Department of Computer Science & Engineering

Course Title: Artificial Intelligence and Expert Systems Lab

Course Code: CSE 404

Date of Submission: 06/04/2025

Submitted By:

Sabber Hossen Shuvo

Reg no: 21201087

Sec: B2

Submitted To:

Noor Mairukh Khan Arnob

Lecturer,

Department of CSE, UAP

i) Problem Title:

Implementation of a small Address Map (from my own home to UAP) using A* Search Algorithm.

ii) Problem Description:

The objective of this problem is to determine the optimal path & the optimal path cost from Lutfur Rahnan Ln- Bangshal Road(Home) to UAP(University of Asia Pacific) using the A* search algorithm.

A* search algorithm formula,

$f(n) = g(n) + h(n)$

Where,

$f(n)$ = Estimated cost from path n node to goal node

$g(n)$ = Actual Cost from start node to n-node

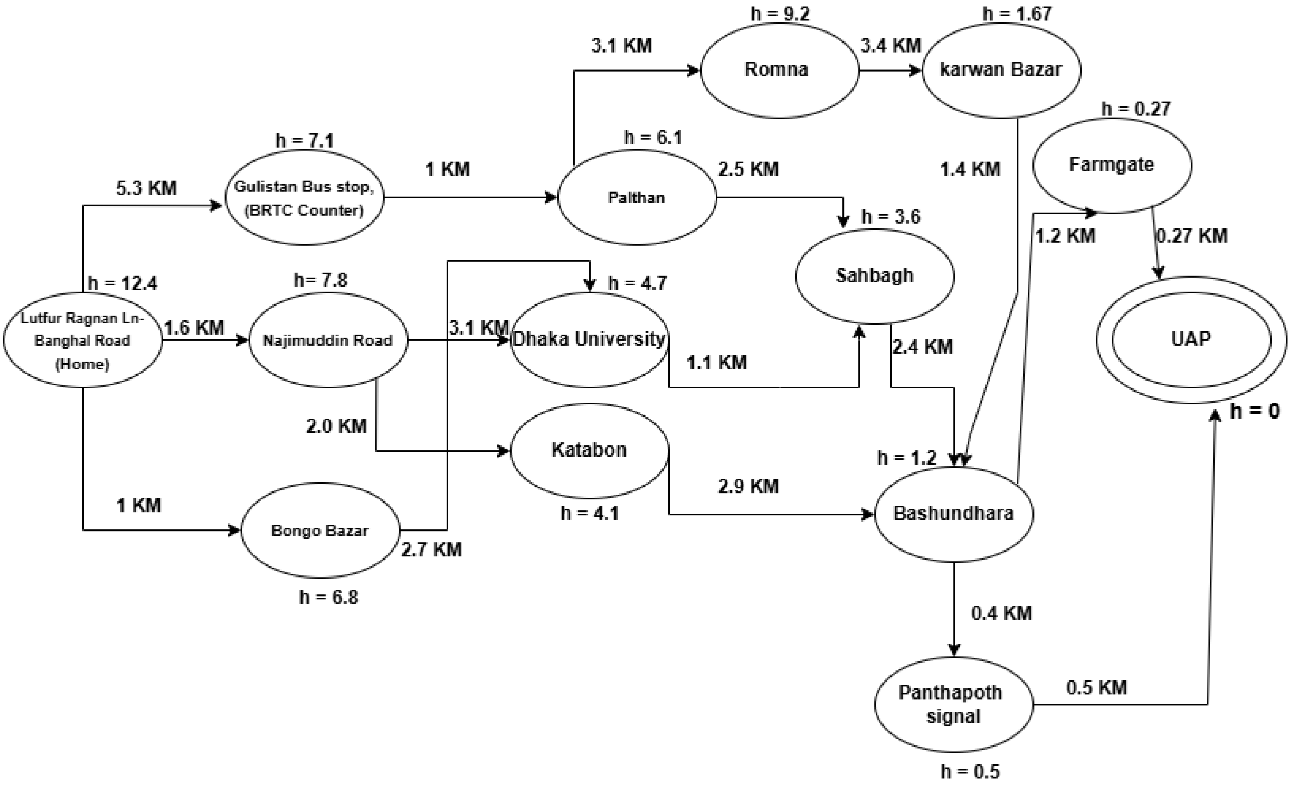
$h(n)$ = Estimated Cost from n-node to goal node

iii) Tools and Languages Used:

