**Report**

Students: Abdulaziz Alhajri-441102528

Mohammed Aljalajil-442101136

**1. Introduction**

Our implemented project focuses on a pathfinding algorithm using dynamic programming principles. The algorithm efficiently finds the optimal path from a start point to an end point in a grid, considering obstacles and determining the minimum cost. The primary components include a grid representation, obstacle marking, and an interactive user interface for input/output.

**2. Algorithm Explanation**

**2.1 Grid Representation**

The grid is represented as an **n x m** matrix initialized with zero values. The start and end points are marked within the grid, and obstacles are represented by **-1**.

**2.2 Dynamic Programming Principles**

The algorithm utilizes dynamic programming principles to determine the optimal path and minimum cost efficiently. Key steps include:

**2.2.1 Initialization**

A 2D table (**dp**) is initialized with all values set to positive infinity except for the starting point, where the cost is set to zero.

**2.2.2 Cost Calculation**

The algorithm iterates through the grid, considering valid neighbors and updating the costs based on the minimum path. The cost of moving to any neighboring cell is assumed to be equal in our case, 1.

**2.2.3 Path Reconstruction**

The optimal path is reconstructed by backtracking from the end point to the start point. At each step, the algorithm selects the neighbor with the minimum cost.

**2.3 Path Reconstruction**

The optimal path is reconstructed by backtracking from the end point to the start point, considering valid neighbors and avoiding obstacles. The path is stored in a list, and at the end of the reconstruction, it is reversed to represent the correct order from start to end.

**3. Implementation Details**

**3.1 Code Structure**

The code is organized into a **Grid** class representing the grid and methods for obstacle marking, start/end point setup, and pathfinding. The **terminal\_interface** method facilitates user interaction to set up the grid and find the optimal path.

**3.2 User Interface**

The user interface prompts users to input the grid dimensions, start and end points, and obstacle locations. It then visualizes the original grid, finds the optimal path, and displays the results.

**4. Experimental Results**

**4.1 Test Scenarios**

The algorithm has been tested in various scenarios:

* Valid paths with obstacles
* Valid paths without obstacles
* Grids of different sizes
* Scenarios where no path is possible

**4.2 Observations**

The algorithm performs effectively, accurately finding the optimal path and providing the minimum cost. It handles various scenarios, including edge cases, with the expected termination messages when no valid path is found.

**5. Time and Space Complexity Analysis**

**5.1 Time Complexity**

The time complexity of the algorithm is O(n \* m), where n is the number of rows and m is the number of columns. This is because the algorithm iterates through each cell in the grid once.

**5.2 Space Complexity**

The space complexity is O(n \* m) as well. The dynamic programming table (**dp**) requires space proportional to the size of the grid.

**6. Conclusion**

The implemented pathfinding algorithm demonstrates correctness and efficiency. Dynamic programming principles ensure optimal path determination. The user interface facilitates easy interaction and visualization of results. Code quality, documentation, and handling of edge cases contribute to the robustness of the solution.