07/03/2022

Minimum Spanning Tree

Given an undirected graph

$$G = (V, E)$$

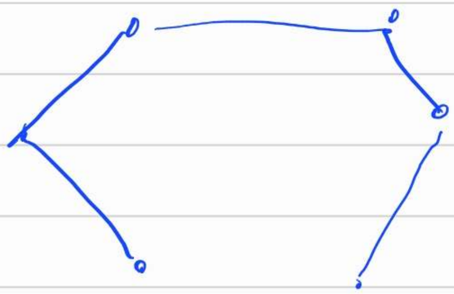
$$w : E \to \mathbb{R}^{\geq 0}$$

Goal: $T \subseteq E$ s.t. (V, T) is connected and $\sum_{e \in T} w(e)$ is minimal.

Lemma: If T is an optimal solution to the above problem the (V, T) is a tree.

Proof:- Suppose not. Let T be a minimal cost solution s.t. T has a cycle.

Delete any edge on this cycle to obtain another Solution T'. T' is connected. and smaller cost.



Defn :- $T \subseteq E$ is a Spanning Tree of $G = (V, E)$ if T is a tree and connects every vertex.

Q: Suppose you want to find a minimum Spanning tree in an unweighted graph G ?

Every tree has (no. of vertices − 1) edges. So find a BFS-tree or a DFS-tree.

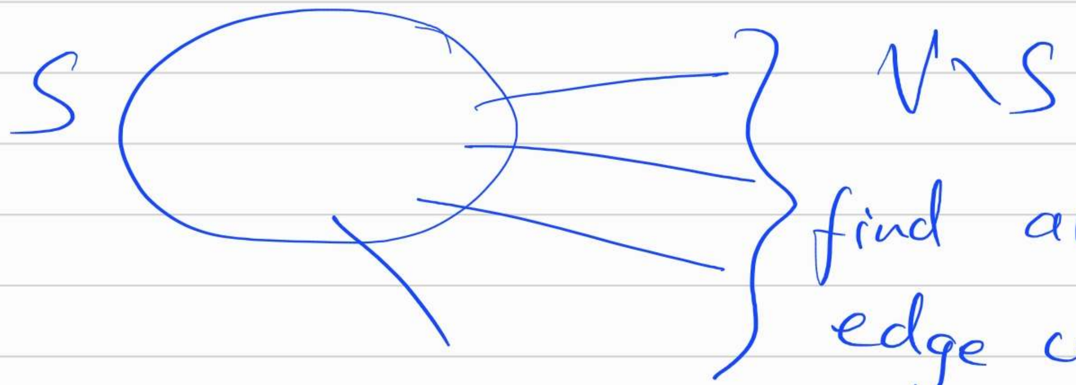
Weighted graphs.

Kruskal's Algo:

- Sort the edges in increasing order wrt costs.
- Add an edge in the tree if adding it doesn't make a cycle with the edges previously added. else discard and move ahead.

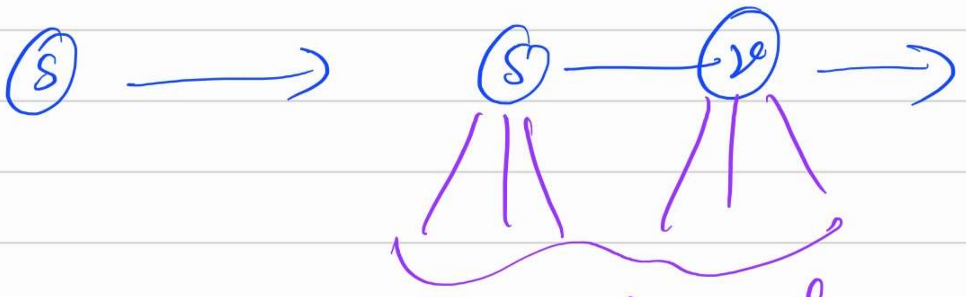
Prim's Algo: grow a tree greedily starting at some vertex

- Initialise $S = \{s\}$



find an edge with minimum weight and add it to the tree.

- add the edge that has minimal weight and connect a vertex in S and $V \setminus S$.



