**CS 211 Data Structures and Algorithms Lab**
**Aug -- Nov, 2019**
**Assignment 8**
**Total Marks: 10**
**Due on 2nd November**

The objective of this assignment is to implement Dijkstra's algorithm to find the shortest path distances from a source vertex to every vertex in the input graph, which is directed and has non-negative weights on edges.

## Inputs

Your program should accept two command-line arguments: an input file and the label of a source vertex. A typical execution of your program will be ./a.out sample.graph 34.

The input file represents a directed graph with non-negative integer weights on edges. Every node in the graph is uniquely labelled with a non-negative integer. Every line in the input file is of the form *x y w*, which represents a directed edge from node *x* to node *y*, where the weight of the edge is *w*. No edge is repeated in the input file. The second command-line argument is the label of a source vertex, which is guaranteed to be a vertex in the given graph.

## Task

Implement Dijkstra's algorithm to find the shortest path distances from the given source vertex to all vertices in the given graph. It is recommended that you use min-priority queue with the binary min-heap implementation, but a simpler implementation is also accepted with full credits.

## Output

Your program should create a file named 'dijkstra.txt'. Every line in the output file should corresponds to a shortest path distance from source to a vertex and should be of the form:

<target> <shortest-path-distance-from-source>

For example, if there is a line "21 10023" in the output file and 34 is the source vertex, then it implies that the shortest path distance from 34 to 21 is 10023. If there is no path from the source to a vertex, say 59, then the corresponding output line must be "59 -1".

Shortest path distances of vertices from the source can be printed in the output file in any order.

## Submission and Evaluation

- The program you submit should output dijkstra.txt' when run.

- The main file of your program should be named as <roll no>.<extension>, where roll no. specifies your roll no. and the extension depends on the language you choose (Usage of C/C++ is mandatory for this assignment). Ex: 180040001.c
- Test well before submission. You may use the attached sample input files for testing. The corresponding output files are also attached. We have some hidden inputs with us to test your program. The mark you obtain is purely based on whether your program correctly gives outputs for the hidden inputs.
- If your program has only a single source file, please submit the file as it is. If your program has multiple source files, please submit your code as a zip file where the name of the zip file should be your roll number. It is important that you follow the input/output conventions exactly (including the naming scheme) as we may be doing an automated evaluation. There will be a penalty of 10% (on the mark you deserve otherwise) if you do not follow the naming conventions exactly.
- Follow some coding style uniformly. Provide proper comments in your code.
- Submit only through moodle. Submit well in advance. Any hiccups in the moodle/internet at the last minute is never acceptable as an excuse for late submission. Submissions through email or any other means will be ignored.
- Acknowledge the people (other than the instructor and TA) who helped you to solve this assignment. The details of the help you received and the names of the people who helped you (including internet sources, if applicable) should come in the beginning of the main file as a comment. Copying others' programs and allowing others to copy your program are serious offences and deserving penalty will be imposed if found.
- To consider for first evaluation without penalty, you have to submit your program by **2nd November (11:59 pm)**. If you submit after **2nd November** but on or before **9th November (11:59 pm)**, there will be a penalty of 10% on the marks you deserve otherwise.
- If you do not submit by **9th November**, your program will not be considered for the first evaluation.
- We will do the first evaluation after **9th November**. The marks you obtain will be proportional to the number of correct vertex-distance pairs in the output file. There will be a second evaluation for the assignment for the needy. The details will be shared after the first evaluation.