Part (a): Kruskal’s Algorithm - Steps

Kruskal (G, V, E):

1. Initialize an empty set T to store the MST edges.
2. Sort all edges in E in non-decreasing order by their weights w(e).
3. Initialize a disjoint set (Union-Find) for the vertices in V:
   a. MakeSet(v) for each vertex v in V. // Each vertex is its own component.
4. For each edge (u, v) in the sorted edge list:
   a. If Find(u) ≠ Find(v): // Check if u and v are in different components
      i. Add edge (u, v) to T.
      ii. Union(u, v). // Merge the components of u and v
   b. If |T| = |V| - 1: // Stop if MST contains (|V| - 1) edges
      Break.
5. Return T and the total weight of the MST.

Part (b): Analysis of Kruskal’s Algorithm

Time Complexity:

1. **Sorting the edges:** Sorting takes O(E Log E), where E is the number of edges.
2. **Find Operation:** Each find operation is O(α(V)) , where α\alphaα is the inverse Ackermann function (which grows very slowly and is effectively constant for all practical input sizes).
3. **Union Operation:** The union operation is O(α(V)).
4. Since we process all edges, the overall time complexity is dominated by the sorting step, making it O(E log E).

Thus, the time complexity of Kruskal’s algorithm is O(E log E).