⇒ Spanning Tree

" A spanning tree of a graph G is a sub graph of G that is a tree and contains all the vertices of G"

Importance of Spanning Tree

1. It plays a vital role in designing efficient routing algorithm.
2. Spanning trees construct a sparse sub-graph that tells a lot about the original graph.
3. It has wide applications in many areas such as network design etc.
4. Some hard problem like traveling salesman problem can be solved approximately by using spanning trees.

Important Facts about Spanning Trees.

- Any two vertices in a tree are connected by a unique path.
- Let T be a spanning tree of a graph G, and let "e" be an edge of G not in T. Then (T+e) contains a unique cycles.

Minimum Spanning Tree

A minimum Spanning Tree (MST) of a weighted Graph G is a spanning tree of G whose edges sum is minimum weight.

⇒ Prim's Algorithm

- This algorithm starts from an arbitrary vertex (root) and at each stage, add a new branch (edge) to the already constructed the algorithm halts when all the vertices in the graph have been reached. "In Prim's algorithm the greedy strategy is applied in the sense that at each step the partial spanning tree is augmented with an edge that is the smallest among all possible adjacent edges.

Steps involved in Prim's Algorithm

Step-1 : Start with an Vector u

Step-2 : Select another vertex v such that an edge is formed from u and v and is of minimum weight join uv and included it to set of vertex V.

Step-3 : Among the set of all vertices find other vertex v_i that is not included such that (v_i, v_j) is of minimum labeled and add it to V.

Step-4 : Continue this process until you get a MST

Algorithm

1) $Q \leftarrow V(G)$

2) $A \leftarrow \phi$

3) while $Q = \phi$

4) do $u \leftarrow$ Extract - min (Q)

5) $v \leftarrow Adj(u)$

6) $K \leftarrow w(u,v)$

7) $S \leftarrow \{(u,v)\}$

8) for each $v \in Adj(u)$

9) do if $w(u,v) < K$

10) then $A \leftarrow A \cup \{(u,v)\}$

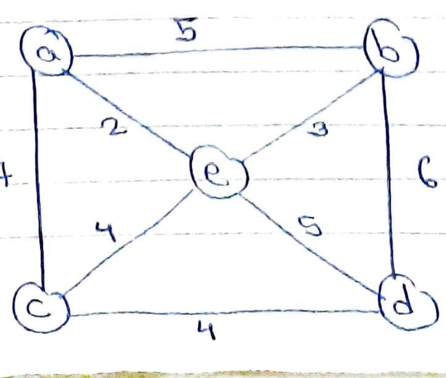
11) else

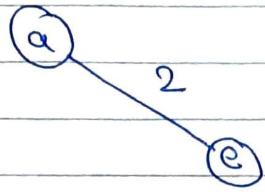
12) $A \leftarrow A \cup S$

13) Union (u,v)

Example

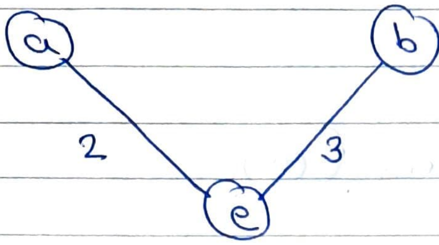
Apply prim's algorithm for the following graph.





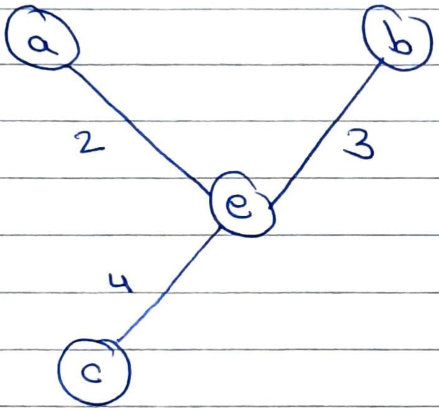
$V = \{a, e\}$

Cost = 2

=> 

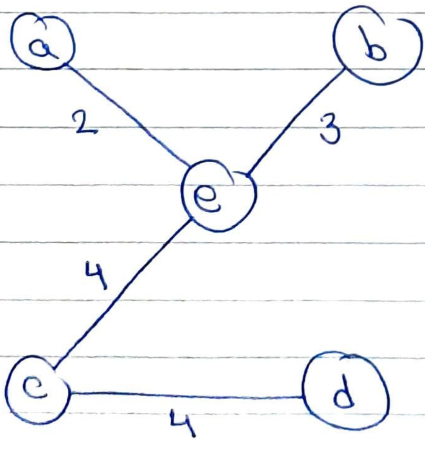
$V = \{a, e, b\}$

Cost = 2+3 = 5

=> 

$V = \{a, e, b, c\}$

Cost = 5+4 = 9

=> 

$V = \{a, e, b, c, d\}$

Cost = 9+4

= 13

Kruskal's Minimum Spanning Tree Algorithm

- In Kruskal's algorithm the selection function chooses edges in increasing order of length without worrying too much about their connection to previously choose edges, except that never to form cycle.

Steps used in Kruskal's Algorithms:

Step-1: Arrange all the edges in increasing order of their weight (length)

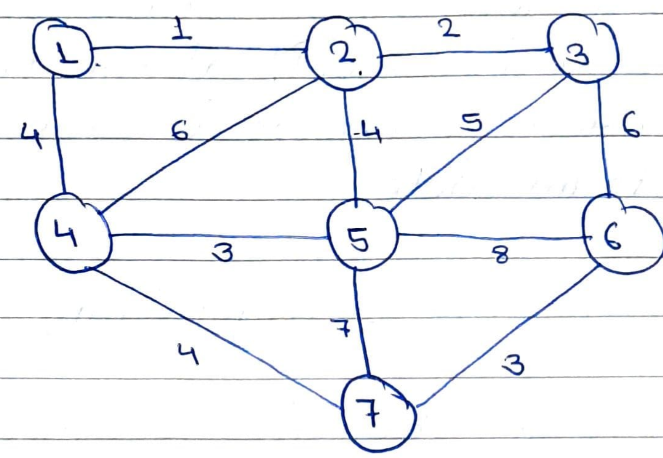
Step-2: Include to minimum spanning tree the edge if it does not form a cycle (closed region / circuit)

Step-3: Continue till all the edges are visited and an MST is formed.

