

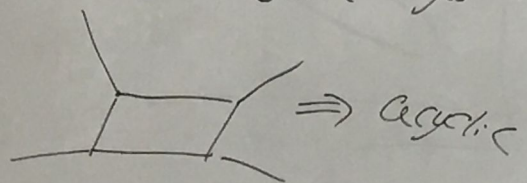
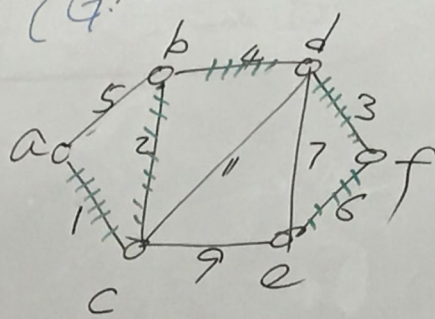
□ Minimum spanning tree (MST)

□ Knapsack (fractional)

Undirected and weighted graph G (G : connected)
positive

Goal: pick subtree of edges of G
s.t. all nodes connected
by selected edges

& cost of edges is cheapest



Assume: edge weights are distinct

Greedy
□ Prefer cheaper edges

□ Kruskal's algo

for each edge e in increasing edge wts

select e if no cycles among selected edges

Cut property: Assume distinct edge wts

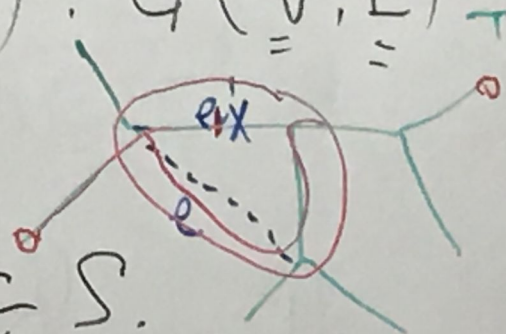
Let S (subset of nodes of G). $G(V, E)$
 $S \subset V$

cut: $(S, V-S)$

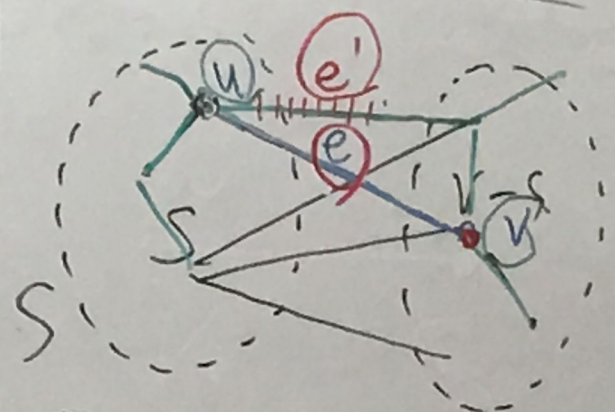
let $e = (u, v)$, s.t. $u \in S$,
 $e \in E$ $v \in V-S$

and is the Cheapest crossing

then e belongs to every
 MST



Proof by switching argument



For contradiction, consider
 MST T , $e \notin T$
 then $\exists T'$ is cheaper than T

(Contradiction)
 $T' = T - \{e\} + \{e\}$

Why Kruskal's work?

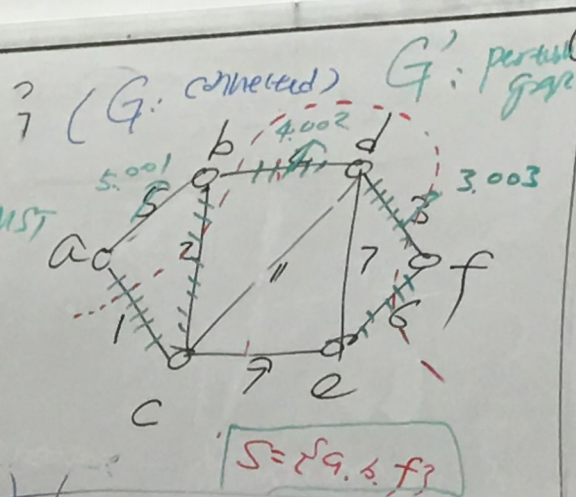
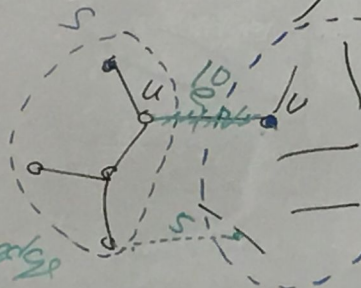
Consider $e = (u, v)$ selected by Kruskal's $\Rightarrow e \in \text{every MST}$

Let $S = \{ \text{nodes radiate from } u \text{ w/ chosen edges} \}$
 $(S, V-S)$

$\Rightarrow v \notin S$

$u \in S$

e is crossing edge
 also cheapest



Assume: edge weights are distinct

Q: 2 edges w/ same weight?

