TOPIC 5.11 : KRUSKAL'S ALGORITHM FOR MINIMUM SPANNING TREE

Problem Statement  
Given a graph represented by an edge list, implement Kruskal's Algorithm to find the Minimum Spanning Tree (MST) and its total weight.

Test Case 1  
Input:  
n = 4  
m = 5  
edges = [(0, 1, 10), (0, 2, 6), (0, 3, 5), (1, 3, 15), (2, 3, 4)]  
Output:  
Edges in MST: [(2, 3, 4), (0, 3, 5), (0, 1, 10)]  
Total weight of MST: 19

Test Case 2  
Input:  
n = 5  
m = 7  
edges = [(0, 1, 2), (0, 3, 6), (1, 2, 3), (1, 3, 8), (1, 4, 5), (2, 4, 7), (3, 4, 9)]  
Output:  
Edges in MST: [(0, 1, 2), (1, 2, 3), (1, 4, 5), (0, 3, 6)]  
Total weight of MST: 16

Aim  
To write a program that finds the Minimum Spanning Tree of a given graph using Kruskal’s Algorithm.

Algorithm

1. Start
2. Sort all edges in non-decreasing order of weights
3. Initialize parent and rank arrays for disjoint set union
4. For each edge (u, v, w):
   * If u and v belong to different sets, include this edge in MST and union the sets
5. Continue until MST contains (n-1) edges
6. Return MST edges and their total weight
7. Stop

Input and Output  
A screenshot of a computer

AI-generated content may be incorrect.

Result  
The program successfully finds the Minimum Spanning Tree and calculates its total weight using Kruskal’s Algorithm.

Performance Analysis  
Time Complexity: O(m log m) for sorting edges, plus nearly O(m α(n)) for union-find  
Space Complexity: O(n) for disjoint set data structures