# January 2020 CSE208: Data Structures and Algorithms II Sessional

### Offline on Single Source Shortest Path Problem

_____

In this assignment, you need to implement the Dijkstra and Bellman Ford algorithms to solve the Single Source Shortest Path problem.

## Input/Output:

You will take input from an input file and give  output to an output file.

## Input Format:

The graph is **directed**.

The first line has two space-separated integers $N$  and $M$, the number of nodes and edges in the graph.

In the next $M$ lines, there will be three space-separated integers $u$, $v$, and $w$ 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 the **distance** and **path** between the source and destination which will be given in the next line as two space separated integers $S$ and $D$, the source and destination.

## Output Format:

In the first line of the output file, you need to print the distance between the source and the destination as a single integer.

In the next line, you need to print the 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:

$1 < N \leq 1000$

$1 < M \leq N * (N-1)/2$

$0 \leq u, v < N$

$0 \leq S, D < N$

$-100000 \leq w \leq 100000$ (for Dijkstra, consider |w|)

## Special Instructions:

Write ***readable, re-usable, well-structured, quality*** code. This includes but is not limited to writing appropriate functions for implementation of the required algorithms, meaningful naming of the variables, suitable comments where required, proper indentation etc. In online, you will need to use these algorithms to solve some realistic problems. With good quality code, your life will be much easier for subsequent assignments.

Please **DO NOT COPY** solutions from anywhere (your friends, seniors, internet etc.). Implement the algorithms with your own style of coding. Any form of plagiarism (irrespective of source or destination), will result in getting -100% marks in the offline. It is your duty to protect your code.

Also, be informed that for repeated offense of plagiarism, the departmental policies suggest stricter measures.

## Submission Guideline:

1. Create a directory with your 7 digit student id as name
2. Put the source files only into the directory created in 1
3. Zip the directory
4. Upload the zip into moodle

For example, if your student id is 1405123, create a directory named 1405123. Put your source files(.c, .cpp, .java, .py, .h, .hpp etc) only into 1405123. Zip 1405123 into 1405123.zip and upload the 1405123.zip into moodle.

Failure to follow the above mentioned submission guideline will result in some penalty.

**Submission Deadline:**

## March 06, 2020 11:55 PM

This is a hard deadline and there shall be no extensions for any reason whatsoever.

**Sample I/O:**

| Input | Output |
|---|---|
| 9 17<br>0 7 60<br>7 1 -150<br>4 8 -70<br>6 4 80<br>5 1 4000<br>8 0 100000<br>2 3 -200<br>8 2 1000<br>0 3 300<br>3 8 50000<br>3 7 -200<br>2 5 120<br>6 3 1500<br>4 0 90<br>5 7 -50<br>1 6 100<br>4 1 -90<br>0 5 | Bellman Ford Algorithm:<br>1140<br>0 -> 7 -> 1 -> 6 -> 4 -> 8 -> 2 -> 5<br><br>Dijkstra Algorithm:<br>1580<br>0 -> 7 -> 1 -> 6 -> 4 -> 8 -> 2 -> 5 |