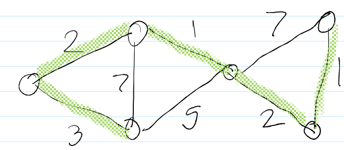
* HW2 - due Thurs Sep 11
* Exam 1 on Sep 18 : 4 problems : 1st pb short answer $\approx 35\%$

Recap : Graph Algorithms, BFS, Shortest path, (Priority) Queue

Today : MST, Matchings

Minimum Spanning Tree

$G = (V, E)$, $w : E \to \mathbb{R}_{\geq 0}$
undirected, connected

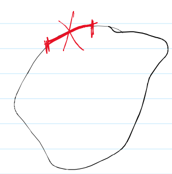


Find $T \subseteq E$ s.t.

(1) T is connected

(2) $\sum_{e \in T} w(e)$ is minimized

Obs 1: Optimal soln is a tree (there is no cycle)

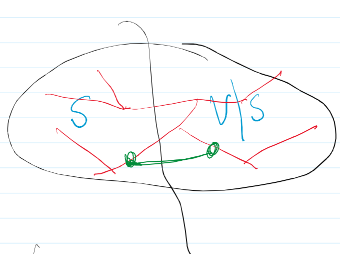


Thm : (Cut-property)

For any cut $(S, V \setminus S)$ where $S \subseteq V$ and $S \neq \emptyset$, $S \neq V$

Among edges $E(S, V \setminus S)$, the cheapest edge will be in the MST.

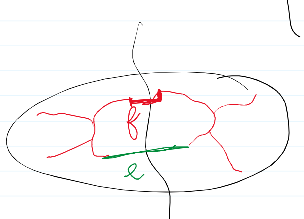
Pf:



Prove by contradiction: Suppose this is not true :

$T \leftarrow$ optimal MST

$e \leftarrow$ cheapest cross edge



Note $T \cup e$ contains cycle

Drop the most expensive edge in this cycle.

$\Rightarrow$ Cost goes down while maintaining connectivity

Consider $(T \setminus f) \cup e$

Consider $(T \setminus \{f\}) \cup e$ $\Rightarrow$ Cost goes down while maintaining connectivity ■

Prim's Algorithm

Thm: Finds MST in $O(m \log n)$ time.

Idea: (1) Expand MST starting at a root

(2) To expand, take the cheapest cut edge where $(\text{Red}, V \setminus \text{Red})$

Algo: Start at a root $r \in V$

Priority Queue Q = Add all edges to root with wts

$$\{((r,v), w_{rv})\}_{(r,v) \in E}$$

$T = \emptyset$, $\text{Visit} = \{r\}$

Repeat m times:

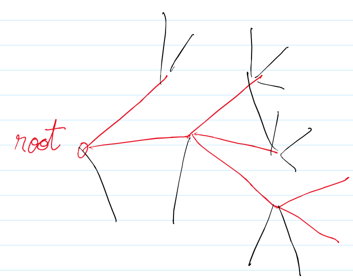
$(u,v) = Q.\text{out}$;

If $v \notin \text{Visit}$

$\text{Visit} = \text{Visit} \cup \{v\}$

$T = T \cup (u,v)$

$Q.\text{add}$ (edges $(v,w) \in E$ with wt w_{vw})



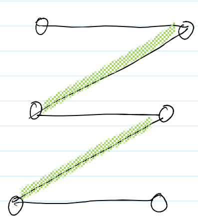
Recall in Shortest path $d_v + w_{vw}$

Correct because of cut-property: Each step we add cheapest cut edge

Time: $O(m \log n)$.

Matchings

$G = (V, E)$ unwtd, undirected



Matching M (but not maximum)

Find largest $M \subseteq E$ s.t. no vertex incident to more than 1 edge in M

i.e., $\forall u \in V$: $|\text{Edges}(u) \cap M| \leq 1$

no vertex incident to more than 1 edge in M

i.e, $\forall u \in V: \quad |Edges(u) \cap M| \leq 1$

Applications : 1) Courses to classrooms

2) Kidney exchange

3) Advertising / Dating / Ridesharing platforms

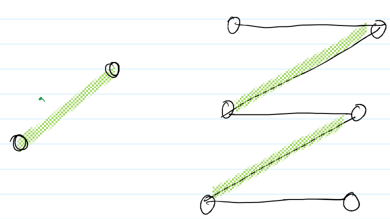
⋮

Q: How to find $\underbrace{\text{max-matching}}_{\text{wtd matchings also studied}}$ in poly-time?

Augmenting Paths

Idea: (1) Try increasing the size of current matching by one (until possible)

(2) Find an a path with alternating matched & unmatched edges, and unmatched end-points.



Now 'swap' the edges.

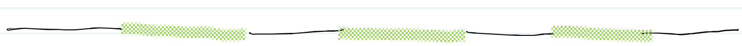
Defn (Augmenting path given matching M):

Any path of G consisting of

(1) Alternating unmatched and matched edges

(2) First and last vertices are unmatched in M.

(M | Green) V Black



Obs: Swapping matched and unmatched in an augmenting path increase the matching $\underbrace{\text{size}}_{= \# \text{ edges}}$ by 1

Thm: M is a max-matching iff there is no augmenting path.

Pf: If there is an aug path then M is not optimal.

As above

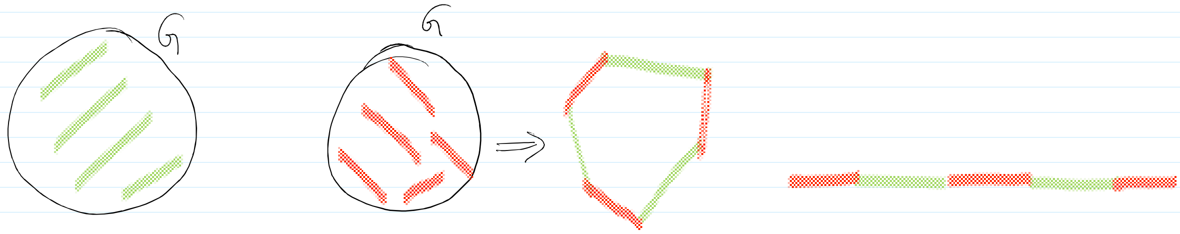
If there is no aug path then M is optimal.

Pf by contradiction: Suppose not $\Rightarrow \exists$ matching M^* where $|M^*| > |M|$

Symmetric difference

Consider $M \triangle M^* := (M \setminus M^*) \cup (M^* \setminus M)$

Obs: Edges of $M \triangle M^*$ form cycles or maximal-paths



Obs: There must be a path which starts and ends with M^* edges.

$\Rightarrow$ this is an augmenting path for M

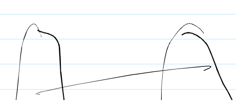
$\Rightarrow$ a contradiction to the assum that M has no aug path $\blacksquare$

Plan for Alg is to find an aug path, or argue that none exists.

Remarks: On general graphs, this is possible using Edmonds' Blossom alg $\leftarrow$ 1960s

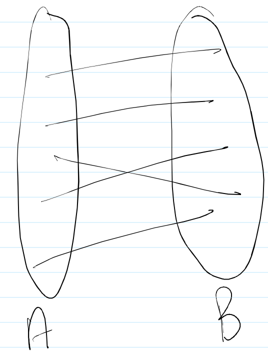
Bipartite Graphs

Is a graph where $V = A \cup B$



Is a graph where $V = A \cup B$

All $e \in E$ are of the form $e = (a, b)$ where $a \in A$, $b \in B$



Exercise: There is a simple alg to find an aug path in bip graphs. (or answer none exists)