

AI LAB TASK 7

Write code for A* Algorithm

Node Class:

Represents a node with position $((x, y))$, `g_cost`, `h_cost`, `f_cost`, and a reference to its parent node for path reconstruction.

The `__lt__` method allows nodes to be compared by their `f_cost`, so that the node with the lowest `f_cost` is dequeued first.

*A Algorithm** (`astar_algorithm` function):

Uses a priority queue (`open_list`) implemented by a heap (`heapq`) to always process the node with the lowest `f_cost`.

If the goal is reached, it reconstructs the path by following the parent pointers from the goal node to the start node.

`closed_list` is used to track nodes that have already been evaluated.

Heuristic (`heuristic` function):

In this example, we use the **Manhattan distance** heuristic, which is suitable for grid-based pathfinding where movements are restricted to horizontal and vertical directions.

Neighbors (`get_neighbors` function):

Generates the neighbors of a given node by checking adjacent grid cells (up, down, left, right).

Ensures the neighbor is within bounds and not an obstacle (represented by 1).

Reconstruct Path (`reconstruct_path` function):

Reconstructs the path from the goal to the start by following the `parent` references.

OUTPUT:



The screenshot shows a Python IDE interface with a terminal window. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is active), and PORTS. The terminal content shows a PowerShell prompt running a Python script, which outputs a list of coordinate pairs.

```
PS C:\Users\Usman Ghani\Desktop\myworld> & "C:/Program Files/Python312/python.exe" "c:/Users/Usman Ghani/Desktop/myworld/AI_LAB_Task_7.py"
Path found: [(0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (4, 1), (4, 2), (4, 3), (4, 4)]
PS C:\Users\Usman Ghani\Desktop\myworld>
```