1. Why do the stack and queue versions output different paths for some mazes?
The stack-based algorithm (DFS) explores paths in a Last-In-First-Out (LIFO) order, prioritizing deeper exploration of one branch before backtracking. The queue-based algorithm (BFS) explores paths in a First-In-First-Out (FIFO) order, expanding all neighbors at the current depth before moving deeper. This difference in exploration order can lead to different paths: DFS may find a longer, winding path, while BFS guarantees the shortest path in unweighted grids.

---

2. Which implementation explores one path at a time?
The **stack-based (DFS)** implementation explores one path at a time. It follows a single branch as far as possible before backtracking to explore alternative routes.

---

3. Which implementation explores all possible paths at the same time?
The **queue-based (BFS)** implementation explores all possible paths "simultaneously" by processing nodes layer by layer. At each step, it explores all valid neighbors of the current node before moving to the next depth level.

---

4. Which implementation finds the shortest path? Is this guaranteed for all input mazes?
The **queue-based (BFS)** implementation finds the shortest path. This is guaranteed for all solvable mazes because BFS systematically explores all nodes at distance d before nodes at distance d+1, ensuring the first path found to the exit is the shortest.

---

5. Example Maze with Different Paths
Consider the following 4x4 maze:

```
Copy
0 0 1 0
1 0 1 0
0 0 1 0
0 1 0 0
```

- **Entrance**: [0][0], **Exit**: [3][3].
- **BFS Path**: (0,0) → (0,1) → (1,1) → (2,1) → (2,0) → (3,0) → (3,1) → (3,2) → (3,3) (shortest path).
- **DFS Path**: (0,0) → (1,0) → (2,0) → (3,0) → (3,1) → (2,1) → (1,1) → (0,1) → (0,2) → ... (longer path due to backtracking).

Explanation: BFS finds the shortest path by exploring layers, while DFS takes a longer route due to its depth-first nature and neighbor order (Down, Up, Right, Left).

BFS Path (Shortest):

X X 1 0
1 X 1 0
X X 1 0
X X X X
DFS Path (Longer):
X X 1 0
X 0 1 0
X X 1 0
X 1 0 X

This example demonstrates how BFS and DFS diverge based on their exploration strategies.