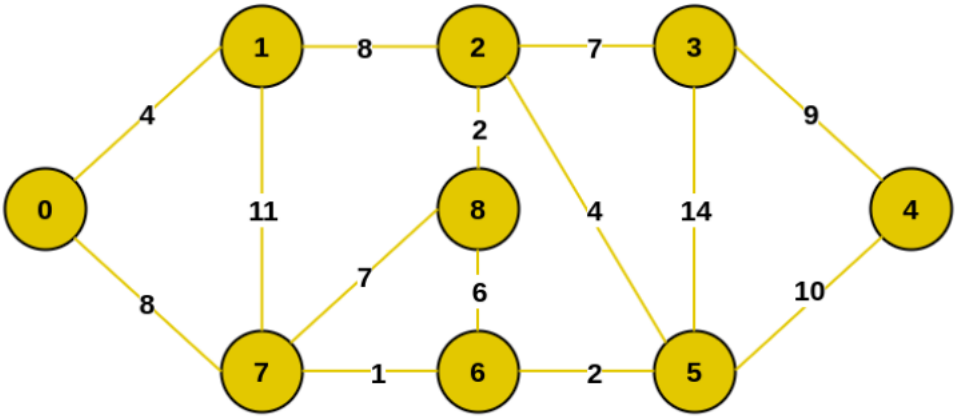
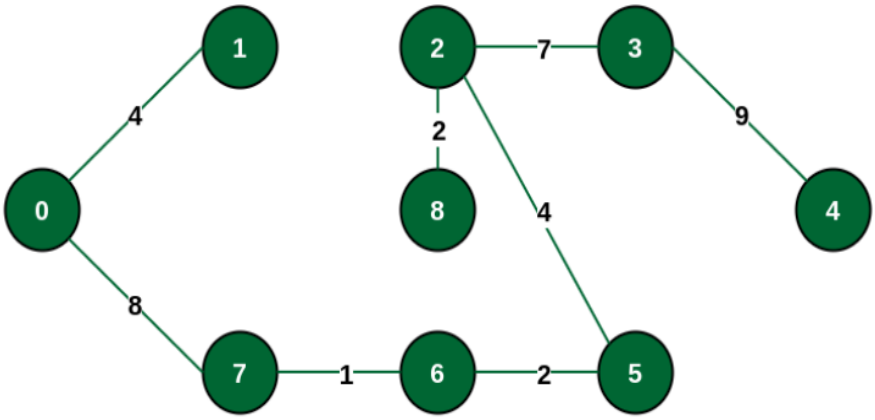
Assignment-8

N – Number of vertices = |V|
M – Number of edges = |M|

Minimum Spanning Tree



Example Graph



Minimum spanning Tree for the given graph

Ans-1) **Kruskal Algorithm.**

Kruskal Algorithm uses greedy and Disjoint Set Union to find the minimum spanning tree. It starts by taking all the minimum edges that are not yet chosen in the graph and tries to add them in the MST. If the two nodes connected by the edges are already present in the tree then we don't take the given edge otherwise we proceed greedily to add edges in increasing order of their weights while they still can be added.

Time Complexity – O(M*logM)
Time Complexity Analysis –
Here the Time taken to sort the edges in increasing order of their weights is O(M*logM) and the dsu functions parent() and union_set() works in nearly constant time over a large number of operations.

Space Complexity – O(N+M)
Space Complexity Analysis –
Space complexity O(N+M) is the amount required to store all the edges of the graph and for the making parent and size array in dsu.

Ans-2) **Prims Algorithm.**

Prims Algorithms uses an approach similar to Dijkstra to find the MST. First it takes an unvisited starting vertex then it goes to all of its neighbouring edges and add them in a set/priority queue and selects the edge with the minimum weight in the priority queue. Thus it finds all the edges.

Time Complexity – O((N+M)*logN)
Time Complexity Analysis –
Here the Time taken to insert an edge in the set/priority queue in O(logN) and there are overall N+M times this operations is done on the queue. So time complexity is O((N+M)*logM)

Space Complexity – O(N+M)
Space Complexity Analysis –
Space complexity O(N+M) is the amount required to store all the edges and maintain priority queue.

- **A graph with a single connected component**
  Both algorithms return the optimal answer in this case.
- **A graph with a multiple connected components**
  Kruskal algorithm works in this case with any modification. However, Prims algorithm in its original method can't detect multiple connected components, so in order to find multiple components we maintain a visited array and ensure that each vertex is visited at least once. Thus we handle the case for multiple connected components.
- **Do both algorithms return the same MST?**
  No, both algorithms can return the same or different MST based on the input graph. However, the cost of MST will be the same in both graphs.
  For instance, take the input:
  Input:
  4 6
  2 3 5
  2 1 5
  2 4 5
  4 3 5
  4 1 5
  3 1 5
  Kruskal Output:
  Cost: 15
  Edges in MST:
  1 2
  1 3
  1 4
  Prims Output:
  Cost: 15
  Edges in MST:
  2 1
  2 3
  2 4

  For the given input of a complete graph the Kruskal and Prim's both give optimal cost however the MST returned by them are different.