Assignment: Pathfinding with A*

Problem Statement

We want to find the shortest path for a robot navigating a **10×10 grid** from a start cell to a goal cell using the **A*** algorithm. The grid contains obstacles and varying movement costs. You will implement A* with a consistent heuristic (Manhattan distance) and test it on a sample problem.

Tasks

1. Heuristic Consistency

- Implement the **Manhattan distance heuristic** for grid-based navigation.
- Verify whether it satisfies the **consistency (monotonicity)** property.
- Add a short explanation in comments about why consistency is crucial for A* (it guarantees optimality and avoids node re-expansions).

---

2. Algorithm Implementation

- Implement **standard A*** for grid-based pathfinding.
- Maintain:
  - Open and closed lists.
  - `f(n) = g(n) + h(n)` where `g(n)` is the path cost so far and `h(n)` is the heuristic.
- Output the computed path and path cost.

---

3. Sample Grid Problem

Use the following **10×10 grid**:

```
S . . . . . . . . . .
. . . # # # # . . .
. . . . . . . . . .
. # # # # . . . . .
. . . . . . . . . .
. . . . . . . . . .
. . . . # # # # . .
. . . . . . . . . .
. . . . . . . . . .
. . . . . . . . . G
```

- Start = **S (0,0)**
- Goal = **G (9,9)**
- Obstacles = # (row 3, cols 2–5; row 6, cols 4–7)
- Rough terrain = row 5 cells, cost = 3
- All other cells, cost = 1

---

4. Experimentation

1. Run A* on the initial grid and find the path from **S** to **G**.
2. Record:
    - Path sequence

    - Path cost
    - Number of nodes expanded
3. Modify the grid by **blocking cell (4,4)**.
4. Rerun A* and record the new path, cost, and expanded nodes.
5. Compare the paths before and after the change.

---

5. Submission Requirements

- Python code with clear comments.
- Example run showing:
    - Initial grid and computed path.
    - Updated grid and recomputed path after adding the new obstacle.
- A short explanation (in comments or text file) about:
    - Why A* finds the optimal path.
    - Why heuristic consistency matters.