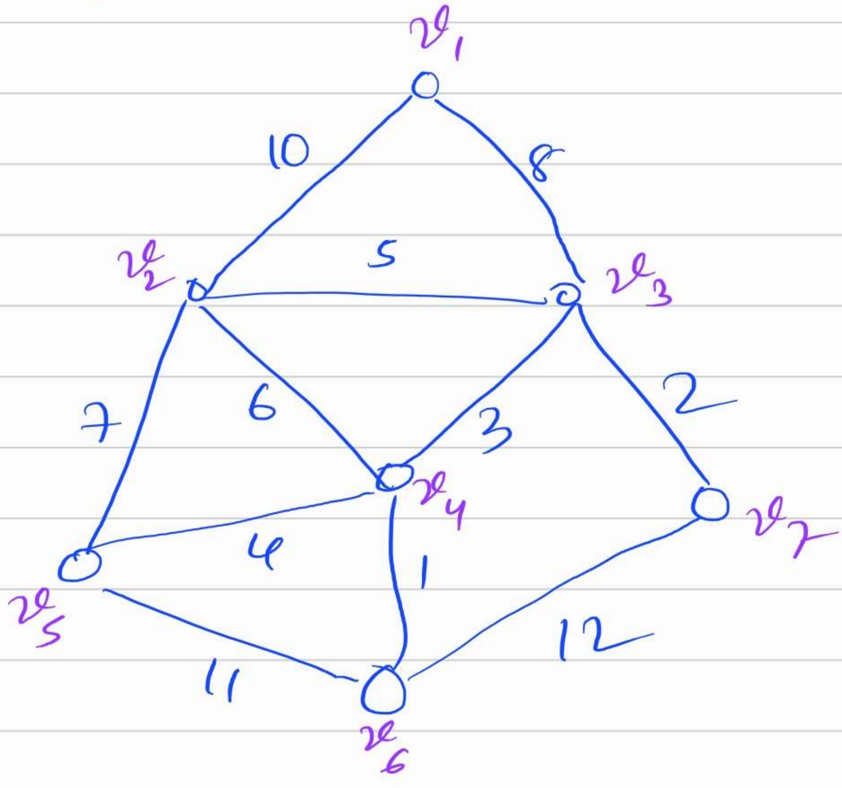
Choose minimal among these edges.

Reverse - Delete Algo:

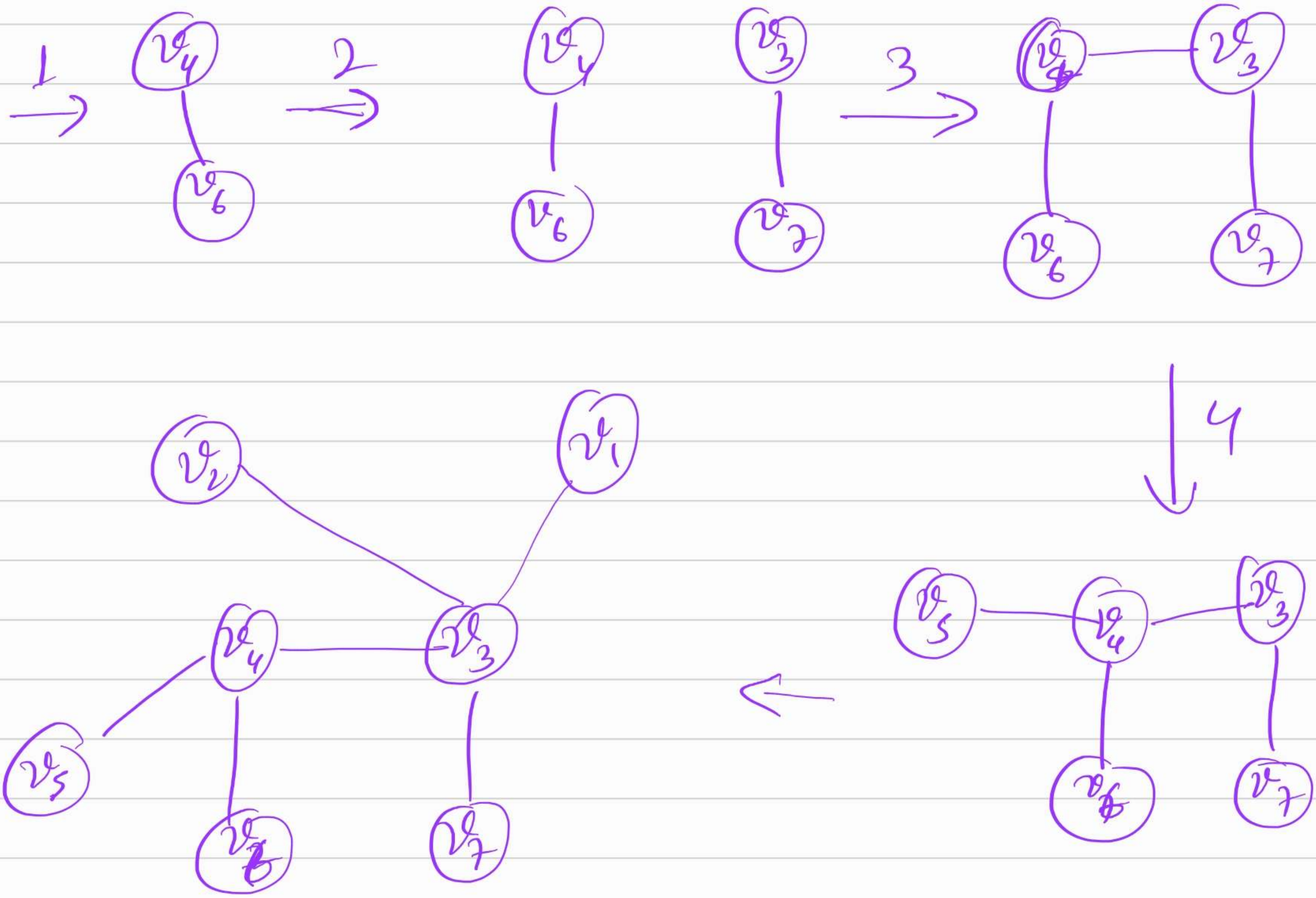
- Sort the edges in decreasing order wrt costs.

