Aim

To implement the A\* (A-star) pathfinding algorithm in Python to find the shortest path between a start and goal point in a 2D grid, avoiding obstacles.

Algorithm

A\* is a best-first search algorithm that finds the shortest path by combining the cost to reach a node (G) and a heuristic estimate to the goal (H). It uses the function:

f(n)=g(n)+h(n)f(n) = g(n) + h(n)f(n)=g(n)+h(n)

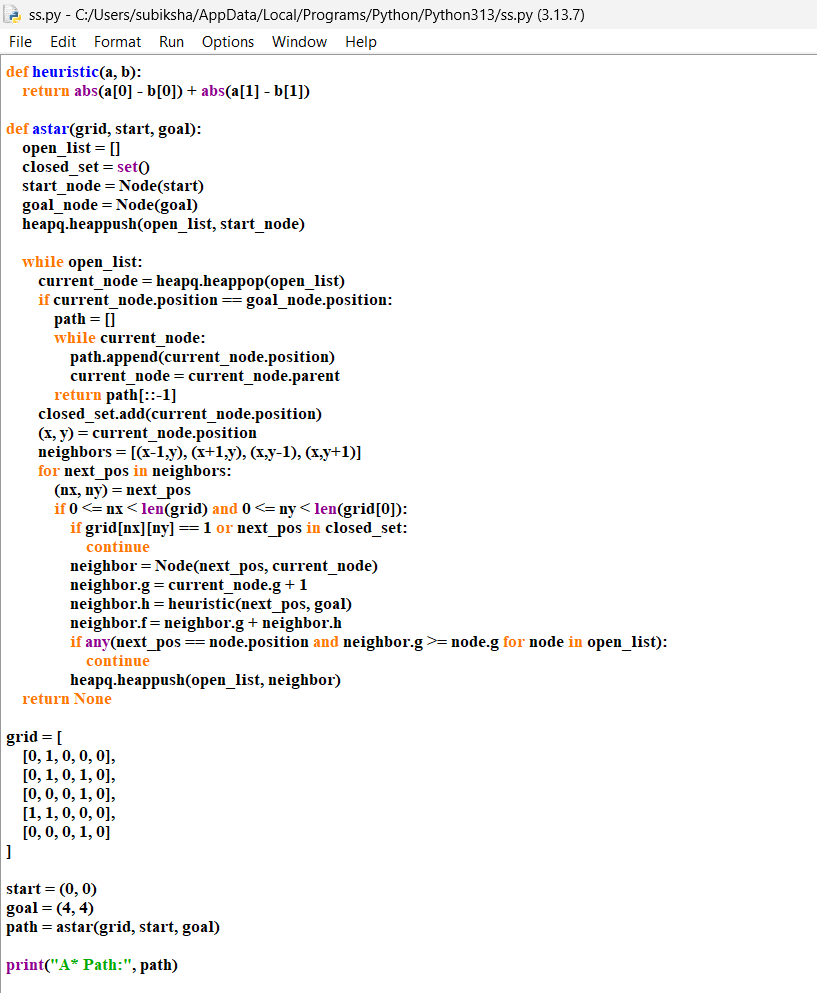
Where:

* g(n) = cost from start node to current node n
* h(n) = estimated cost from node n to goal (heuristic)
* f(n) = total estimated cost of the path through n

Steps:

1. Initialize the open list with the start node.
2. Loop until the open list is empty:
   * Pick the node with the lowest f value.
   * If it's the goal, reconstruct and return the path.
   * Move it to the closed list.
   * For each neighbor:
     + If it's an obstacle or already evaluated, skip.
     + If new or has a better g cost, update and add to open list.
3. If the goal is not reached, return None.

CODE



Result :

