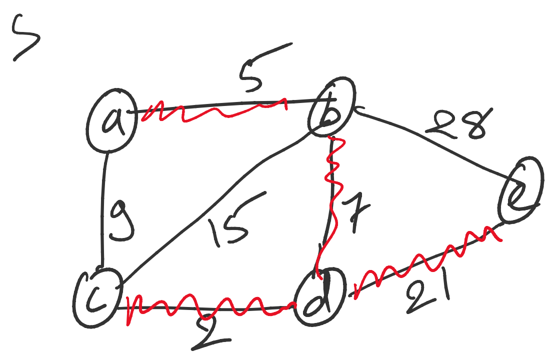
Tuesday, April 12, 2022 10:59 AM

Min Spanning Tree (MST).

Given an undirected connected weighted graph
$G=(V,E)$, $w: E \to \mathbb{R}^{(+)}$,

Find a connected subgraph that includes all the vertices with minimum total weight

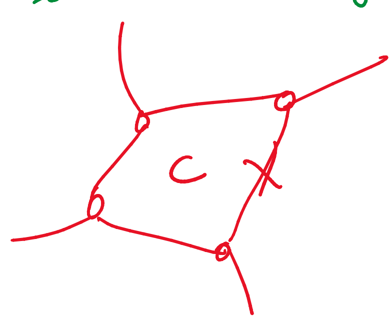
eg.



One sol'n:
$5+15+7+28=55$

$$\boxed{\begin{array}{l}\text{Better Sol'n:}\\ 5+7+2+21=35\end{array}}$$
Best.

Obs1: opt sol'n is acyclic.



Because removing "any" edge from c leaves the graph connected

$\Downarrow$

A feasible/opt sol'n is acyclic, undirected graph

|||

opt sol'n is a tree.

(Appln: Network Design)

Obs2: The #edges in a tree with n vertices is (n-1).

tree with n vertices is $(n-1)$.

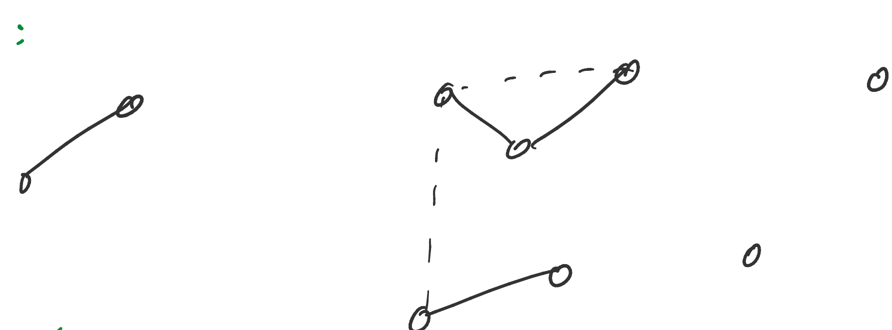
Idea 0: Enumerate $\Rightarrow$ Exponential time.

Idea 1: Greedy!

★ Kruskal's Alg'm (1956): High level.

1. $T = \emptyset$
2. repeat {
3. Pick next smallest weight edge e
4. → if $T \cup \{e\}$ does not contain a cycle then insert e to T.
5. }

Snap shot:

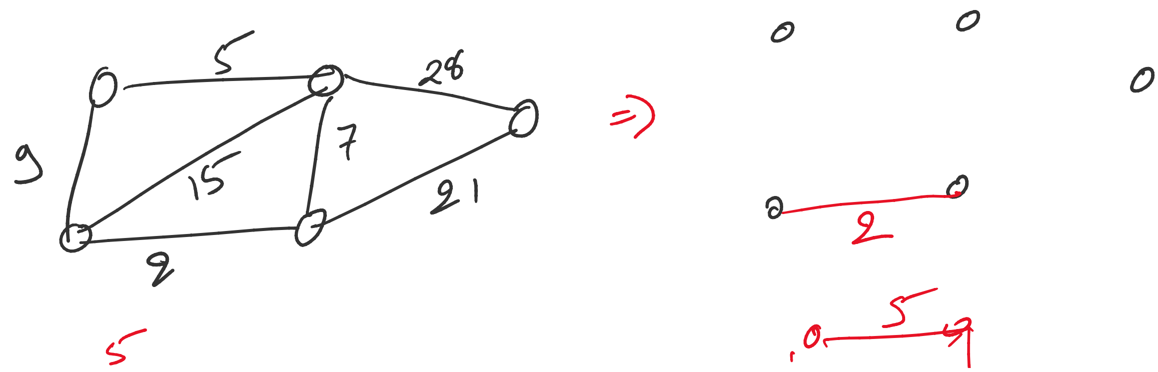


Naive Running Time:

