Date 02/03/2024

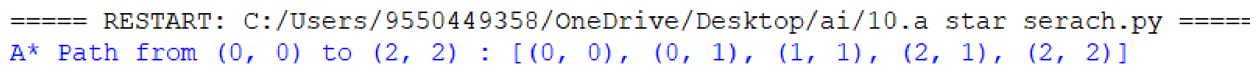
10.Aim: Write the python program to implement A* algorithm

Program:

```
import heapq


class Graph:
    def __init__(self):
        self.graph = {}

    def add_edge(self, u, v, weight):
        if u not in self.graph:
            self.graph[u] = []
        self.graph[u].append((v, weight))

    def astar(self, start, goal):
        open_list = []
        closed_list = set()

        heapq.heappush(open_list, (0, start))
        g_scores = {node: float('inf') for node in self.graph}
        g_scores[start] = 0
        f_scores = {node: float('inf') for node in self.graph}
        f_scores[start] = self.heuristic(start, goal)

        while open_list:
            current_cost, current_node = heapq.heappop(open_list)

            if current_node == goal:
                return self.reconstruct_path(start, goal)
            closed_list.add(current_node)

            for neighbor, weight in self.graph[current_node]:
                if neighbor in closed_list:
                    continue

                tentative_g_score = g_scores[current_node] + weight
                if tentative_g_score < g_scores[neighbor]:
                    g_scores[neighbor] = tentative_g_score
                    f_scores[neighbor] = tentative_g_score + self.heuristic(neighbor, goal)
                    heapq.heappush(open_list, (f_scores[neighbor], neighbor))

        return None

    def heuristic(self, node, goal):
        # This heuristic function can be replaced with any other admissible heuristic
        return abs(node[0] - goal[0]) + abs(node[1] - goal[1])

    def reconstruct_path(self, start, goal):
        current = goal
        path = [current]
        while current != start:
            current = self.came_from[current]
            path.append(current)
        path.reverse()
        return path

# Example usage:
```

```
if __name__ == "__main__":
    graph = Graph()
    graph.add_edge((0, 0), (0, 1), 1)
    graph.add_edge((0, 0), (1, 0), 1)
    graph.add_edge((0, 1), (1, 1), 1)
    graph.add_edge((1, 0), (1, 1), 1)
    graph.add_edge((1, 0), (2, 0), 1)
    graph.add_edge((1, 1), (2, 1), 1)
    graph.add_edge((2, 0), (2, 1), 1)
    graph.add_edge((2, 1), (2, 2), 1)
    graph.add_edge((2, 2), (1, 2), 1)
    graph.add_edge((1, 2), (0, 2), 1)

    start = (0, 0)
    goal = (2, 2)
    path = graph.astar(start, goal)
    print("A* Path from", start, "to", goal, ":", path)
```

Output:



```
===== RESTART: C:/Users/9550449358/OneDrive/Desktop/ai/10.a star serach.py =====
A* Path from (0, 0) to (2, 2) : [(0, 0), (0, 1), (1, 1), (2, 1), (2, 2)]
```

Result: The given program has been executed successfully