Step-4: Add the length of all edges (take union) in MST to get the minimum cost of the spanning tree.

Example

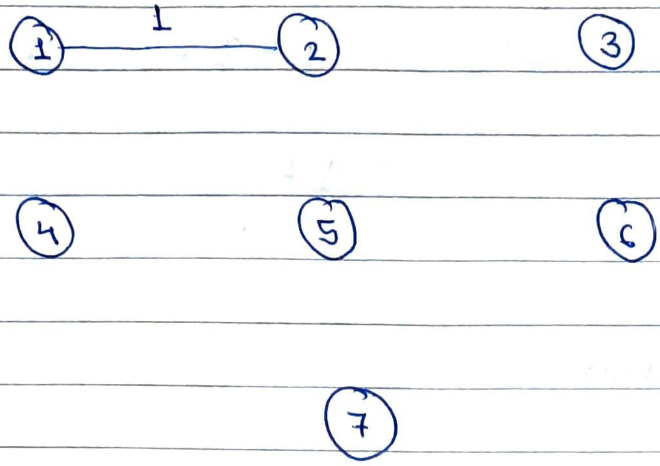
Consider the graph G=(V,E) given below.



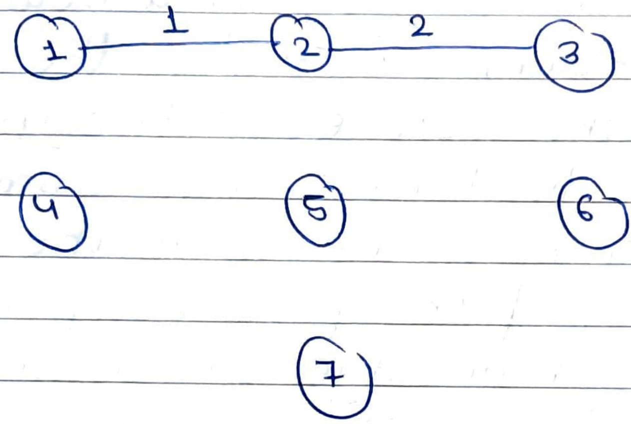
Step-1 Arrange all the edges in increasing order of their weight.

Edge	Weight	Included / Not Included
1 - 2	1	✓
2 - 3	2	✓
4 - 5	3	✓
6 - 7	3	✓
2 - 5	4	× (cycle formed)
1 - 4	4	✓
4 - 7	4	✓
3 - 5	5	× (cycle formed)
2 - 4	6	× (cycle formed)
3 - 6	6	× (cycle formed)
5 - 6	8	× (cycle formed)

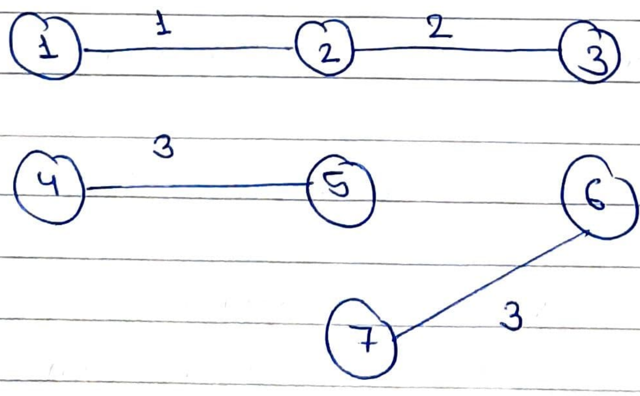
Step-2 Include to minimum spanning tree the edge if it does not form a cycle.



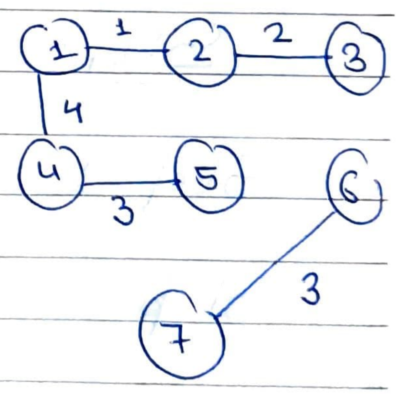
$\Rightarrow$



$\Rightarrow$

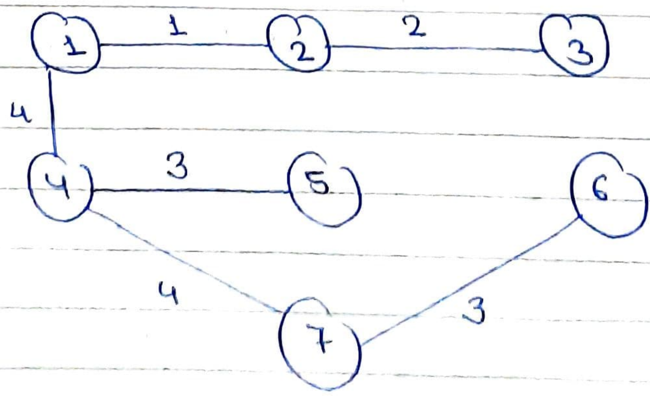


$\Rightarrow$



$\Downarrow$

$\Rightarrow$



Minimum span $w(T) = 1+2+3+3+4+4$

$= 17$