Proof of the Theoretical Complexity of the Maze Resolution Algorithm

Introduction

In this report, we provide a proof of the theoretical complexity of our maze resolution algorithm. We have implemented the Breadth-First Search (BFS) algorithm to find the shortest path from the entrance to the exit in a maze. We will demonstrate that the time complexity of this algorithm is linear with respect to the number of cells in the maze, specifically $O(V + E)$, where V is the number of vertices (cells) and E is the number of edges (open paths between cells).

Maze as a Graph

A maze can be represented as an undirected graph $G = (V, E)$, where:

- Each cell in the maze corresponds to a vertex $v \in V$.
- An edge $e \in E$ exists between two vertices if the corresponding cells are adjacent and there is no wall between them.

Since each cell can have up to four neighbors (North, East, South, West), the maximum degree of any vertex is 4.

Breadth-First Search (BFS) Algorithm

BFS explores the graph level by level, starting from the entrance cell. It uses a queue to keep track of the next vertices to visit and marks each visited vertex to avoid revisiting.

Algorithm Steps

1. Initialize a queue Q and enqueue the entrance cell.
2. Mark the entrance cell as visited.
3. While Q is not empty:
   (a) Dequeue a cell u from Q.
   (b) For each unvisited neighbor v of u:
      i. Mark v as visited.
      ii. Set the predecessor of v to u (for path reconstruction).
      iii. Enqueue v into Q.

Complexity Analysis

Time Complexity

The time complexity of BFS on a graph $G = (V, E)$ is $O(V + E)$. This is because:

- Each vertex $v \in V$ is enqueued and dequeued at most once.
- The algorithm examines all edges $e \in E$ adjacent to each vertex.

Detailed Proof

- **Initialization:** Enqueuing the entrance cell takes constant time $O(1)$.
- **Processing Vertices:** Each vertex is dequeued exactly once. Since there are V vertices, the total time for dequeuing is $O(V)$.
- **Exploring Edges:** For each dequeued vertex u, we examine its adjacent edges. The sum of the degrees of all vertices is $2E$ (by the Handshaking Lemma for undirected graphs). Therefore, the total time spent exploring edges is $O(E)$.

Thus, the overall time complexity is $O(V + E)$.

Space Complexity

The space complexity of BFS is $O(V)$, due to:

- The queue Q can hold at most V vertices.
- The visited array or marker requires $O(V)$ space.
- The predecessor array for path reconstruction also requires $O(V)$ space.

Application to the Maze

In the context of the maze:

- $V = h \times w$, where h is the height and w is the width of the maze.
- Each cell (vertex) has up to 4 edges (to its North, East, South, and West neighbors if there are no walls).
- The number of edges E is less than or equal to $2V$, since each cell can have up to 4 edges, but each edge is shared between two cells.

Therefore, both V and E are proportional to the number of cells in the maze.

Time Complexity in the Maze

Since $E = O(V)$ in a maze, the time complexity simplifies to $O(V)$.

Conclusion

Our BFS-based maze resolution algorithm has a time complexity of $O(V)$, where V is the number of cells in the maze. This linear complexity ensures that the algorithm is efficient and scalable for mazes of varying sizes.