Project Proposal: Maze Generator and Solver

Group 20 : Abdullah Masood & Arshman Ali

December 21, 2025

Introduction

The goal of this project is to design an interactive **Maze Generator and Solver** that allows users to visualize maze creation and solution processes. The project aims to demonstrate graph algorithms in action and provide insights into algorithmic efficiency.

Objectives

- Implement maze generation using Randomized Depth-First Search (DFS) and Prim's algorithm.
- Solve mazes using BFS (shortest path) and DFS (exploratory path).
- Support multiple maze types and difficulty levels.
- Provide step-by-step visualization of both maze generation and solving processes.
- Compare path lengths obtained from BFS and DFS solutions.
- Enable export and import of maze configurations for reuse or analysis.

Methodology

Maze Generation

- Represent the maze as a 2D grid with walls and paths.
- Use Randomized DFS to carve paths recursively from a starting cell.
- Use Prim's algorithm as an alternative method to generate mazes with different characteristics.
- Allow configuration of maze size and difficulty.

Maze Solving

- Implement BFS to find the shortest path from start to end.
- Implement DFS to explore the maze and find a solution path (may not be shortest).
- Visualize the exploration process by updating the display at each step.
- Highlight the final solution path once the maze is solved.

Visualization and Interaction

- Display the maze in a graphical or ASCII-based interface.
- Animate the step-by-step generation and solving process.
- Compare BFS and DFS paths in terms of length and efficiency.
- Support saving and loading maze configurations for experimentation.

Expected Outcomes

- An interactive application to generate and solve mazes with visual feedback.
- Side-by-side comparison of BFS and DFS paths.
- Insight into algorithmic differences between BFS and DFS in maze solving.
- Ability to test mazes of different sizes and difficulty levels.
- Exportable maze data for further analysis or sharing.

Timeline

Day	**Task**
1–2	Research maze generation and graph algorithms
3–4	Implement Randomized DFS and Prim's algorithm for maze generation
5–6	Implement BFS and DFS maze solvers with step-by-step visualization
7	Add maze difficulty settings and multiple maze types
8	Implement maze export/import feature
9	Testing, debugging, and performance evaluation
10	Documentation and final report preparation