Homework 2: Robot Navigation in a Dynamic Environment

Artificial Intelligence Course

Description

An autonomous robot needs to navigate through a warehouse grid from a starting position to a destination efficiently, avoiding both static and dynamic obstacles. The warehouse is represented as a two-dimensional grid of cells. Each cell can be empty, contain a static obstacle (e.g., shelves), or be occupied temporarily by a dynamic obstacle (e.g., moving carts or other robots).

The robot has limited battery life, which imposes a constraint on the maximum number of moves it can make. The goal is to find a path from the starting position to the destination that minimizes the total cost, considering the battery constraints and avoiding collisions with obstacles.

The robot receives real-time updates on the positions of dynamic obstacles at each time step.

Tasks

Implement the following search algorithms to solve the robot navigation problem:

Tasks

Implement the following search algorithms to solve the robot navigation problem:

1. A* Search
2. Hill Climbing

3. Simulated Annealing

4. Tabu Search

For each algorithm, your implementation should find a path from the starting position to the destination, if one exists within the battery constraints.

Input Format

The input will be provided in a text file with the following format:

- The first line contains two integers N and M, representing the number of rows and columns in the grid.
- The second line contains two integers B and T, representing the robot's battery life (maximum number of moves) and the total number of time steps.
- The third line contains four integers S_x, S_y, D_x, and D_y, representing the starting and destination positions of the robot.
- The next N lines each contain M characters, representing the static grid layout:
  - `.` represents an empty cell.
  - `#` represents a static obstacle.
- The following T sections each represent the positions of dynamic obstacles at each time step:
  - Each section starts with a line containing the integer K, the number of dynamic obstacles at that time step.
  - The next K lines each contain two integers X and Y, representing the position of a dynamic obstacle.

Output Format

Your program should output the path found by each algorithm in the following format:

- The first line should contain the name of the algorithm.
- The second line should contain an integer L, the length of the path.
- The next L lines should contain two integers each, representing the positions in the path in order.
- If no path is found within the battery constraints, output `No path found`.

Example

Input

```
5 5
15 3
0 0 4 4
.#...
..#..
...#.
.....
.....
2
1 2
3 3
1
2 2
0
```

Explanation

- Grid size: 5 rows $\times$ 5 columns.
- Battery life: 15 moves.

- Time steps: 3.
- Starting position: $(0, 0)$.
- Destination position: $(4, 4)$.
- Static obstacles are at positions where the grid has `#`.
- At time step 1, dynamic obstacles are at positions $(1, 2)$ and $(3, 3)$.
- At time step 2, dynamic obstacle at position $(2, 2)$.
- At time step 3, no dynamic obstacles.

Output

```
A* Search
9
0 0
1 0
2 0
3 0
4 0
4 1
4 2
4 3
4 4
Hill Climbing
No path found
Simulated Annealing
10
0 0
0 1
0 2
1 2
2 2
3 2
4 2
4 3
4 4
```

```
Tabu Search
8
0 0
1 0
2 0
3 0
3 1
3 2
4 3
4 4
```

Instructions

- Implement each algorithm separately and ensure that your code is well-documented.
- Consider the robot's battery life in your implementations.
- Avoid collisions with both static and dynamic obstacles.
- You may assume that the robot moves one cell at a time in the four cardinal directions (up, down, left, right).
- Submit your code along with a brief report explaining your implementation and any assumptions made.
- The output provided in the example is not necessarily the best or only possible output.