

CS69201: Computing Lab-1
Assignment - Graphs
August 20, 2024

===== Instructions =====

1. In the case of user input assume only valid values will be passed as input.
2. You can use C or C++ as the programming language. **However, you are not allowed to use any STL libraries in C++.**
3. Regarding Submission: For each question create a separate C file. -> <rollno>_Q1.c, <rollno>_Q2.c. Create a zip file of all these C files in the name <rollno>_A4.zip and submit it to Moodle. For example, if your roll number is 24CS60R15, then your file name will be 24CS60R15_Q1.c, 24CS60R15_Q2.c and your zip file name will be 24CS60R15_A4.zip.
4. **Inputs should be taken from a file and outputs should be printed to file named output1.txt, output2.txt, etc. The input file name will be passed as a command line argument.**

=====

Question 1: Graph Traversal and Exploration

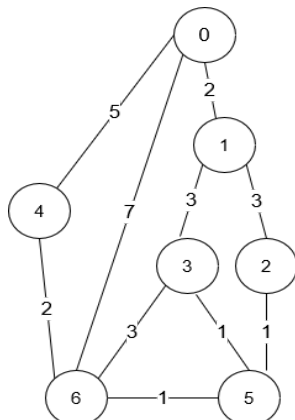
There are N houses in a city that are numbered [0,N-1]. There are bi-directional roads present between the houses. Each road takes some time to travel. Bob is at house 0 and want to reach Alice, who is at house N-1. Bob wants to know 2 things:

1. Find the weight of the smallest path that covers all the houses and print the edges in descending order of the edge weights (Output format is mentioned in the example).
2. What is the minimum amount of time required to reach Alice.
3. How many ways it is possible to reach Alice in the shortest amount of time calculated in part 2.

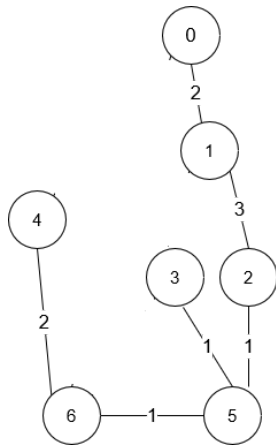
Example :

For n=7

Roads [[0,6,7],[0,1,2],[1,2,3],[1,3,3],[6,3,3],[3,5,1],[6,5,1],[2,5,1],[0,4,5],[4,6,2]]



Output-1



Total weight = 10

Output 2.

The shortest amount of time it takes to go from intersection 0 to intersection 6 is 7 minutes.

Output 3.

The four ways to get there in 7 minutes are:

- $0 \rightarrow 6$ (7)
- $0 \rightarrow 4 \rightarrow 6$ (5+2)
- $0 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 6$ (2+3+1+1)
- $0 \rightarrow 1 \rightarrow 3 \rightarrow 5 \rightarrow 6$ (2+3+1+1)

Input Format :

First line contains N, the next lines contain 3 space separated integers : a b w
Where w is the weight of edge a and b.

Output to the file(output.txt) should look like this :

Weight of the MST: 10

3 1 2

2 0 1

2 4 6

1 2 5

1 3 5

1 5 6

Shortest Path length: 7

#Possible Shortest Paths: 4

Question 2: Longest Increasing Path

Given an N*M matrix, containing integers. Your task is to find the longest path in the matrix, that is strictly increasing in nature. That is, every element that you pick next in the path should be strictly greater than the current value in the path. Kindly note that you can only travel vertically or horizontally.

<https://leetcode.com/problems/longest-increasing-path-in-a-matrix/description/>

[Hint: Trying to find the longest path from every possible cell can take a very long time. Try to store the value for each calculated cell to reduce the time complexity]

Example:

Input: matrix = [[9,9,4],[6,6,8],[2,1,1]]

9	9	4
6	6	8
2	1	1

Output: 4

Question 3: The Sneaky Pizza Delivery

You're delivering a pineapple pizza in a neighborhood, but here's the twist: the neighborhood is filled with angry Italians who think pineapple on pizza is a crime against humanity. Every step you take, they might try to intercept you and reclaim their pizza dignity. Your goal now is to take the pizza to the edge of the neighborhood without getting caught by the pizza purists. Is it possible to keep you and your controversial pizza safe?

The first input line has two integers n and m: the height and width of the neighborhood. After this there are n lines of m characters describing the map.

Each character is one of the following: . (floor), * (wall), S (start), or P (pizza purists). There is exactly one S in the input. If possible to escape, print "YES", else print "No".

Example

Input:

5 8

P..S.

*.*P*.*

P..*..

* *****

.

Output: YES [He would follow the path : Right -> Right -> Down -> Down -> Right]

Bonus for Question 3: Print the path you need to follow to escape if possible. Print the path in the next line of the output file.