CS 211 Data Structures and Algorithms Lab
Autumn 2022

Assignment no.	10
Objective	To implement Dijkstra's algorithm to find the shortest path distances from a source vertex to every vertex
Total marks	10
Due date (without penalty)	4th November (Friday) 11:59 pm
Cut-off date (with penalty - 5%)	7th November (Monday) 11:59 pm
Penalty for violating naming convention(s)	5%

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.

Command-line argument:
Your program should receive two command line arguments: a file (input file) and a label of a source vertex.

Input:
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 14.

- The input file represents a directed graph with non-negative integer weights on edges.
- Every node in the graph is uniquely labeled 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.

[Note#1: 14 represents the source vertex]

[Note#2: The provided sample output files such as: dijkstra1.txt, dijkstra2.txt and dijkstra3.txt, corresponds to source vertex 14, 442 and 75, respectively.]

[Note#3: The source to source should be printed zero. This can be verified in the provided output files]

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>

Example:

- If there is a line "47 1452" in the output file and 14 is the source vertex, then it implies that the shortest path distance from 14 to 47 is 1452.
- If there is no path from the source to a vertex, say 21, then the corresponding output line must be "21 -1".
- Shortest path distances of vertices from the source can be printed in the output file in any order.

Submission:

- 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 is mandatory for this assignment). Ex: 210010001.c.
- Do the stress test of your program well before submission.
  (i) You may use the attached sample input files for testing; the corresponding output files are also attached;
  (ii) 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*.
- The marks you obtain will be proportional to the number of correct vertex-distance pairs in the output file.
- 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 5% (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 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 is a serious offense and a deserving penalty will be imposed if found.***

Evaluation:

- To consider for the first evaluation without penalty, you have to submit your program by the due date. If you submit after the ***due date*** but on or before the ***cut-off date***, ***there will be a penalty of 5% on the marks you deserve otherwise***.
- If you do not submit by the ***cut-off date,*** your program will not be considered for the first evaluation.
- We will do the second evaluation later. This is for those who want to improve their marks obtained in the first evaluation or who do not submit for the first evaluation. There will be ***a penalty of 20%*** for those who are submitting for the second evaluation. The details of the second evaluation will be shared later.