**Name: Y.BHARGAV**

**Reg. No.: 2116231701063**

**Aim:**

Python program to implement A\* Search.

**Program:**

import heapq

def astar(grid, start, goal):

open\_list = []

heapq.heappush(open\_list, (0, start))

g\_cost = {start: 0}

came\_from = {start: None}

while open\_list:

current = heapq.heappop(open\_list)

if current == goal:

path = []

while current:

path.append(current)

current = came\_from[current]

return path[::-1]

for neighbor in get\_neighbors(grid, current):

tentative\_g\_cost = g\_cost[current] + 1

if neighbor not in g\_cost or tentative\_g\_cost < g\_cost[neighbor]:

g\_cost[neighbor] = tentative\_g\_cost

f\_cost = tentative\_g\_cost + heuristic(neighbor, goal)

heapq.heappush(open\_list, (f\_cost, neighbor))

came\_from[neighbor] = current

return []

def heuristic(node, goal):

x1, y1 = node

x2, y2 = goal

return abs(x1 - x2) + abs(y1 - y2)

def get\_neighbors(grid, node):

x, y = node

neighbors = []

directions = [(0, 1), (1, 0), (0, -1), (-1, 0)]

**Output:**

grid = [

[0, 1, 0, 0, 0],

[0, 1, 0, 1, 0],

[0, 0, 0, 1, 0],

[1, 1, 0, 1, 0],

[0, 0, 0, 0, 0]

]

start = (0, 0)

goal = (4, 4)

path = astar(grid, start, goal)

print(path)

**Result:**

Successfully implemented A\* Search algorithm and output verified.