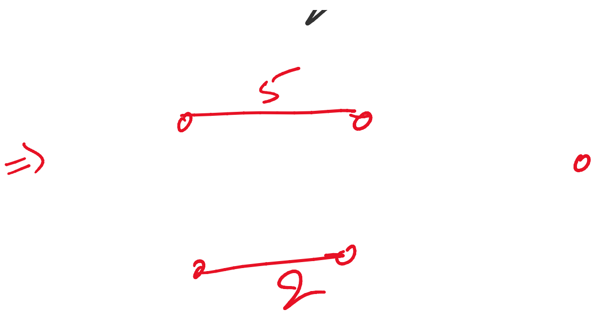
line 4: $O(n)$ (BFS/DFS)

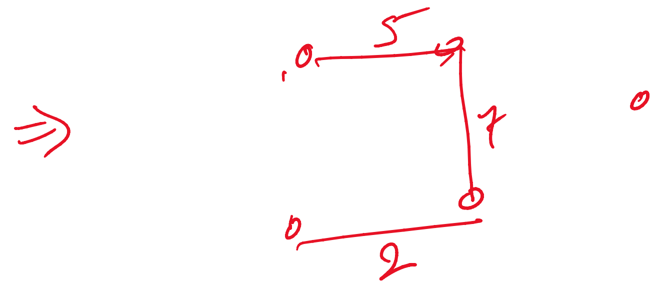
Total: $O(mn)$

eg.

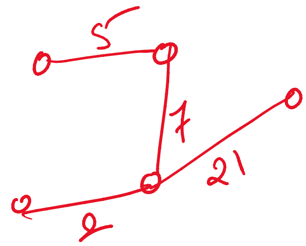


5





Skip 9
Skip 15
⇒

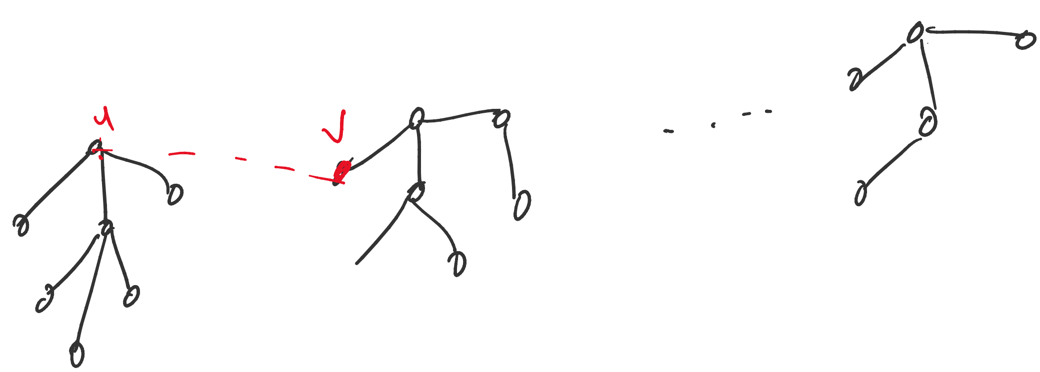


★ Faster Implementation using "union-find" data structure.

1. Sort edges in increasing order of weights. ← $O(m \log m) = O(m \log n)$
2. create $\{v\}$, $\forall v \in V$
3. for each (u,v) in the order
4. if u & v are in two different "sets" then
5. output (u,v). Union the two sets.

Snapshot:

Forest



Running Time:

"union-find"

Lines 4-5: ... for v → $O(\alpha(n))$

Running time.

"union-find"

lines 4-5:

Find set containing given vertex v. → $O(\alpha(n))$ amortized.

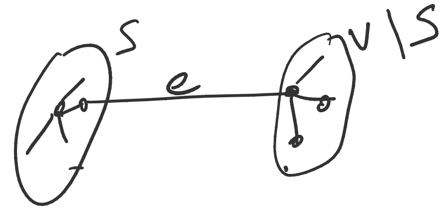
Union two sets. → $O(1)$

$\alpha(n) << \log\log\log \dots \log n$

lines 3-5: $O(m \cdot \alpha(n))$

Total: $O(m \log n + m\,\alpha(n)) = O(m \log n)$.

