Assignment-1
Artificial Intelligence Lab

Problem Description:

You are tasked with designing a pathfinding system for a robot navigating through a maze. The maze is represented as a grid where each cell can either be an open space (denoted by **0**) or a blocked space (denoted by **1**).
The robot starts at a specific location, and its goal is to find the **shortest path** to a destination point while avoiding obstacles.

Unlike A*, **Dijkstra's algorithm does not use a heuristic**. It explores nodes in order of their shortest known distance from the start.

Input:

- A 2D grid representing the maze where:
  - **0** → open cell
  - **1** → blocked cell (impassable)
- The robot's starting position: **(start_x, start_y)**
- The robot's destination position: **(end_x, end_y)**
- The robot can move **up, down, left, right** only.

Task:

Use **Dijkstra’s algorithm** to find the shortest path from the starting cell to the destination cell.
If no path exists, return **None**.

Requirements:

- Implement **Dijkstra’s algorithm**, using a priority queue to always expand the node with the smallest distance.
- Treat each step from one cell to an adjacent cell as having cost **1**.
- Avoid all blocked cells.
- Output the **sequence of grid coordinates** representing the shortest path.
- If no valid path exists, return **None**.

Example:

Input:

Maze grid (5×5):

```
0 0 1 0 0

0 1 1 0 0

0 0 0 0 1

1 1 0 1 0

0 0 0 0 0
```

Starting position: **(0, 0)** (top-left)
Destination position: **(4, 4)** (bottom-right)
Going down increases **y**
Going right increases **x**

Output:

```
Path found: [(0, 0), (0, 1), (0, 2), (1, 2), (2, 2), (2, 3), (2,
4), (3, 4), (4, 4)]
```

Explanation:

Dijkstra’s algorithm explores the grid by expanding the node with the **smallest known distance from the start**, without any heuristic.

- Every move costs **1**.
- Distances accumulate as the robot moves.
- The algorithm keeps track of the shortest distance to every reachable cell.
- When the destination cell is removed from the priority queue, the shortest path has been found.
- If the destination is unreachable, the algorithm returns **None**.