CS 211 Data Structures and Algorithms Lab
Spring, 2022-23

Assignment no.	7
Objective	To implement Dijkstra's algorithm to find the shortest path distances from a source vertex to every vertex
Total marks	6
Due date (without penalty)	29th June (Thrursday) 11:59 pm
Penalty for late submission	10%
Penalty for violating naming convention(s)	10%

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 442, 75 and 14, 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:

- Submit a valid **.c** file. [**.cpp** and **.txt** files as source code are not accepted.]
- The program you submit should output: ***'anagrams.txt*** when we run the program for evaluation, any other names for the output file are not accepted during auto evaluation, you will be straight away awarded with **0** marks and it is final.
- The main file of your program should be named after your IIT Dharwad roll number. For example, ***<roll_no>.c***, where ***roll_no*** specifies your IIT Dharwad roll number (220010001.c)
    - Do the stress test of your program 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 marks you obtain are 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) 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 and allowing others to copy your program are serious offenses and a deserving penalty(100%) will be imposed if found.***
- ***Introducing irrelevant code, irrelevant comments, commented code, unnecessary blank lines, etc. will be considered as malpractice.***

- To be considered for the evaluation without penalty, you have to submit your program by the due date. ***No single minute relaxation on late submission.***

- **Submissions after 24 hours of the due date will not be accepted.**