- Programming Language: Python
- Tools: PyCharm Professional

iv) Diagram/Figure:

Designed Map:

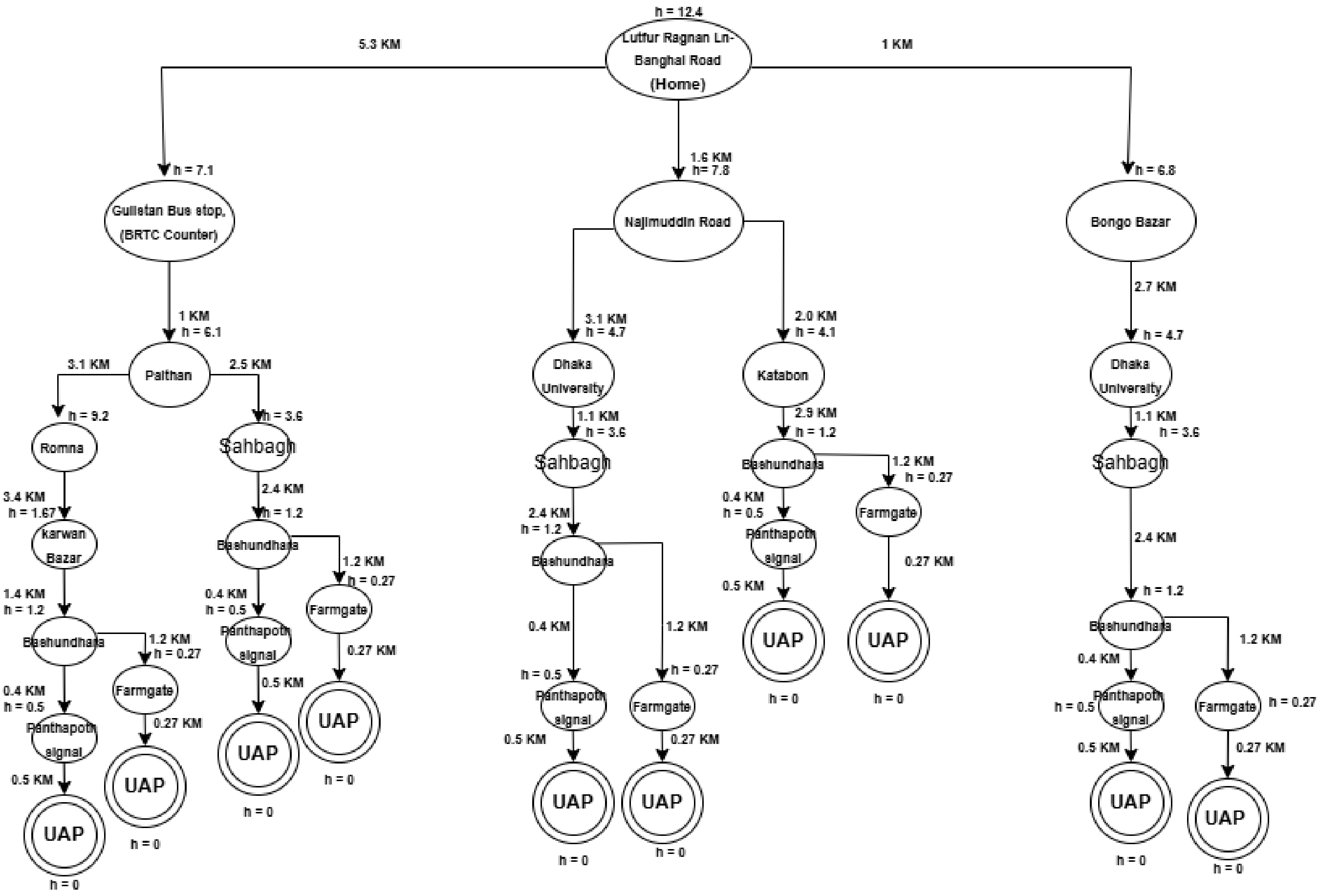


