Algorithms to Implement

Program Functionality

Implementation

Algorithms and Data Structures Graph Traversal, MST & SPT Algorithm Assignment

You are required to implement Prim's and Kruskal's algorithm for finding the minimum spanning tree for a weighted connected graph and Dijkstra’s shortest path tree (SPT) algorithm. If the implementation is too difficult and/or you can’t debug it, a paper based simulation of Prim/Dijkstra and Kruskal will suffice to pass the assignment.

The program when executed at the command line will 1:prompt the user for the name of a text file which contains a sample graph and also will 2:prompt the user for a staring vertex. The user will then 3:enter this name and vertex (as a number). The graph will then be 4:read from the text file and a 5:graph data structure will be constructed. A method should then be called to 6:compute the graph's SPT & MST after which the 7:SPT & MST should be outputted to the console. While the algorithm is running, it should 8:output some of its workings to the console so that you can see it working step by step.

For Prim/Dijkstra, represent the graph using an adjacency lists data structure.

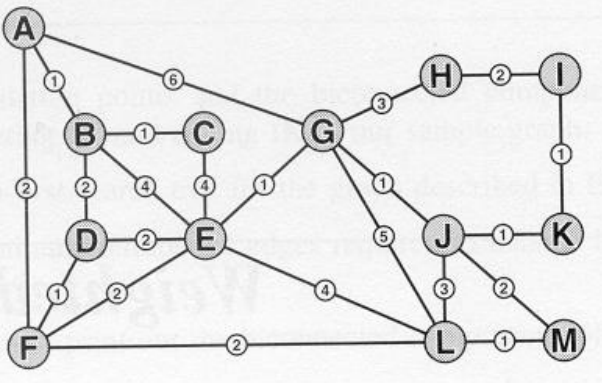
Also include:

* depth first traversal using recursion
* breadth first traversal using a queue (try to use Cormen’s version)

All the above 4 algorithms are all related and should be in one Java source code file.

Kruskal has only one version which uses an array of edges. In contrast Prim & Dijkstra with adjacency lists requires a priority queue or heap. You may use a heap for Kruskal too. The heap code is quite different for Kruskal. Alternatively, with Kruskal you can use Quicksort or Heapsort to sort the edges instead of using a heap. Also, two improvements on Kruskal that you might consider are are: union by rank and path compression.

You are to test your code on the graph below, save this graph in text file wGraph1.txt. For DF() BF(), Prim() and Dijkstra() start with vertex L. I have provided some of the Java code for the Graph and Heap classes.



Also, see if your code can find SPT/MST for one of the very large real world examples of the road graphs provided.

Make sure that

* your code is well commented and well structured
* messy code will lose marks, even if it works