Data Structures Lab

[22-Mar-2024]

Note: The algorithms for the following graph problems should be implemented using the adjacency list representation of the input graph.

1. **Implement Dijkstra's algorithm.** Not only the shortest path length but also you have to print the path. **Two test case files** are attached for this problem. One test case with 8 vertices and another test case with 200 vertices.

   File Information:

   The file contains an adjacency list representation of an undirected graph with 8 (Test-Case-I) and 200 (Test-Case-I) vertices labeled 1 to 8 (in Test case-I) and 1 to 200 (in Test case-II). Each row indicates the edges incident to the given vertex along with their (nonnegative) lengths.

   For example, in Test case-II file, the sixth row has a "6" as its first entry indicating that this row corresponds to vertex 6. The next entry of this row "141,8200" indicates that there is an undirected edge between vertex 6 and vertex 141 that has length 8200. The rest of the pairs in this row indicate the other vertices adjacent to vertex 6 and the lengths of the corresponding edges. Vertex 1 is the starting vertex. What are the shortest-path distances from vertex to the following ten vertices?: 7,37,59,82,99,115,133,165,188,197.

2. **Implement Kruskal's Minimum Spanning Tree algorithm using UNION and FIND** data structure. There are **two test cases** attached for this problem.
   File format:
   [number_of_vertices] [number_of_edges]
   [one_endpoint_of_edge_1] [other_endpoint_of_edge_1] [edge_1_cost]
   [one_endpoint_of_edge_2] [other_endpoint_of_edge_2] [edge_2_cost]
   ...
   Edge costs can be negative, and are not necessarily distinct.

Test Case-I : Answer: 14.