Exp. No: 4

A* search

Date:

Aim:

To find the shortest path from a start node to a goal node using the A* search algorithm.

Algorithm:

Step 1: Create open & closed sets: start with the initial node.

Step 2: Add the start node to the open set with an initial cost of 0.

Step 3: If the remove the node with the lowest value from the open set.

Step 4: If the current node is the goal node, reconstruct the path.

Step 5: For each neighbour, calculate g, h, & f values.

Step 6: If the neighbour is not in the open set or a lower cost path is found, update costs & parent.

Step 7: Add the neighbour to the open set if it is not already in the closed set.

Step 8: Repeat until the open set is empty or the goal is found.

Program:

```
import heapq
def a_star (star, goal, h, neighbours) :
open_set = [ ]
heapq. heappush (open_set. (0+ h (start), 0, start))
 came_from = { }
 g_score = {start : 0 }
 f_score = {start : h (start) }
 while open_set :
  - current_g, current = heapq. heappop
  (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 neigbour in neighbour (current)
  tentative_g = g_score [current] + 1
```

```
if neighbour not in g-score or tentive-g <
  g-score [neighbour]:
    Came - from [neighbour] = current
      g-score [neighbour] = tentative - g
       f-score [neighbour] = tentative - g + h (neighbour)
 if neighbour not in [i[1] for i in open-set]:
  heapq. heappush (open-set, (f. score [neighbour],
  tentative - g. neighbour) )
 return None
 def heuristic (node):
   goal - position = (5,5)
 return abs (node [0] - goal - position [0]) +
   abs (node [1] - goal - position [1]))
 def neighbour (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 - start (start. goal. heuristic, neighbours)
  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 :

Thus the A* search program is executed and the output is verified successfully.

[illegible]