Trick: perturb G by a small amount
 Assume: all edge wts integer

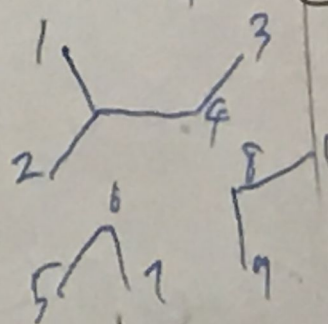
Perturb: add (very small) but distance constant ϵ_i to each edge e_i s.t. $\sum_i \epsilon_i < 0.1$

Claim: a MST in perturbed $G' \Rightarrow$ MST in G

T_1 is MST. 100
 T_2 is not MST. 101 (b/c G has integer wts)
 T_1 is still better than T_2 in G

$m = |E|$, $n = |V|$
 $m = O(n^2) \Rightarrow \log m = O(\log n)$

- ① first add small distance wt in G
- ② run Kruskal's on G
- ③ return same MST



② # iter = $O(m)$
 $\{1, 2, 3, 4, 5\}$
 $\{6, 7\}$
 $\{8, 9, 10\}$
 $O(n)$
 $O(m \log n)$
 Union-Find $\Rightarrow O(n^3)$
 $\Rightarrow O(\log n)$

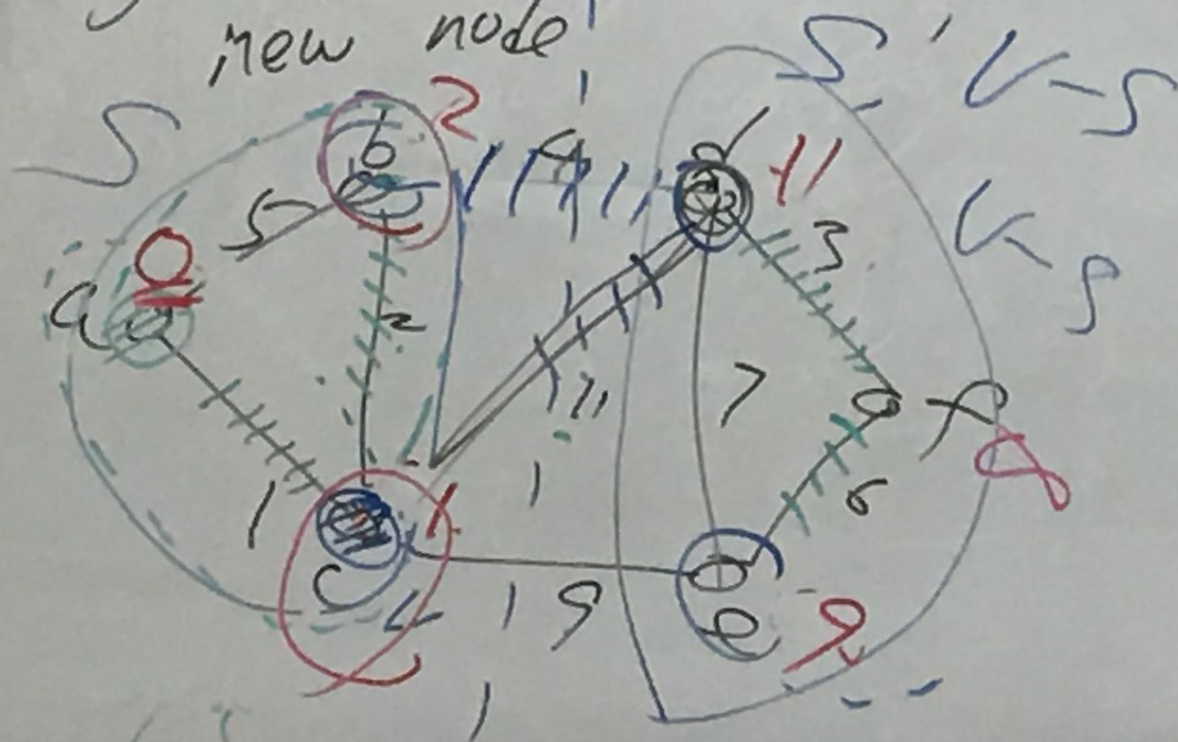
Prim's algo

$C(V)$ = ^{min} cost to V to MST T'

Time:

- ① Start from V_0 , $C(V_0) \leftarrow 0$ all other $C(V) \leftarrow \infty$
- ② Extract V w/ minimum $C(V)$ from V_0
- ③ Grow MST by cheapest edge to add new node!

Priority Queue



$O(\log n)$
 $O(m \log n)$