above is

# GREEDY TECHNIQUES

## Minimum Spanning tree:

A Spanning tree of an undirected connected graph and it should be a acyclic graph which contains all the vertices of graph.

-> Connected graph
-> No cycles (subgraphs)
-> all vertices.

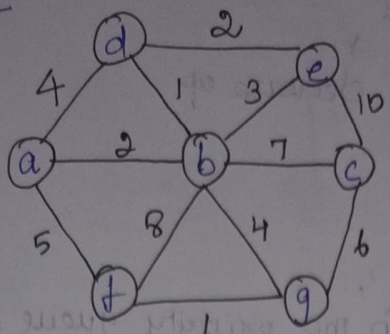
## Prim's Algorithm:

Before that,

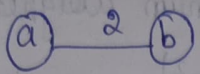
A greedy algorithm is an approach for solving by selecting the best available option which avails at immediate moment.

"PRIM'S" algorithm also called as "JARNIK'S" Algorithm is a greedy algorithm which finds a minimum Spanning tree for weighted undirected graph.

Ex:

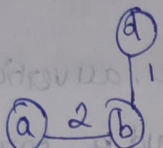



→ connected
→ No cycles
→ all vertices shld be connected.

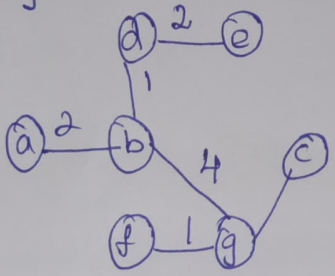



from b - b→d, b→e, b→c, b→f, b→g min is b→d

so take b→d




Similarly




hence all conditions are satisfied.

To find MST:-

weight = 1+4+6+2+1+2

MST weight = 16

Time complexity:

→ Initialize the key value
→ select min vertices
→ find adjacency
→ Reform all possible adjacency places.

$$Tc = V + V\log V + V^2 + E\log V$$

↓ Build heap ↓ delete min ↓ find adjacency ↓ decrease op.

$$= O(V^2 + E\log V)$$

$$TC = O(E\log V)$$

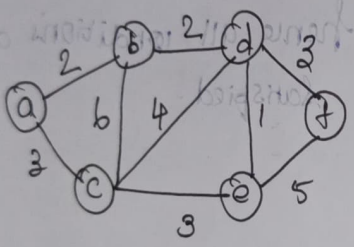
As each edge is inserted in the priority queue only once and the insertion in priority queue take logarithmic time.

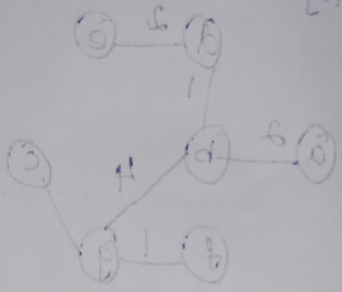
here we are search for minimum weighted edge. So, worst case time complexity $O(V^2)$ - V-no of edges.

## KRUSKAL'S ALGORITHM

It should be a MST, connected vertex, all vertices should be covered and acyclic (No cycle should be formed).

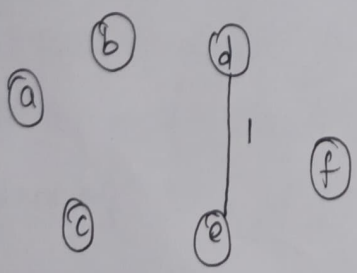
Ex:



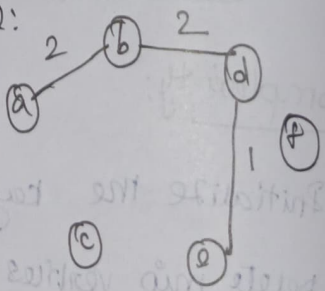



| de | ab | bd | ac | ce | df | cd | ef | bc |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 2 | 3 | 3 | 3 | 4 | 5 | 6 |

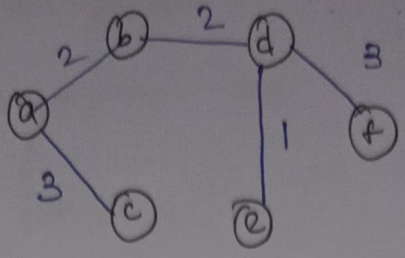
Step 1:




step 2:



Step 3:




Hence connected, all vertices are covered and No cycle is formed.

MST = 2+2+3+1+3 => 11

MST weight = 11

Time Complexity:

O(E log V) where V -> no of vertices.

As a first step in this algorithm, we need to sort the edges in an ascending order of weight. It takes some O(m+v) -> linear time. here we are using small number of edges, E = O(V).

Difference between Prim's and Kruskal's:

Prim's algorithm is faster when compared to Kruskal. As, Prim's algorithm traverse to adjacent nodes for each node and there is no need to sort the edges.

Sorting of smallest weighted edge is done in Kruskal's. So, Prim's is faster than Kruskal's algorithm.