January 2023 CSE 208
Offline Assignment: Single Source Shortest Path

In this assignment, you need to implement the Bellman-Ford and Dijkstra algorithms to solve the Single Source Shortest Path problem for a weighted **directed** graph.

Input

Take input from a file. The first line will contain two space-separated integers n and m, denoting the number of vertices and the number of edges respectively.

In each of the following m lines, there will be three space-separated integers u, v, w denoting an edge, where u and v denote the starting and ending nodes and w denotes the weight of the directed edge.

You need to find both the shortest **distance** and the **path** between the source and destination which will be given in the next line as two integers s and d.

Output

Print the outputs from both the algorithms in a single file. In each case, you need to print the shortest distance between the source and the destination in the first line.

In the next line, you need to print the shortest path between the source and the destination separating the nodes with "->".

For Bellman-Ford algorithm, you need to detect negative weight cycles as well.

See the Sample I/O for further clarification.

Constraints

$1 < n \leq 1000$
$1 < m \leq \binom{n}{2}$
$0 \leq u, v < n$
$0 \leq s, d < n$
$-100000 \leq w \leq 100000$ (**consider** $|w|$ **for Dijkstra**)

Sample I/O

Input File	Output File

5 10
0 1 6
1 2 5
2 1 -2
0 3 7
1 3 8
1 4 -4
3 2 -3
4 2 7
3 4 9
4 0 2
0 4
	Bellman Ford Algorithm:
	-2
	0 ->3 ->2 ->1 ->4
	
	Dijkstra Algorithm:
	10
	0 ->1 ->4

6 8
1 2 -20
0 3 -20
4 5 30
2 4 -40
4 1 10
0 1 10
2 5 50
3 4 10
0 5
	Bellman Ford Algorithm:
	Negative weight cycle present
	
	Dijkstra Algorithm:
	60
	0 ->3 ->4 ->5

Explanation of Test Case #1

Bellman Ford

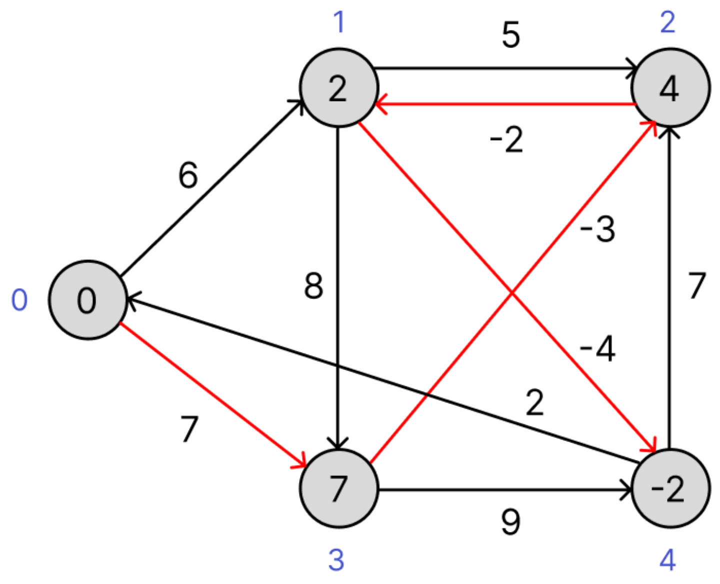

Figure 1: Bellman-Ford shortest path (marked in red)

Dijkstra

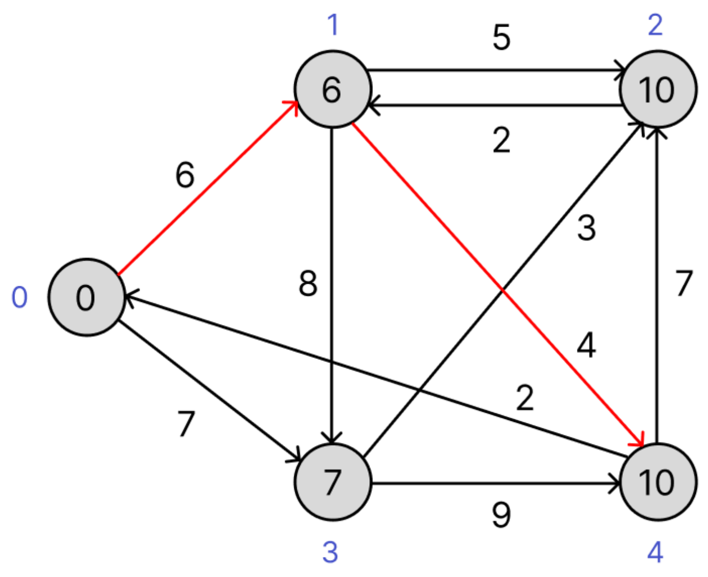

Figure 2: Dijkstra shortest path (marked in red)

Note that we are using the **absolute weights** on the edges for Dijkstra algorithm.

Submission Guideline

1. Create a directory with your 7 digit student id as its name
2. Put the source files only into the directory created in step 1
3. Zip the directory (compress in .zip format; .rar, .7z or any other format is not acceptable)
4. Upload the .zip file on moodle.

For example, if your student id is 2005xxx, create a directory named 2005xxx. Put only your source files (.c, .cpp, .java, .h, etc.) into 2005xxx. Compress 2005xxx into 2005xxx.zip and upload the 2005xxx.zip on moodle.

Submission Deadline: Thursday, June 22, 2023; 11:55 PM

Special Instructions

When writing code, it is essential to ensure readability, reusability, and good structure. This involves using appropriate functions to implement algorithms, giving variables meaningful names, adding suitable comments when necessary, and maintaining proper indentation. You will need to use your offline implementation to solve the onlines. There will be a viva too. So, please understand the concepts before you proceed to code.

Please note that you must rely on your own implementation to solve the assigned tasks. It is strictly prohibited to copy code from any source, including friends, seniors, or the internet. **Any form of plagiarism, regardless of its origin or destination, will result in a deduction of 100% marks for the offline assessment**. Moreover, repeated instances of plagiarism may lead to stricter consequences in accordance with departmental policies.