Ex no : 4 A* Search Algorithm.

AIM:
To implement A* search algorithm technique to find path and traverse graphs.

PROGRAM:

```
import math
import heapq
class cell
def __init__(self):
    open_set = []
heapq.heappush(open_set, (oth(start),
                          0, start))
came_from = {}
g_score = {start : 0}
f_score = {start : h(start)}
while open_set :
    _, current = g, current = heapq.
                  heap pop (open_set)
if current == goal:
    path = []
```

```
while current in came_from:
    path.append(current)
    current = came_from[current]
    path.append(start)
    return path[::-1]
for neighbor in neighbors(current):
    tentative_g = g_score[current]+1
    if neighbor not in g_score or tentative_g
                    < g_score[neighbor]:
        came_from[neighbor] = current
        g_score[neighbor] = tentative_g
        f_score[neighbor] = tentative_g + h
                            (neighbor
if neighbor not in [i[2] for i in open_
                                set]:
    heapq.heappush(open_set)
    return None:
def heuristic(node):
    goal_position = (5,5)
return abs(node[0] - goal_position[0]) +
        abs(node[1] - goal_position[1]).
def neighbor(node):
    x, y = node.
```

return [(x+1, y), (x-1, y), (x, y+1), (x, y-1)]

start = (0, 0)

goal (5, 5).

path = a-star (start, goal, heuristic, neighbors).

print (path).

Output:

[(0, 0) (1, 0) (2, 0) (3, 0) (4, 0) (5, 0)
(5, 1) (5, 2) (5, 3) (5, 4) (5, 5)].

Result:

The program is successfully executed and output is verified.