Question:11

A* SEARCH ALGORITHM

AIM

To implement A* Search algorithm using Python

ALGORITHM

1. Heuristic Function: Estimates the cost from a node to the goal.
2. A* Search:
   - Initialize a priority queue with the start node and dictionaries to track costs and paths.
   - While the priority queue is not empty:
     - Pop the node with the lowest combined cost (actual cost + heuristic) from the priority queue.
     - If the popped node is the goal, stop and reconstruct the path.
     - For each neighbor of the current node:
       - Calculate the new cost to reach that neighbor.
       - If it's a new node or the new cost is lower, update the cost and add it to the priority queue.
       - Record the current node as the parent of the neighbor.
3. Path Reconstruction: Reconstruct the path from start to goal using the recorded parent nodes.
4. Main Function: Define the graph, start, and goal nodes, then call the A* search function and print the path found.

CODE

```
import heapq
def heuristic(node, goal):
    return abs(node[0] - goal[0]) + abs(node[1] - goal[1])

def astar(graph, start, goal):
    frontier, came_from, cost_so_far = [(0, start)], {start: None}, {start: 0}
    while frontier:
        _, current = heapq.heappop(frontier)
        if current == goal: break
        for next_node in graph[current]:
            new_cost = cost_so_far[current] + graph[current][next_node]
            if next_node not in cost_so_far or new_cost < cost_so_far[next_node]:
                cost_so_far[next_node] = new_cost
                heapq.heappush(frontier, (new_cost + heuristic(next_node, goal), next_node))
```

```
                came_from[next_node] = current
    path, current = [], goal
    while current != start: path.append(current); current = came_from[current]
    return path + [start]

if __name__ == "__main__":
    graph = {(0, 0): {(0, 1): 1, (1, 0): 1}, (0, 1): {(0, 0): 1, (1, 1): 1},
             (1, 0): {(0, 0): 1, (1, 1): 1}, (1, 1): {(0, 1): 1, (1, 0): 1}}
    start, goal = (0, 0), (1, 1)
    print("Path found:", astar(graph, start, goal))
```

OUTPUT

```
= RESTART: C:\Users\Saaniya\Downloads\ai\11.py
Path found: [(1, 1), (0, 1), (0, 0)]
```