Exp. No: 4

Date:

A* Search

```
def a-star (grid, start, goal):
    def heuristic (a,b):
    return abs (a[0]- b[0]) + abs (a[1]- b[1])

rows, cols = len (grid), len (grid[0])
    open_list = [(0 + heuristic (start, goal), 0, start)]
     came_from = { }
    cost_so_far = {start: 0}
    while open_list:
        _, current_cost, current = heappop (open_list)
        if current == goal:
            path = [ ]
            while current in came_from:
                path. append (current)
                current = came_from [current]
            path. append (start)
            return path [::-1]
    for dx, dy in [(-1,0), (1,0), (0,-1), (0,1)]:
        neighbour = (current [0] + dx, current [1] + dy)
if [0 <= neighbour [0] < rows and 0 <= neighbour
                                    [1] < cols and
grid [neighbor [0]] [neighbor [1]] == 0):
```

new_cost = cost_so_far [current] + 1

if neighbor not in cost_so_far or new_cost < cost_so_far [neighbor]:

cost_so_far [neighbor] = new_cost

priority = new_cost + heuristic (goal, neighbor)

heappush (open_list, (priority, new_cost, neighbor))

came_from [neighbor] = current.

Result:

Thus the Program was executed successfully & the o/p is verified.