



Lord of the Rings

CO222 PROJECT MILESTONE 01

OPTIMAL PATH TO MOUNT DOOM

DEPARTMENT OF COMPUTER ENGINEERING
FACULTY OF ENGINEERING
UNIVERSITY OF PERADENIYA



MILESTONE 01



Frodo Baggins, a humble hobbit from the Shire, finds himself thrust into an epic quest to destroy the One Ring. This ring, created by the Dark Lord Sauron, possesses malevolent power that corrupts and enslaves those who seek to wield it. Frodo's task is to carry the Ring to Mount Doom, located in the treacherous land of Mordor, and cast it into the fiery depths to ensure its destruction and prevent Sauron from regaining his full strength.

Frodo is not alone on this perilous journey. His loyal friend and companion, Samwise "Sam" Gamgee, accompanies him. Together, they form an unyielding bond that becomes a beacon of hope in the face of overwhelming darkness. In addition to Sam, Frodo is joined by a diverse fellowship, including the wise and powerful wizard Gandalf the Grey, the skilled ranger Aragorn, the elf Legolas, the dwarf Gimli, and others. Each member of the fellowship brings unique skills and strengths to aid Frodo in his quest.

The journey takes them through key cities and regions in Middle-earth, including the Shire, Rivendell, the Mines of Moria, Lothlórien, and Gondor. However, the ultimate destination is Mount Doom in Mordor, where the fate of Middle-earth will be decided.

To navigate this perilous journey, finding the minimum distance path on the map of Middle Earth is crucial. Your task is to assist Frodo in discovering the optimal path with the least total distance traveled. To do that, you will be given the map, and you need to find a path that gives the minimum distance from a given starting point to a given endpoint. But, when you are moving along the path, some conditions need to be met for each move. So you must find the minimum distance path according to the given conditions. Before going into the tasks and conditions, let's consider the input format.

Input Format:

As the input, we are giving 4 lines, that contain

1st line: H W

(Two integers for the height (H) and the width (W) of the map)

2nd line: S1 S2

(Two integers for the starting position - 0 Indexed, S1 row number, S2 column number)

3rd line: E1 E2

(Two integers for the ending position - 0 Indexed, E1 row number, E2 column number)

4th line: String that contains H * W characters

(For the map creation)

Sample input

5 4

0 0

4 2

ABCCDGLCKEZZBLFXHYCMGDVEGH

Your first task is to create a 2D array (As the map) from the given string. To do that, you need to divide that string into H sub-arrays, and each sub-array contains W characters. As the characters, we use "A" to "Z" (Only capital English letters), and each character gives an altitude. The altitude of a character can be calculated as the ASCII value of that character. For the sake of simplicity, each position of the array will be referred to as a cell.

After creating the 2D array, it will look like the following:

```
[  
[A B C C D],  
[G L C K E] ,  
[Z Z B L F] ,  
[X Y C M G],  
[D V E G H]  
]
```

According to the given input, the starting point is (0,0), and the ending point is (4,2). Therefore, A is the starting point, and E is the ending point.

Now your task is to find the minimum distance between the starting point and the ending point and you need to follow the conditions given below,

- 1) In a single move, you can go to the next cell, **up, down, left or right**. (No diagonal moves)
- 2) If you are in a cell with character (C1), you need to go to the new cell with character (C2). The only way you can move is $\text{ASCII}(C2) - \text{ASCII}(C1) \leq 2$ (**Climbing**) or $\text{ASCII}(C1) - \text{ASCII}(C2) \leq 3$ (**Going down**). Otherwise, you can't move to that cell.

According to the given input, there are two paths which follow the above conditions. The paths are shown in **Pink** and **Blue**

```
[
[A B C C D],
[G L C K E] ,
[Z Z B L F] ,
[X Y C M G],
[D V E G H]
]
```

Since it has only 6 moves and the blue path has 10 moves, the minimum distance path should be the pink path. Therefore, the output should be as follows

Output

The minimum distance from A to E is 6

Note that, if there is no possible path for a given input, the output should be as follows.

Output

There is no possible path from A to E

IMPORTANT

- 1) You should use **global arrays**.
- 2) There must have **at least 2 separate functions**,
 - a) For creating 2D array for a given string
 - b) For finding the minimum distance path

****You may have more than 2 functions. But at least the above-mentioned functions should be used.**

Test cases

Here are some test cases. This is **NOT** the complete set of test cases used to evaluate your submission. Therefore, you are advised to formulate your test cases (in addition to what is given here) and test your program.

Test case 0

Input

5 4
0 0
4 2
ABCCDGLCKEZZBLFXCYCMGDVEGH

Output

The minimum distance from A to E is 6

Test case 1

5 5
0 0
4 4
KKKKKMMMMMMOOOOOLLLLLIIIII

Output

The minimum distance from K to I is 8

Test case 2

6 6
5 5
4 4
AAAAAAFAFFFAAFFFAAFFBAAFAFFFAAAAAA

Output

The minimum distance from A to B is 20

Submission

Marks for milestone 1

10% of the final grade (zero marks for submissions that have compilation errors)

Deadline for milestone 1

Strictly on or before Jan 12th 2023 at 11.55 pm. Late submissions will be accepted with a 20% penalty per day they are late.

Plagiarism policy

Copied submissions (including those from the Internet) will receive zero marks. Your program must be entirely your own work. **Do not copy from others, and do not allow anyone else to see your code.**

Submission files

For milestone 1, you should create the file and name this file “co222-project-m1.c”

Note that marks will also be awarded for legible and readable code and proper comments.

Submission method will be notified later.