- delete an edge from G if deleting it doesn't disconnect the current graph.

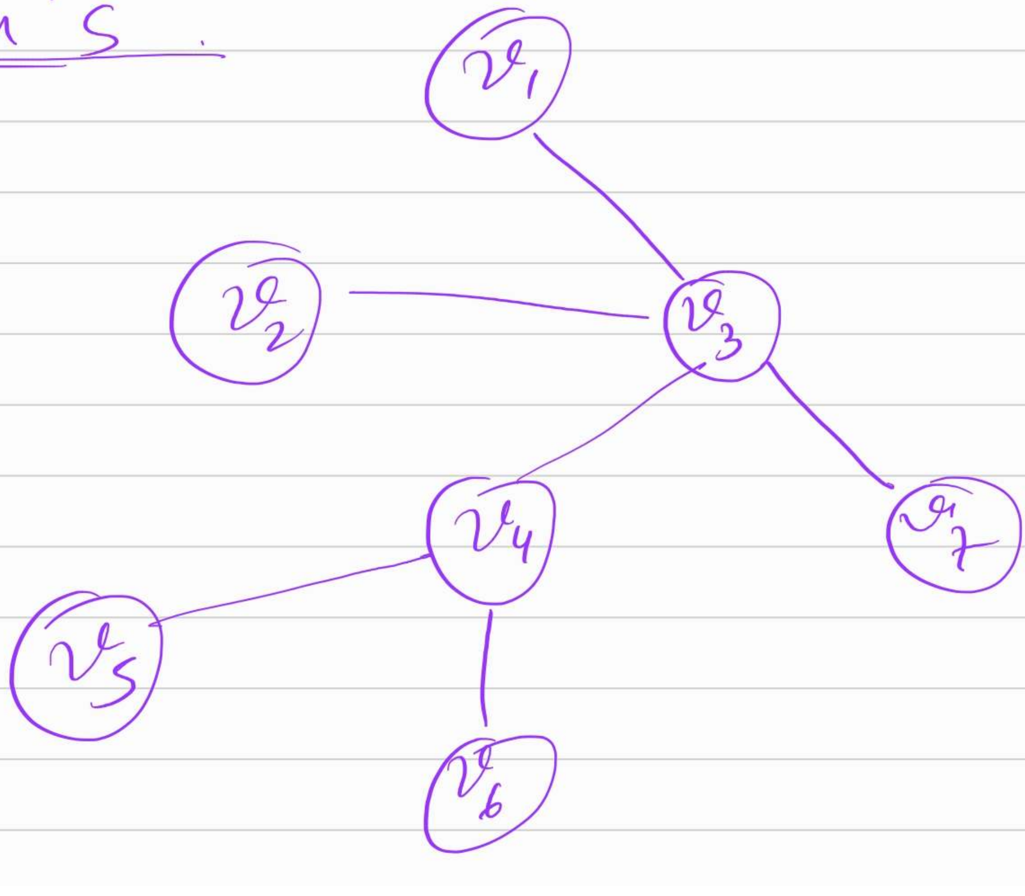
Example:



Kruskal's Algo: 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12



Prim's :



Lemma :- If all edge weights are distinct, then the graph has a unique minimal spanning tree.

Assumption: All edge weights are distinct in the graph.

Q:- When is it safe to add an edge to the tree that you are building?

(CUT Property)

<u>Lemma</u>: Assume all edge weights are distinct.

