

CS2203: Artificial Intelligence

DFS (Depth First Search)

Date: 14/01/2025
Deadline: 21/01/2025

Max Marks: 30

Instructions:

1. The assignment should be completed and uploaded by the deadline.
2. Please make a **detailed report in PDF format** for the assignment.
3. Markings will be based on the correctness and soundness of the outputs. Marks will be deducted in case of plagiarism.
4. Proper indentation and appropriate comments are mandatory.
5. You should zip all the required files and name the zip file as:
roll_no_of_all_group_members.zip , eg. **2301cs11_2301cs03_2321cs05.zip**.
6. Upload your assignment (**the zip file**) in the following link:
- 7.

For any queries regarding this assignment, you can contact:
Utsav Kumar (utsavkumarnareti@gmail.com)

Treasure Hunt: The Maze of Mysteries

The Adventure Begins...

You are a brave treasure hunter on a quest to find a legendary treasure hidden deep within the Maze of Mysteries. This maze is said to be cursed, with walls that shift and paths that disappear. However, an ancient scroll containing the secrets of the maze has been passed down through generations. It holds the key to navigating through this labyrinth using the ancient art of **Depth-First Search (DFS)**.

The treasure lies at the very end of the maze, marked with an **E** on your map, and your journey begins at the **S**, the starting point. Can you use the wisdom of **DFS** to uncover the treasure and find your way out of the maze?

Your Problem

Your goal is simple: **Find your way from S to E** in the maze. However, there's a twist — the maze has many dead-ends, and you must be careful not to get lost. You can only proceed along open paths (represented by **0**) and must avoid the treacherous walls (represented by **1**). To succeed, you'll need to use the magical technique of **Depth-First Search (DFS)**.

Input: First take 2 integers N and M as input. Where N represents the number of rows in the maze and M represents the number of columns in the maze. Then take N x M input to make a grid. (There should be exactly one S and one E is allowed).

Output:

- Print the number of visited cells in the maze.
- If Path exists, show the path from S to E and the length of the path.
- Else print “Oh no! You've reached a dead-end. No treasure found this time.”

Sample Input: N=4, M=5

```
maze = [  
    ['S', '0', '1', '1', '1'],  
    ['1', '0', '1', '0', '0'],  
    ['1', '0', '0', '1', '1'],  
    ['1', '1', '0', '0', 'E']]
```

Sample Output:

“number of visited cells is the maze”: 11

“Congratulations, brave adventurer! You've found the treasure! Here's the path you took:”

[(0, 0), (0, 1), (1, 1), (2, 1), (2, 2), (3, 2), (3, 3), (3, 4)]

“Path Length”:8

Questions:

- Modify the problem to find all solution paths.
- How does the density of walls affect the DFS algorithm's performance?
- How does backtracking affect the runtime for larger mazes?
- Does DFS always find the optimal path? Why or why not?