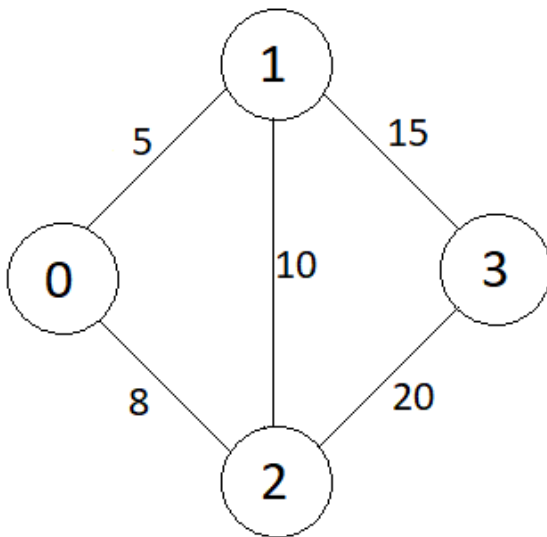


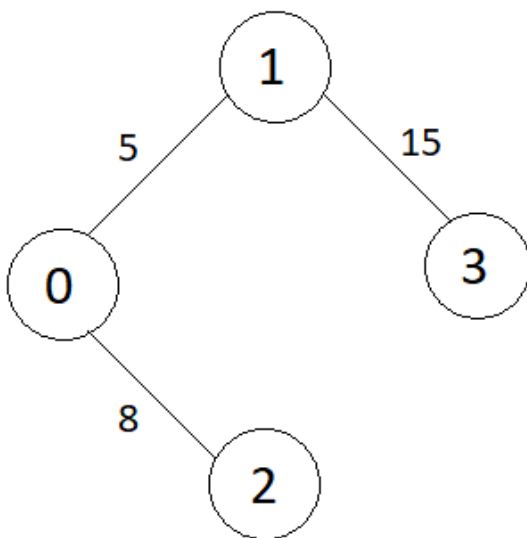
Find Minimum Spanning Tree by using Krukshal's Algorithm [CodeStudio](#)

You are given an undirected connected weighted graph having 'N' nodes numbered from 1 to 'N'. A matrix 'E' of size M x 2 is given which represents the 'M' edges such that there is an edge directed from node E[i][0] to node E[i][1]. You are supposed to return the minimum spanning tree where you need to return weight for each edge in the MST.

Example:



Output:



Approach 1: Kruskal's algorithm to find the minimum spanning tree

- **Explanation:**
 - The edges are sorted in ascending order based on their weights.
 - Union-find operations with path compression and ranking are employed to efficiently check and merge sets of nodes.
 - Iterate through sorted edges and add to the minimum spanning tree if it doesn't form a cycle.
 - The minimum spanning tree is constructed as a vector of edges and their weights.
- **Time Complexity:**
 - **Sorting the edges:** $O(E \log E)$ where E is the number of edges.
 - **Performing union-find operations:** $O(E \log V)$ in practice, where V is the number of vertices.
 - **Overall time complexity:** $O(E \log E)$
- **Space Complexity:**
 - **Additional space for parent and rank arrays:** $O(V)$
 - **Additional space for the minimum spanning tree vector:** $O(E)$
 - **Overall space complexity:** $O(V + E)$