Prim's & Kruskal's Algorithm....

Learn Vid Fun...

Prim's & Kruskal's Algorithms:

Prim's Algorithm and Kruskal's Algorithm are the two greedy algorithms that are used to find the minimum spanning tree (MST) of a given weighted connected undirected graph.

Steps for implementing Kruskal's Algorithm:

Step-1: Sort all the edges from low weight to high weight.

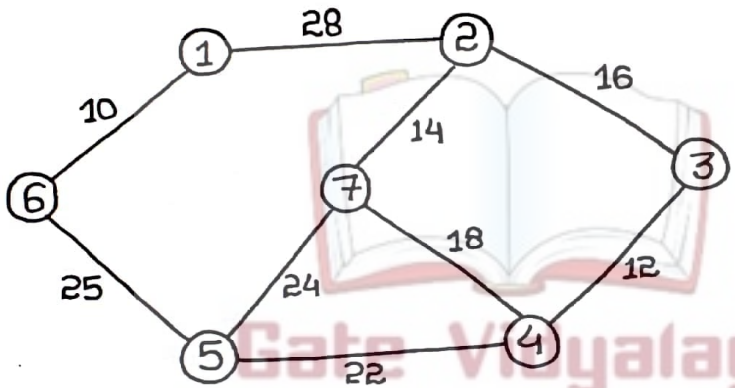
Step-2: Take the edge with the lowest weight and use it to connect the vertices. If adding the edge creates a cycle, then reject that edge.

Step-3: Keep adding edges until all vertices are connected and you get a minimum spanning tree (MST)

Remember: Simply draw all the vertices on the paper and connect them using edges with minimum weights such that no cycle gets formed.

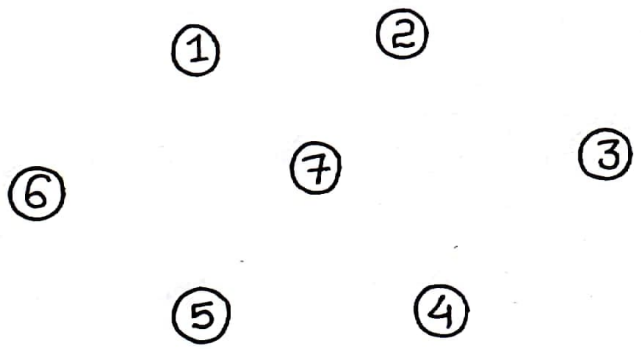
Question:

Construct minimum spanning Tree (MST) for the given graph using Kruskal's Algorithm-

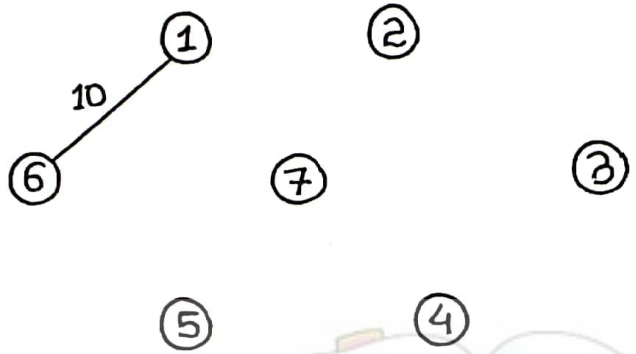


Solution:

Step-1:

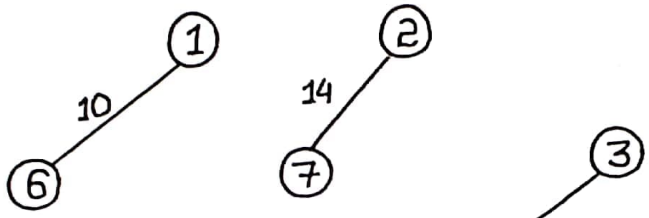


Step-2:

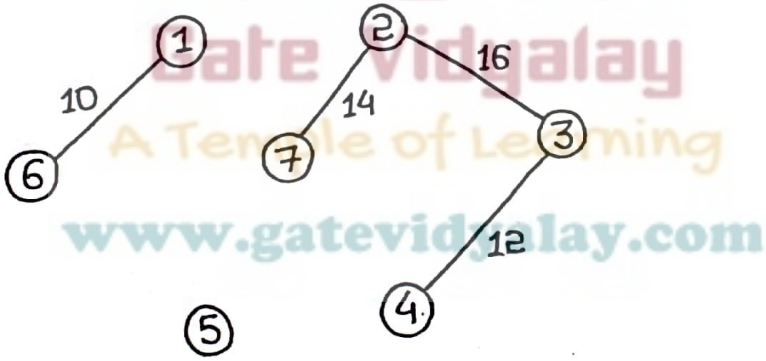


Step-3:

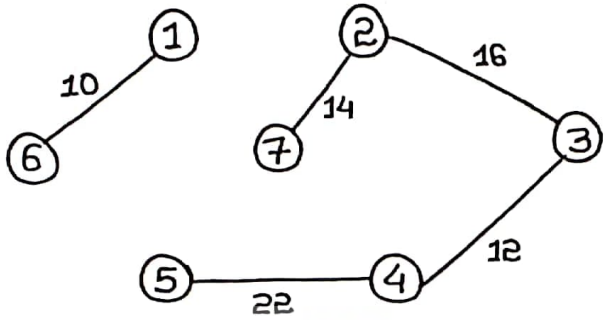
Step-4:



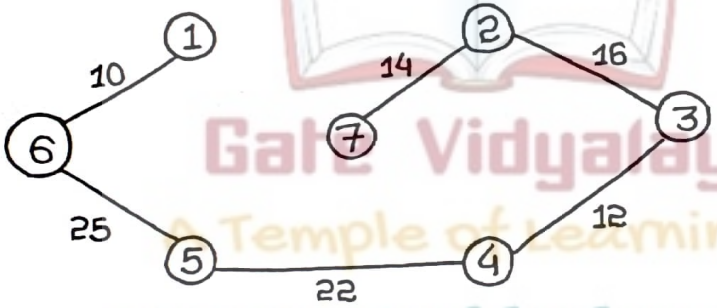
Step-5:



Step-6:



Step-7:



Weight of MST = 10 + 25 + 22 + 12 + 16 + 14

= 99 units

Time Complexity of Kruskal's Algorithm:

Worst case time complexity of Kruskal's Algorithm $= O(E \log V)$ or $O(E \log E)$

How?

The edges are maintained as min heap. The next edge can be obtained in $O(\log E)$ time if graph has E edges. Reconstruction of heap takes $O(E)$ time. So, Kruskal's Algorithm takes $O(E \log E)$ time. The value of E can be almost $O(V^2)$. So, $O(\log V)$ and $O(\log E)$ are same.

Special Case:

If the edges are already sorted, then there is no need to construct min heap. So, deletion from min heap time is saved.

In this case, time complexity of Kruskal's Algorithm $= O(E+V)$