Data Structures Programming Assignment 2

Due: 2004/6/8 before Class

Problem

Write a program to (1) find the minimum cost spanning tree of a graph, and (2) find the shortest paths from a source to all destinations.

Input

The graph is input as a collection of edges. An edge is represented as $(v_i, v_j, \text{weight}(v_i, v_j))$. An example input file is as below: (Page 320, Figure 6.22 (a))

\[
\begin{align*}
1, & \ 2, \ 28 \\
1, & \ 6, \ 10 \\
2, & \ 3, \ 14 \\
2, & \ 3, \ 16 \\
& \ \vdots
\end{align*}
\]

Output

a. Output the minimum cost spanning tree by printing all the edges in the tree:

\[
\begin{align*}
1, & \ 6, \ 3, \ 4, \ 2, 7, \ 2, 3, \ 4, 5, \ 5, 6, \ \text{and also the total cost}
\end{align*}
\]

b. Output the lengths of the shortest paths from vertex 1 to all destinations:

\[
\begin{align*}
1, & \ 6, \ 10, \ (* \text{shortest path from v1 to v6 is 10} *) \\
1, & \ 2, \ 28
\end{align*}
\]

c. Give three test cases, two cases are connected graph and one is not.

Processing Requirements

a. Use adjacency list to represent the graph.
b. The edges of the graph are non-directional and non-negative.
c. Use Kruskal’s algorithm to find the minimum cost spanning tree.
d. Your program must have Graph, Heap, and Set ADTs.
e. The Graph ADT contains at least the following functions/procedures.
   - CreateGraph(graph), InsertEdge(graph, edge), MinSpanTree(graph),
   - EdgeWeight (graph, vi, vj), ShortestPath(graph, vi).
f. The Heap ADT should include at least: CreateHeap, InsertHeap, and DeleteHeap.
g. The Set ADT should include at least: WeightedUnion, and CollapsingFind.

Extra Credits

a. Obtains the shortest paths, in addition to the lengths of these paths. (10 pts)

Turn In

a. Listing of your programs
b. Listing of the input used
c. Listing of the output produced by the program
d. the input files and programs via the net