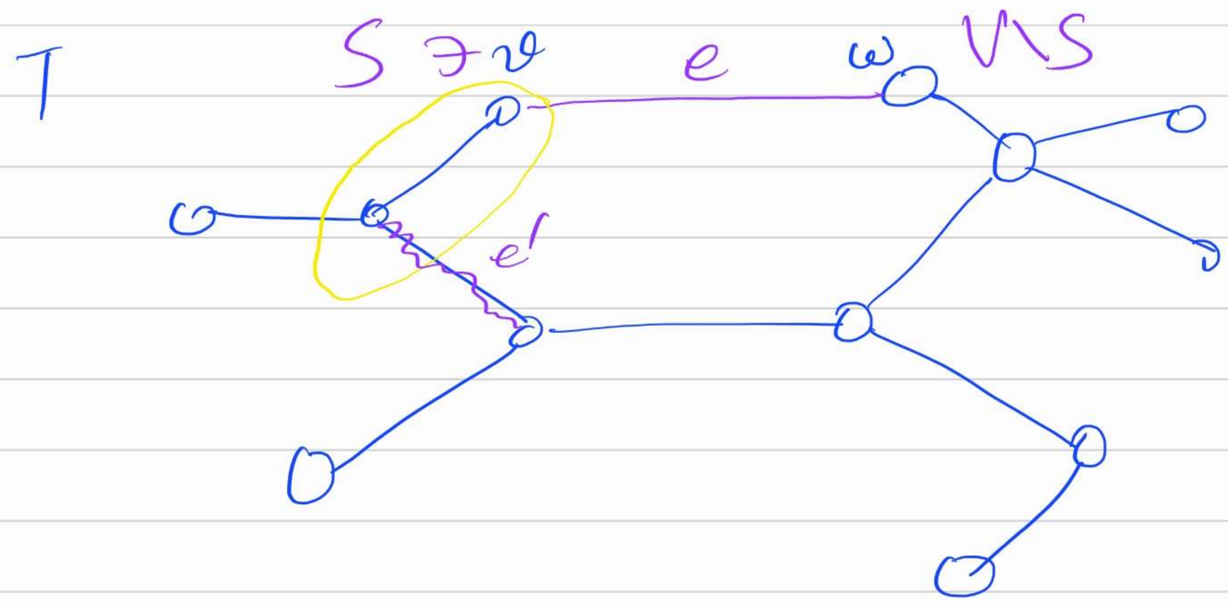
Let $\emptyset \neq S \subsetneq V$.

Let $e = (v, w)$ be the edge of minimal weight crossing the cut $(S, V \setminus S)$.

Then every minimal spanning tree contains the edge e.

<u>Proof</u>: Let T be a spanning tree that doesn't contain e.

We will show that $\exists$ another spanning Tree T' with smaller cost.



Adding e to T will create a unique cycle.

Following the path from v to w in T find the first edge that connects a vertex in S to a vertex in $V \setminus S$.

Call this edge e'.

We know:

$$w(e) < w(e').$$

Delete e' from $T \cup \{e\}$.

$T' := T \cup \{e\} \setminus \{e'\}$.

<u>Claim</u>: T' is a spanning tree with smaller cost.

<u>Lemma</u>:- Kruskal's Algo produces a minimal spanning tree.

<u>Proof</u>:-

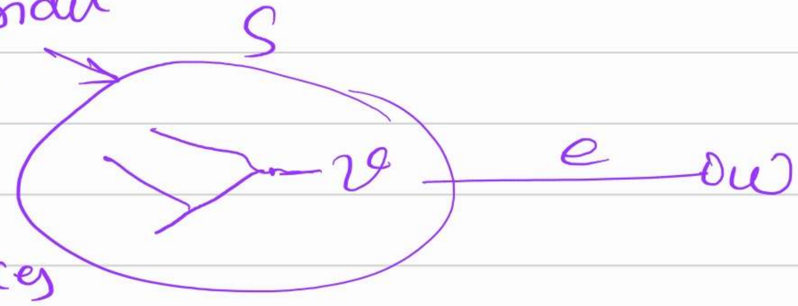
Note it constructs an acyclic graph.

By the behaviour of the algorithm it will produce a single component.

Hence it produces a spanning Tree.

Take any edge e added by the Kruskal's Algo. we will show that $\exists$ a cut $(S, V \setminus S)$ s.t. e is the minimal weight edge crossing this cut.

Consider the set of vertices reachable from v



Consider all the edges in the cut.

None of them have been considered before.

and by our choice of the edge this is minimal among all edges in the cut.