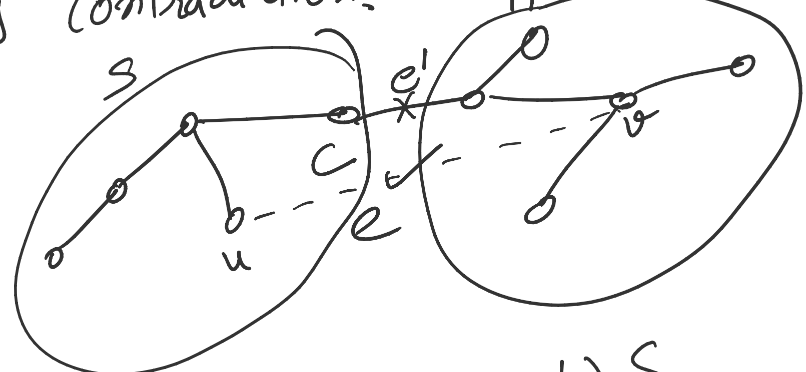
☆ Correctness Pf: (Assume all weights are distinct)



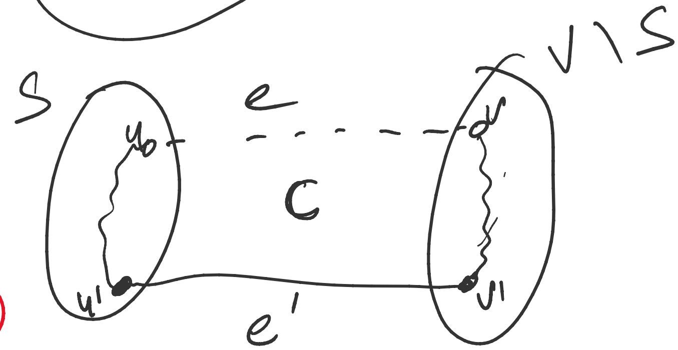
Key lemma: Given $S \subseteq V$

smallest weight edge e betⁿ S & $V \setminus S$ must be in the MST T^*.

Pf: By contradiction. suppose $e \notin T^*$



$T^* \cup \{e\}$



$w(e') > w(e)$

(exchange arg.)

(exchange arg.)

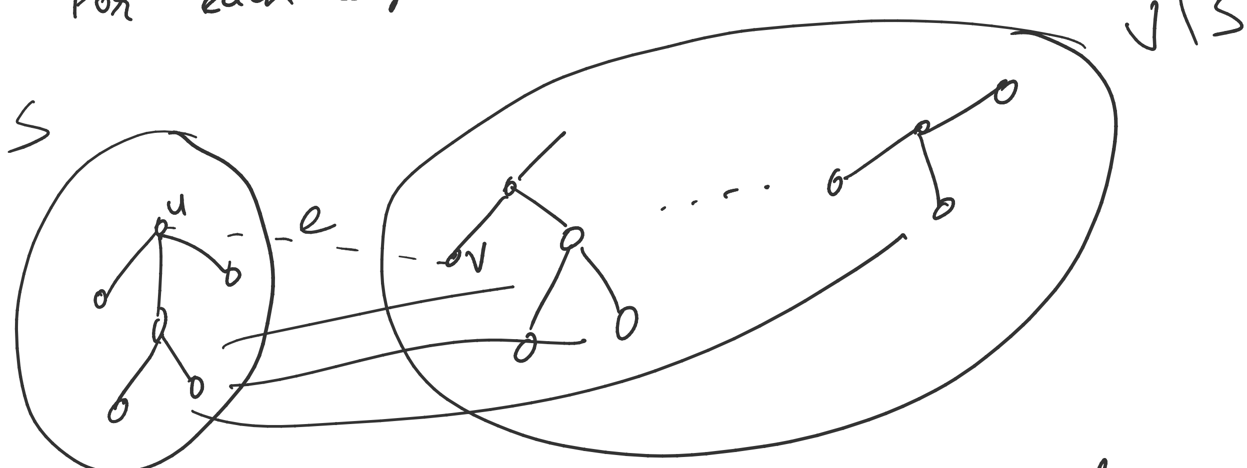
$T = T^* \cup \{e\} \setminus e'$ is a tree that spans all the vertices.

& $w(T) = w(T^*) + w(e) - w(e') < w(T^*)$ because $w(e) < w(e')$.

