→ **Spanning Tree:** A subgraph S of the given graph G is said to be spanning tree iff:

1. S should contain all the vertices of G.
2. S should contain $(V-1)$ edges where V is the number of vertices.
3. S can't contain cycle.

→ **Minimum Cost Spanning Tree:**

To find the minimum cost spanning tree for a given graph there is two algorithms:

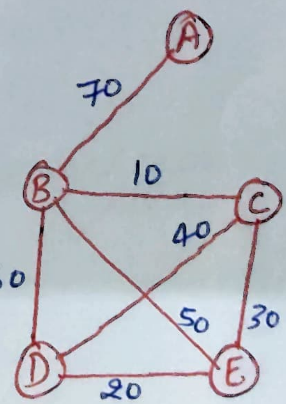
1. Kruskal's Algorithm
2. Prim's Algorithm

Kruskal's Algo

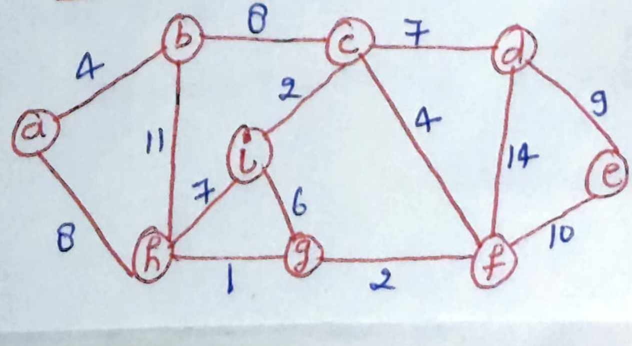
MST_KRUSKAL(G, w)

1. $A = \phi$
2. for each vertex $v \in G.V$
3. MAKE-SET(v)
4. Sort the edges of G.E into nondecreasing order by weight w.
5. for each edge $(u, v) \in G.E$, taken in nondecreasing order by weight.
6. If FIND-SET(u) $\neq$ FIND-SET(v)
7. $A = A \cup \{(u, v)\}$
8. UNION(u, v)
9. return A

EX-1



EX-2



Time Complexity

$O(E) + (V-1)\log E + (E+V)$ [BC]
$O(E) + E\log E + (E+V)$ [WC]
[Using Min heap]

If Array is already sorted:

$O(1) * (V-1) + (E+V)$ [BC]
$O(1) * E + (E+V)$ [WC]

If we use selection sort insted of min heap:

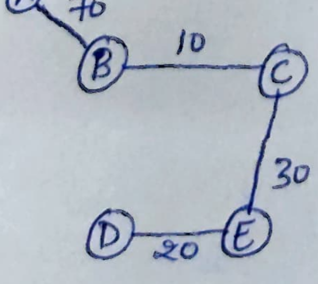
$(V-1) * O(E) + (E+V)$ [BC]
$E * O(E) + (E+V)$ [WC]

Solution EX-1

Step-1 Sort all the edges into increasing order by weight w.

$B \xrightarrow{10} C$
$D \xrightarrow{20} E$
$C \xrightarrow{30} E$
$C \xrightarrow{40} D$
$B \xrightarrow{50} E$
$B \xrightarrow{60} D$
$A \xrightarrow{70} B$

Step-1 Take Edge one by one and add to MST if cycles not formed.



Total cost of MST =

$70 + 10 + 30 + 20 \Rightarrow 130$