Coding Assignment 4: CS2233

7th November, 2024

Max Marks 70

Kindly adhere to the following instructions.

- Please write a C/C++ program corresponding to each problem. Your code should be well commented on, and variable names should be appropriately chosen.
- Create a folder and put all the code files and name all the files as the question number (i.e. 1.cpp/1.c), give a name to the folder as "yourRollNo", zip the folder and submit it to the Google Classroom portal.
- Strictly follow the input and output format for each problem.
- Any code that does not follow the input-output criteria won't be evaluated and will get **ZERO**.
- Your code will also be checked against plagiarism (both from web and peer).
- Any form of plagiarism (web/chatGPT/with peers) will be severely penalised and will result in an F grade.
- The submission (strict) timeline is 22nd November.

1 Coding Problems:

1. **Graph Traversal:** *(Keep 1.a, 1.b, 1.c in* ***separate*** *'.c' files)*

   (a) Suppose $G = (V, E)$ - an undirected graph, and a vertex $s \in V$ are given as input. Given the non-recursive implementation of the DFS traversal algorithm. $\forall v \in V$, you need to output both - the length of the path from s, and the list of vertices in the path. **10 Marks**

(b) Suppose $G = (V, E)$ - an undirected graph, and a vertex $s \in V$ are given as input. Implement the BFS traversal algorithm. $\forall v \in V$, you need to output both - the length of the path from s, and the list of vertices in the path. **10 Marks**

(c) Given a graph as input, write an algorithm to check whether it is bi-partite. Recall that a graph is called bi-partite if its vertices can be partitioned into two sets such that there are no edges within the same partition and edges exist only across the partition. **10 Marks**

Input format

- The first line of input consists of three space-separated integers, N, M and s, denoting the number of vertices in the graph, the number of edges in the graph and the source vertex s (only s will be not there for the problem 1.c) respectively.
- Following M lines consist of two space-separated each a, b, denoting there is an edge between vertex a and vertex b. Assume all vertex is denoted in a numeric fashion (assume it's 0 indexed, which means, for n vertices, vertices numbering would be $0, 1 \ldots n-1$).

Output format

- For problems 1.a and 1.b, Your output should contain n lines. Each line will contain the path from s to a vertex v, followed by the length of that path.
- For problem 1.c, just print YES if the input graph is bipartite o.w print NO.

Example:
Input:

```
4 4 3
0 2
1 2
1 3
2 3
```

Output:
1.a

```
3 1 1
3 1 2 2
3 1 2 0 3
```

1.b

```
3 1 1
3 2 1
3 2 0 2
```

1.c
NO

Note: *Even though the I/P-O/P format is written in a single paragraph, you should code it in three separate files. If you have any questions regarding this, you can ask in Google Classroom.*

2. **Minimum Spanning Tree (MST):** *(Keep 2.a, 2.b in a **single** '.c' file)*

   (a) Suppose $G = (V, E)$ an undirected weighted graph is given as input. Compute the MST of this graph using Prim's algorithm. You need to output the list of edges that comprises MST. **10 Marks**

   (b) Suppose $G = (V, E)$ an undirected weighted graph is given as input. Compute the MST of this graph using Kruskal's algorithm. You need to output the list of edges that comprises MST. **10 Marks**

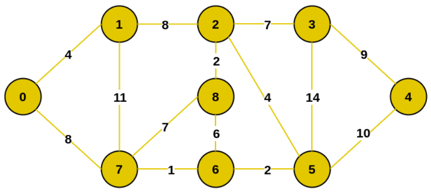
   Input format

   - The first line of input consists of three space-separated integers, N, M and s, denoting the number of vertices in the graph, the number of edges in the graph and the source vertex s, respectively.
   - Following M lines consist of three space-separated integers a, b, and w, denoting there is an edge between vertex a and vertex b, followed by the weight of that edge. Assume all vertex is denoted in a numeric fashion (assume it's 0 indexed, which means, for n vertices, vertices numbering would be $0, 1 \ldots n-1$).

   Output format

   - Your output should contain *2n* lines in total.
   - Print "Prim's Algorithm"
   - First n lines, where the first $n-1$ lines will contain the $n-1$ edges of the MST followed by the total weight of the MST in the last line. This would be the output for Prim's algorithm.
   - Print "Kruskal's Algorithm"
   - Next, n lines, where the first $n-1$ lines will contain the $n-1$ edges of the MST followed by the total weight of the MST in the last line. This would be the output for Kruskal's algorithm.

   Example:
   Input:



Example of a Graph

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

Output:

Prim's Algorithm

```
0 1
0 7
2 3
2 5
2 8
3 4
5 6
6 7
37
```

Kruskal's Algorithm

```
0 1
0 7
2 3
2 5
2 8
```

```
3 4
5 6
6 7
37
```

Note: *There might be multiple MSTs with the same weight; printing anyone would be fine.*

3. **Single Source Shortest Path:** (*Keep 3.a, 3.b in separate ‘.c’ files*)

   (a) Suppose $G = (V, E)$ - a directed weighted graph with non-negative edge weights and a vertex $s \in V$, are given as input. Implement Dijkstra’s algorithm for computing the shortest path of $\forall v \in V/s$ from the vertex s. $\forall v \in V/s$, you need to output both - the length of the shortest path of v from s, and the list of vertices in the path. **10 Marks**

   (b) Suppose $G = (V, E)$ - a directed weighted graph (negative edge weights are also allowed), and a vertex $s \in V$, are given as input. Write a code to check if the graph has any cycle of negative length. If it has one, then report it. If the graph doesn’t have any negative cycle, then compute the shortest path of all the vertices from s. $\forall v \in V/s$, you need to output both - the length of the shortest path of v from s, and the list of vertices in the path. **10 Marks**

Input format

- The first line of input consists of three space-separated integers, N, M and s, denoting the number of vertices in the graph, the number of edges in the graph and the source vertex s, respectively.

- Following M lines consist of three space-separated integers a, b, and w, denoting there is a directed edge from vertex a to vertex b, followed by the weight of that edge. Assume all vertex is denoted in a numeric fashion (assume it’s 0 indexed, which means, for n vertices, vertices numbering would be $0, 1 \ldots n - 1$).

Output format

- Your output should contain $n - 1$ lines in total.

- You need to find path and it’s length between vertex s and v where $\forall v \in V/s$.

- Each line should contain the shortest path length from s to (v) and also print the vertices in path.

- For the problem 3.b, if graph contains negative cycle just print ”Contains negative cycle”.

- If there are multiple shortest path print any one of them.
- If there is no path for any pair of vertices print ”No path exits” for that pair.
- Strictly follow the given input and output format for both questions.

Example:

Input:

```
8 13 0
0 1 2
0 2 5
0 3 1
1 2 3
1 4 4
2 3 2
2 5 1
3 5 7
4 5 6
4 6 8
5 6 2
5 7 3
6 7 1
```

Output:

```
0 -> 1  path length = 2  path = [0, 1]
0 -> 2  path length = 5  path = [0, 1, 2]
0 -> 3  path length = 1  path = [0, 3]
0 -> 4  path length = 6  path = [0, 1, 4]
0 -> 5  path length = 6  path = [0, 1, 2, 5]
0 -> 6  path length = 8  path = [0, 1, 2, 5, 6]
0 -> 7  path length = 9  path = [0, 1, 2, 5, 7]
```