SCS1308: Foundations of Algorithm

Programming Question: Implement BFS Algorithm in C

Problem Statement:

Write a program in C to implement the Breadth-First Search (BFS) algorithm. The program should take the input of a directed/undirected graph from a text file and output the BFS traversal path, the weight of the graph, and any unreachable nodes (if present).

Specifications:

1. **Input:**
    - The graph input will be provided in a text file (graph.txt).
    - **File Format:**
        - The **first line** contains the number of nodes (e.g., 4 for nodes A, B, C, D).
        - The **second line** specifies node name (e.g., nodes A, B, C, D).
        - The **third line** specifies the graph type (directed or undirected).
        - Subsequent lines represent edges in the format: node1 node2 weight. Example: A B 5 means an edge between A and B with weight 5.

Example File (graph.txt):

```
4

A B C D

undirected

A B 3

A C 2
```

Output:

- The **BFS Traversal Path** starting from the first node in the graph.
- The **total weight** of the graph (sum of all edge weights, considering graph type).
- A list of **unreachable nodes**, if any, from the starting node.

Requirements:

1. Read the graph details from the text file.
2. Store the graph using an adjacency matrix or adjacency list.
3. Implement the BFS algorithm for traversal.

4. Compute:
    - BFS traversal path (starting from the first node).
    - Total weight of the graph (considering the type of graph: directed/undirected).
5. Detect and list any unreachable nodes from the starting node.

Example (1) :

Input File (graph.txt):

```
5
A B C D E
directed
A B 4
A C 2
C D 7
D E 1
```

Expected Output:

```
BFS Traversal Path: A -> B -> C -> D -> E
Total Weight: 14
Unreachable Nodes: None
```

Explanation :

- Total nodes = 5 (A, B, C, D, E).
- Nodes = A, B, C, D, E.
- Type = Directed.
- Edges:
- A → B with weight 4.
- A → C with weight 2.
- C → D with weight 7.
- D → E with weight 1.

Example (2) :

Input File (graph.txt):

```
4

A B C D

undirected

A B 3

A C 2
```

Expected Output:

```
BFS Traversal Path: A -> B -> C

Total Weight: 5

Unreachable Nodes: D
```

Explanation

- Total nodes = 4 (A, B, C, D).
- Nodes = A, B, C, D.
- Type = Undirected.
- Edges:
- A to B with weight 3.
- A to C with weight 2.
- Node D is not connected to any other node.

Constraints:

1. Node names are single uppercase letters (A-Z).
2. Graph may contain up to 100 node

Submission guidelines:

1. Submit your code (in c language) by **10th Feb 2026 by 6pm.**
2. Hand in your source code electronically (do not submit .o or executable code).
3. Each student uploads only ONE copy of the assignment
4. Make sure that this code compiles and runs correctly on linux. The makefile must give the executable code the name bfs
5. Write a README file (text file, do not submit a .doc file) which contains
   - Names, section numbers, and email addresses of the student
   - Whether your code was tested on linux.
   - How to execute your program.
   - Briefly describe anything special about your submission that the instructor should take note of.

6. Place all your files under one directory with your cs number name (p1-csnumber for assignment 1, e.g. p1- CS001,
7. Tar the contents of this directory using the command
8. tar –cvf [directory_name].tar [directory_name]
9. E.g. tar -cvf p1CS001.tar p1-cs001/
10. **Any deviations from the specified format, file names, naming conventions will lead to penalty.**
11. Academic Integrity
    - We will use moss to detect plagiarism in this assignment.
    - Use the UGVLE to upload the tared file you created above.

You are expected to clearly explain any part of your code during evaluation. Marks will be awarded based on the depth and clarity of your explanations