Algorithms and Data Structures

Graph Traversal, MST & SPT Algorithm Assignment

You are required to implement Depth First Traversal, Breadth First Traversal, Prim'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 graph traversal and Prim/Dijkstra might suffice.

The program when executed at the command line will prompt the user for the name of a text file which contains a sample graph and also will prompt the user for a staring vertex. The user will then enter this name and vertex (as a number). The graph will then be read from the text file and a graph data structure will be constructed. Methods should then be called to traverse the graph using DF, BF, Prim & Dijkstra. All should send sensible output to the console. While an algorithm is running, it should output some of its workings to the console so that you can see it working step by step.

Represent the graph using an adjacency lists data structure. No need to do matrix representation.
Also include:

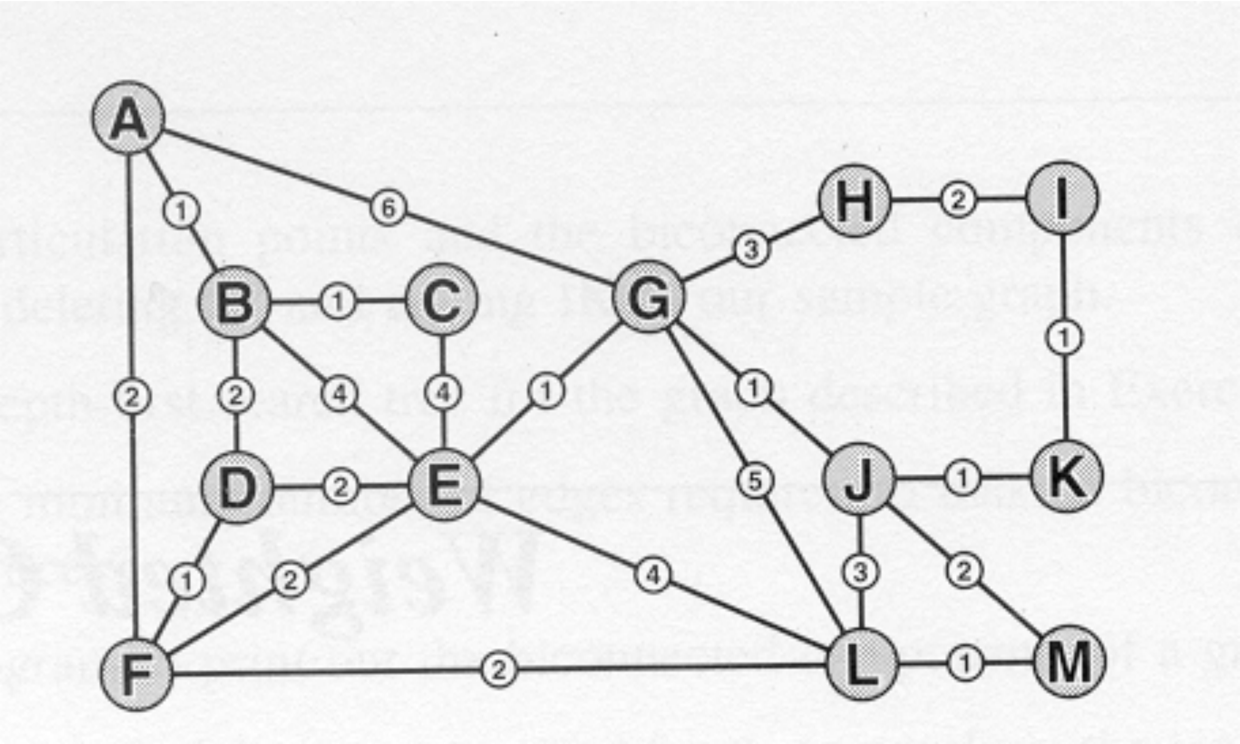
- depth first traversal using recursion (Cormen's version)
- breadth first traversal using a queue (try to use Cormen's version)

Finally, find a large real world graph online, e.g. a road network for a small country, and run your SPT code on it. Make a note of the running time and memory usage. No need to run Prim, DF or BF on this, just Dijkstra.

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

Prim & Dijkstra with adjacency lists requires a priority queue or heap.

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.



Make sure that

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

Report

This is to be submitted to Brightspace in **PDF** format. It should include:

1. Introduction/explanation.
2. An adjacency lists diagram showing the graph representation of the above sample graph.
3. A step by step construction of the MST using Prim's alg, starting from vertex **L**, for the above sample graph. You should show the contents of the heap, parent[] and dist[] arrays for each step in Prim.
4. A step by step construction of the SPT using Dijkstra's alg, starting from vertex **L**, for the above sample graph. You should show the contents of the heap, parent[] and dist[] arrays for each step in Dijkstra.
5. A diagram showing the MST superimposed on the graph (use a highlighter if you want), easy to do using Microsoft Whiteboard.
6. A diagram showing the SPT superimposed on the graph.
7. screen captures showing all your programs executing, especially the output of all implementations for the example graph.
8. Description of your challenging world graph and a discussion on how Dijkstra performed on it.
9. A discussion/analysis/reflection on what you learned or found useful in the assignment.

For submission via Brightspace

1. Report in a file named something like **GraphAlgs-Smith-John-C18345678.pdf**. Make sure to use your name and student number in the report as shown as well as on the report cover page.
2. Two Java code files, one for Prim/Dijkstra/DF/BF and the other for Dijkstra alone on big graph. . Do **not include all the other Visual Studio** files. Also **do not submit compressed ZIP or RAR files**.

Marking

40% for graph traversal (including report with worked examples)
40% for Prim & Dijkstra (including report with worked examples)
20% for Dijkstra on large real world graph