A contradiction to T^* being MST. $\square$.

Correctness of Kruskal's Algo:

For each edge (u,v) that is inserted to T.



We have that e is of smallest weight crossing S = component of u & V\S.

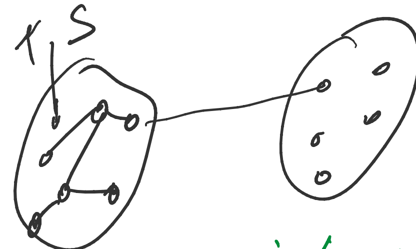
Then by the key lemma (u,v) must be in the MST. $\blacksquare$

☆ Prim's Alg'm (1957): High-level like Dijkstra's.

★ Prim's Alg'm (1957): High-level like Dijkstra's.

1. $S = \{s\}$, $T = \emptyset$
2. while $S \neq V$ {
3. Pick an edge (u,v) s.t. $u \in S$, $v \in V \setminus S$ with min $w(u,v)$.
4. Insert (u,v) to T.
5. Insert v to S

}



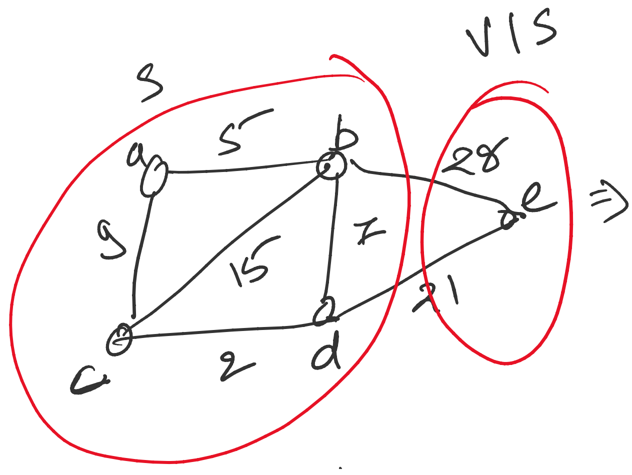
Running Time: like Dijkstra using Fibonacci heap

$O(n \lg n + m)$

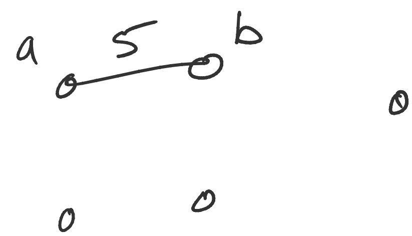
Correctness Pf: Just by the key lemma!

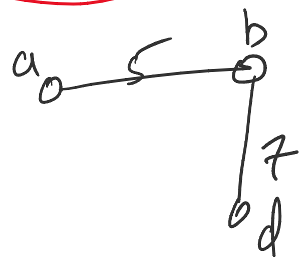
If we add (u,v) to T, then it must be that (u,v) is min weight crossing edge for the current set S.

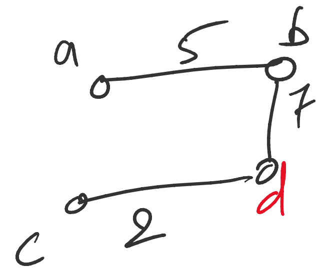
eg.

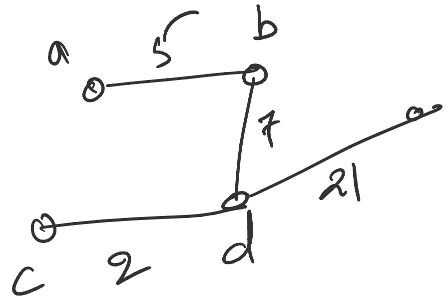


⇒



$\Rightarrow$ 

$\Rightarrow$ 

$\Rightarrow$ 

★ Other Alg'm :

Boruvka	(1926)	$O(m \log n)$
K :	(1956)	$O(m \lg n)$
P	(1957)	$O(n \lg n + m)$
⋮		
Yao'75	(1975)	$O(m \log \log n)$
Fredman, Tarjan	(1985)	$O(m \log^* n)$
⋮		
Gabov et al.	(1986)	$O(m \lg (\lg^* n))$
Karger, Klein, Tarjan	(1994)	$O(m)$ Randomized.
Chazelle '97		$O(m\, \alpha(n))$ det.

OPEN. $O(m)$ det ?

OPEN. $O(n)$ all ;