

CSCI203/CSCI803 ASSIGNMENT 3 Due 23:55 06/11/2020

This assignment involves extension to the single source-single destination shortest path problem.

The Program

Your program should:

1. Read the name of a text file from the console. (Not the command line)
2. Read an undirected graph from the file.
3. Find the shortest path between the start and goal vertices specified in the file.
4. Print out the vertices on the path, in order from start to goal.
5. Print out the length of this path.
6. Find the second shortest path between the start and goal vertices specified in the file.
7. Print out the vertices on the path, in order from start to goal.
8. Print out the length of this path.

The data files are constructed as follows:

- Two integers: $nVertices$ and $nEdges$, the number of vertices and edges in the graph.
- $nVertices$ triples consisting of the label and the x - and y -coordinates of each vertex. (An int followed by two doubles)
- $nEdges$ triples consisting of the labels of the start and end vertices of each edge, along with its weight. Note: the weight associated with an edge will be greater than or equal to the Euclidean distance between its start and end vertices as determined by their coordinates. (Two ints followed by a double)
- Two labels, the indicating the start and goal vertices for which the paths are required. (Two ints)

A possible solution to the second shortest path problem is as follows:

For each edge e_i on the shortest path:

Find the shortest path on $(V, E - \{e_i\})$.

The shortest of these is the second shortest path.

Programs must compile and run under gcc (C programs), g++ (C++ programs), java or python3. Programs which do not compile and run will receive no marks.

Programs should be appropriately documented with comments.

All coding must be your own work. Standard libraries of data structures and algorithms such as STL or its Java equivalent may not be used, nor may code be sourced from textbooks, the internet, etc.

You may use dynamic memory allocation to create the data structures you need at the start of program execution. This is the only dynamic allocation that is permitted.

Do not use classes. Java programmers may use classes that contain only public data as a substitute for structs in C and C++.

Marking Guide:

Programs submitted must work! A program which fails, to compile or run will receive a mark of zero for the program.

A program which produces the correct output, no matter how inefficient the code, will receive a minimum of 50% of the mark for the program.

Additional marks beyond this will be awarded for the appropriateness, i.e. efficiency for this problem, of the algorithms and data structures you use.

Programs which lack clarity, both in code and comments, will lose marks.

Submission:

Programs should be submitted via moodle as `ass3.ext` where *ext* is one of `c`, `cpp`, `java` or `py`.