Question: Pathfinding in a Maze

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 in the maze while avoiding obstacles.

Input:

- A 2D grid representing the maze where:
  - `0` indicates an open space.
  - `1` indicates a blocked space (impassable).
- The starting position of the robot `(start_x, start_y)` in the grid.
- The destination position `(end_x, end_y)` in the grid.
- The robot can move in four directions: up, down, left, and right.

Task:

Use the *A search algorithm** to find the shortest path from the starting point to the destination point. If no path exists, return `None`.

Requirements:

1. Implement the A* search algorithm to find the shortest path.
2. Define an appropriate **heuristic function** (e.g., Manhattan distance or Euclidean distance) to estimate the cost from any given position to the destination.
3. Output the sequence of grid coordinates representing the path from the start to the destination.
4. If no path exists, return `None`.

Example:

Input:

Maze grid (5x5):

```
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:

- The A* algorithm will evaluate nodes (grid cells) based on the combination of the **g-cost** (the cost of moving from the start to the current node) and the **h-cost** (the estimated cost to reach the goal from the current node, calculated using the heuristic function).
- The algorithm will prioritize nodes that minimize the total cost (g + h).
- If the destination is reachable, A* will output the shortest path from start to end. If no path exists, it will return `None`.