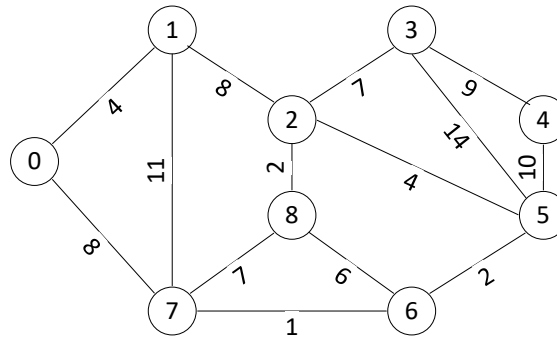


Homework #5

Overview:

You are asked to Create a C++ program that uses Graphs to find the shortest paths between vertices in graphs. We will use the classic Dijkstra's algorithm to find the shortest distance from a starting vertex to every other vertex in a graph. We will use this to find the distance between every pair of vertices in a graph and display the results. As an example, consider a weighted graph:



This graph can be represented in an adjacency matrix as:

--	4	--	--	--	--	--	8	--
4	--	8	--	--	--	--	11	--
--	8	--	7	--	4	--	--	2
--	--	7	--	9	14	--	--	--
--	--	--	9	--	10	--	--	--
--	--	4	14	10	--	2	--	--
--	--	--	--	--	2	--	1	6
8	11	--	--	--	--	1	--	7
--	--	2	--	--	--	6	7	--

Input files

- The input files contain several lines with integer values, separated by spaces as needed. All input files are well formed with valid values. There is no need for exception handling on the inputs and all files follow the same format.
- The first line provides the number of vertices in the graph. This value will always be positive and there will always be multiple rows containing the edges in the graph. There will never be a degenerate graph that is empty or contains no edges.
- The next lines in the file contain the values for each edge in the graph. These lines contain the from-vertex number, the to-vertex number and the weight of the edge. These will be positive integer values, well always appear in groups of three, and each of the numbers is separated by a space.

For example, the **input1.txt** contains:

```
9
0 1 4
0 7 8
1 0 4
1 2 8
1 7 11
2 1 8
2 3 7
2 5 4
2 8 2
3 2 7
3 4 9
3 5 14
4 3 9
4 5 10
5 2 4
5 3 14
5 4 10
5 6 2
6 5 2
6 7 1
6 8 6
7 0 8
7 1 11
7 6 1
7 8 7
8 2 2
8 6 6
8 7 7
```

This corresponds to the graph shown above. We can compute the shortest distance from any vertex to any other vertex. Suppose we pick vertex 0 as the starting point, the distance to every other vertex is then:

```
0   4  12  19  21  11   9   8  14
```

Continuing, we can compute the shortest distance to all vertices starting from vertex 1, 2, and so on.

Output files

- Output the minimum distance through the graph to each other vertex, starting from each vertex in order. Output the results for each starting vertex, one per line.
- Output the minimum distance one each line with 2 spaces allowed for each number and a space before and after each value.

For example, the **ans1.txt** contains:

0	4	12	19	21	11	9	8	14
4	0	8	15	22	12	12	11	10
12	8	0	7	14	4	6	7	2
19	15	7	0	9	11	13	14	9
21	22	14	9	0	10	12	13	16
11	12	4	11	10	0	2	3	6
9	12	6	13	12	2	0	1	6
8	11	7	14	13	3	1	0	7
14	10	2	9	16	6	6	7	0

Reminder

- Turn in your lab assignment to our Linux server.
- Make sure to only have one (1) .cpp file with the main() function in your working directory, otherwise your program will fail the grading script.
- Ensure that the file names do not contain spaces or the grading script will fail
- Create a folder under your root directory, name the folder hw5 (case sensitive), copy all your .cpp and .h files into this folder, and not into a subdirectory (ArgumentManager.h is also needed)
- Only include the necessary files (.cpp and .h files) in your working directory in your final submission.
- To test your program, you may wish to copy the input files and answer files onto the server and run your program. Do not include any outputs files and after verifying that the code passes the tests, **delete any output*.txt files.**

Please reach out to myself or the TAs for any clarifications or typos.