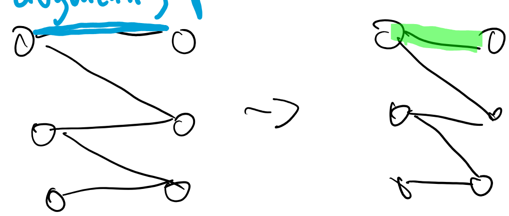
Def: Given G, M an alternating path is a path that alternates between edges in and outside the matching

Def: An augmenting path is a alternating path where both start and end point are unmatched

Given M and augmenting path, P we can "augment" M by the path by adding its unmatched edges and removing the matched edges

$M_{new} = M \triangle P$

augmenting path



G, $M = \emptyset$

Algorithm: start with a matching, eg $\emptyset$
while $\exists$ augmenting path P
$M \leftarrow M \triangle P$

Obs. this has at most $\frac{|V|}{2}$ iterations because any max matching has at most $\frac{|V|}{2}$ edges and each iteration increases size of matching by 1

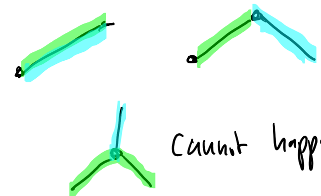
Obs. this has at most $\frac{n}{2}$ iterations because [illegible] edges

and each iteration increases size of matching by 1

Question: Is there always an augmenting path if we do not have a max matching yet?

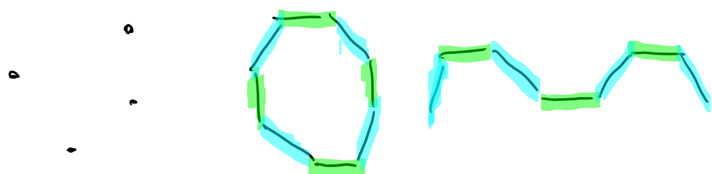
If M not maximum $\Rightarrow \exists$ augmenting path $\leftarrow$ to prove!

Let M be the current not max matching
Let N be a max matching, $M \neq N$



cannot happen

$M \triangle N$ $\leftarrow$ each vertex has degree 0, 1, 2



$|N| > |M|$

such a path with more edges from N must exist
$\Rightarrow$ is a augmenting path