Name: Mahesh Deshpande **Roll no** – 11

Practical – 2

A* Algorithm – 8 Puzzle game

Code:

```
import heapq
goal_state = [[1, 2, 3],
              [4, 5, 6],
              [7, 8, 0]]

def h(state):
    return sum(abs(state[i][j]//3 - i) + abs(state[i][j]%3 - j) for i in range(3) for j in
range(3) if state[i][j])

def a_star(start_state):
    heap = [(h(start_state), start_state, 0)]
    visited = set()
    while heap:
        (cost, state, g) = heapq.heappop(heap)
        if state == goal_state:
            return g
        if str(state) in visited:
            continue
        visited.add(str(state))
        for (i, j) in [(0, 1), (1, 0), (0, -1), (-1, 0)]:
            new_state = [row[:] for row in state]
            row, col = find_zero(new_state) # type: ignore
            new_row, new_col = row+i, col+j
            if 0 <= new_row < 3 and 0 <= new_col < 3:
                new_state[row][col], new_state[new_row][new_col] =
new_state[new_row][new_col], new_state[row][col]
                heapq.heappush(heap, (g+h(new_state), new_state, g+1))
    return -1

def find_zero(state):
    for i in range(3):
        for j in range(3):
            if state[i][j] == 0:
                return i, j

def print_state(state):
    for i in range(3):
        for j in range(3):
            print(state[i][j], end=' ')
        print()

start_state = [[1, 0, 3],
```

```
              [4, 2, 6],
              [7, 5, 8]]

print("Start state:")
print_state(start_state)

print("Goal state:")
print_state(goal_state)

cost = a_star(start_state)
print("Minimum number of moves:",cost)
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS D:\My Files\Programmes\AI_Pract> & "C:/Program Files/Python311/python.exe" "d:/My Files/Programmes/AI_Pract/A_Star.py"
Start state:
1 0 3
4 2 6
7 5 8
Goal state:
1 2 3
4 5 6
7 8 0
Minimum number of moves: 3
PS D:\My Files\Programmes\AI_Pract>
```