A* Search

AIM:

To implement A* search algorithm

Program:

```
import heapq

def a_star(start, goal, grid):
    directions = [(0, 1), (0, -1), (1, 0), (-1, 0)]
    open_list = []
    closed_list = set()

    heapq.heappush(open_list, (0, start, 0, None))

    while open_list:
        f_cost, current, g_cost, parent = heapq.heappop(open_list)

        if current == goal:
            path = []
            while parent:
                path.append(current)
                current, parent = parent
            return path[::-1]

        closed_list.add(current)

        for direction in directions:
            neighbor = (current[0] + direction[0], current[1] + direction[1])

            if (0 <= neighbor[0] < len(grid) and 0 <= neighbor[1] < len(grid[0]) and
                grid[neighbor[0]][neighbor[1]] != 1 and neighbor not in closed_list):

                new_g_cost = g_cost + 1
                h_cost = abs(neighbor[0] - goal[0]) + abs(neighbor[1] - goal[1])
                heapq.heappush(open_list, (new_g_cost + h_cost, neighbor, new_g_cost, (current, parent)))

    return None


grid = [
    [0, 0, 0, 0, 0],
    [0, 1, 0, 1, 0],
    [0, 1, 0, 0, 0],
    [0, 0, 0, 1, 0],
    [0, 0, 0, 0, 0]
]
start = (0, 0)
```

Output:

```
Path found: [(0, 1), (0, 2), (0, 3), (0, 4), (1, 4), (2, 4), (3, 4), (4, 4)]
```

Result:

The problem is solved and the output is verified