AI Assignment 2

You are given an `n × n` binary matrix `grid`. Your task is to implement and compare two search algorithms to find a path from the top-left cell `(0, 0)` to the bottom-right cell `(n - 1, n - 1)`.

A clear path is defined as:

1. **All visited cells along the path must have a value of `0`.**
2. **Moves can be made 8-directionally — i.e., from a cell, you may move to another cell that is horizontally, vertically, or diagonally adjacent.**
3. **The length of the path is the total number of visited cells.**

Part A — Best First Search (Greedy Search)

- **Implement Best First Search using an admissible heuristic (e.g., Euclidean or Manhattan distance to the goal).**
- **Note: This approach may not always return the shortest path.**
- **Record the path found and its length.**

Part B — A* Search

- **Implement A* search using the same heuristic.**
- **This approach should return the shortest path length, or `-1` if no path exists.**

Input / Output Requirements

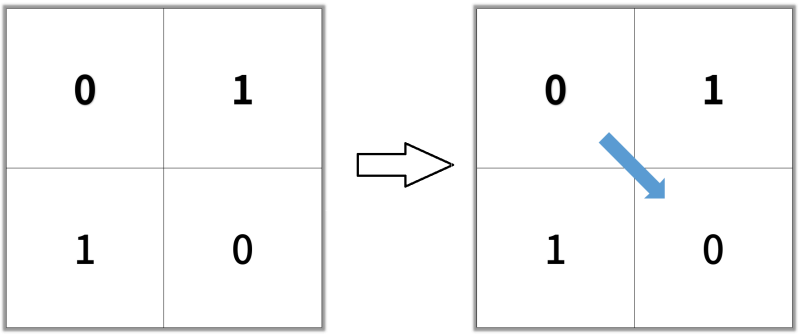
- **Input: an `n × n` binary matrix `grid`.**
- **Output: For each algorithm, print:**
    1. **The sequence of coordinates visited (path).**
    2. **The length of the path (or `-1` if no path exists).**

Example 1:

Input:

grid = [[0, 1],

[1, 0]]



Output (example format):

Best First Search → Path length: 2, Path: [(0,0), (1,1)]

A* Search → Path length: 2, Path: [(0,0), (1,1)]

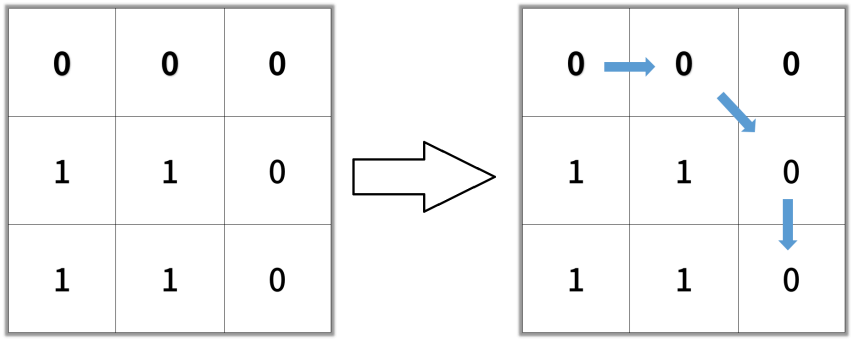
Example 2:

Input:

grid = [[0, 0, 0],

[1, 1, 0],

[1, 1, 0]]



Output:

Best First Search → Path length: 4, Path: [(0,0), (0,1), (1,2), (2,2)]

A* Search → Path length: 4, Path: [(0,0), (0,1), (1,2), (2,2)]

Example 3:

Input:

grid = [[1, 0, 0],

[1, 1, 0],

[1, 1, 0]]

Output:

Best First Search → Path length: -1

A* Search → Path length: -1

Constraints:

- `n == grid.length`
- `n == grid[i].length`
- `1 ≤ n ≤ 100`
- `grid[i][j] ∈ {0, 1}`

Deliverables:

1. **Implementation of Best First Search and A* Search.**
2. **Output for given test cases.**
3. **A short comparison (1–2 paragraphs) discussing differences in results and performance.**

Optional (Ungraded):
Develop a visualization for the problem similar to:

- **A* Visualizer Example 1**
  - https://astar-visualizer.vercel.app/
- **Pathfinding Visualizer Example 2**
  - https://pathfinding-visualizer.researchdatapod.com/

- **Submission Deadline**
  - **22/08/2025**
  - **Upload on your Github account**