Proof of the Theoretical Complexity of the Maze Resolution Algorithm

For our maze resolution algorithm, we represent the maze as a graph where each cell is a node connected to its accessible neighbors. We've chosen a Breadth-First Search (BFS) algorithm to ensure that the shortest path between the start and the exit is found.

Algorithm Description:

- The BFS algorithm explores the maze level by level, starting from the initial cell and gradually adding accessible neighboring cells to a queue.
-Each cell is visited at most once, and unvisited neighboring cells are added to the queue for future exploration.

Time Complexity:

-Let n be the number of cells in the maze.
-In the worst case, each cell is visited only once, and each edge (possible path between two cells) is checked at least once.
-Therefore, the time complexity of the BFS algorithm in a graph with n nodes and m edges is

$$O(n+m).$$

-In a typical maze, m is at most proportional to 4n (since each cell can have up to four neighbors), so the complexity simplifies to O(n) because m is bounded by a constant multiple of n.

Space Complexity:

-The algorithm uses a queue to store cells that are being explored. In the worst case (when the maze is wide and shallow), this queue can contain up to O(n) cells at once.
-Additionally, a list of size O(n) is used to mark visited cells to avoid revisits.
-Overall, the space complexity is thus O(n).

Edge Cases:

-If the maze is a long, straight line without branches, BFS will traverse all cells in the line, resulting in a time complexity of O(n).
-If the maze is entirely open, BFS will inspect all cells to reach the exit, also resulting in a complexity of O(n).
-Therefore, BFS remains efficient even in these cases, as it guarantees the optimal solution (shortest path) with a complexity of O(n).

Conclusion:

-The theoretical complexity of our BFS maze resolution algorithm is therefore O(n) in both time and space.
-This algorithm is efficient and guarantees an optimal solution for any maze represented as an unweighted graph, ensuring thorough exploration while minimizing redundant cell visits.