Here, Start Node: Shuvo’s House(Lutfur Rahnan Ln- Bangshal Road)

Goal Node: UAP

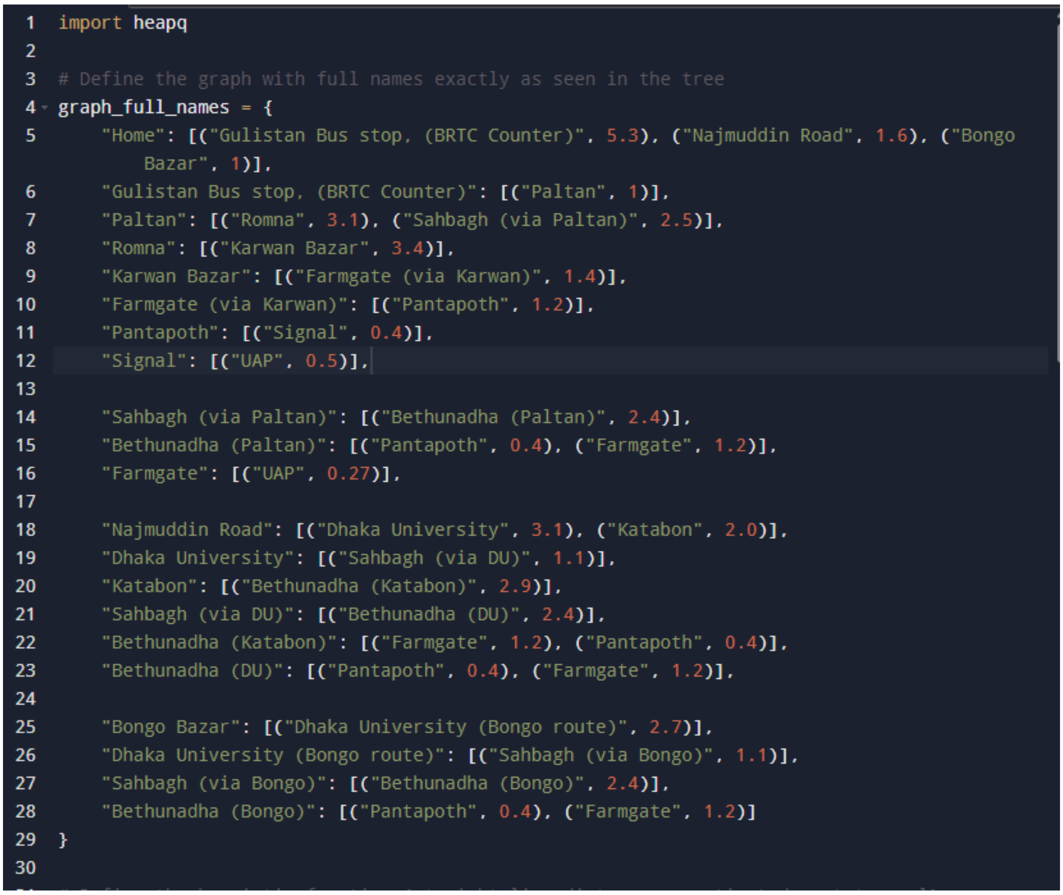
Cost in Distance: Kilometer(km)

Search tree of designed Map:

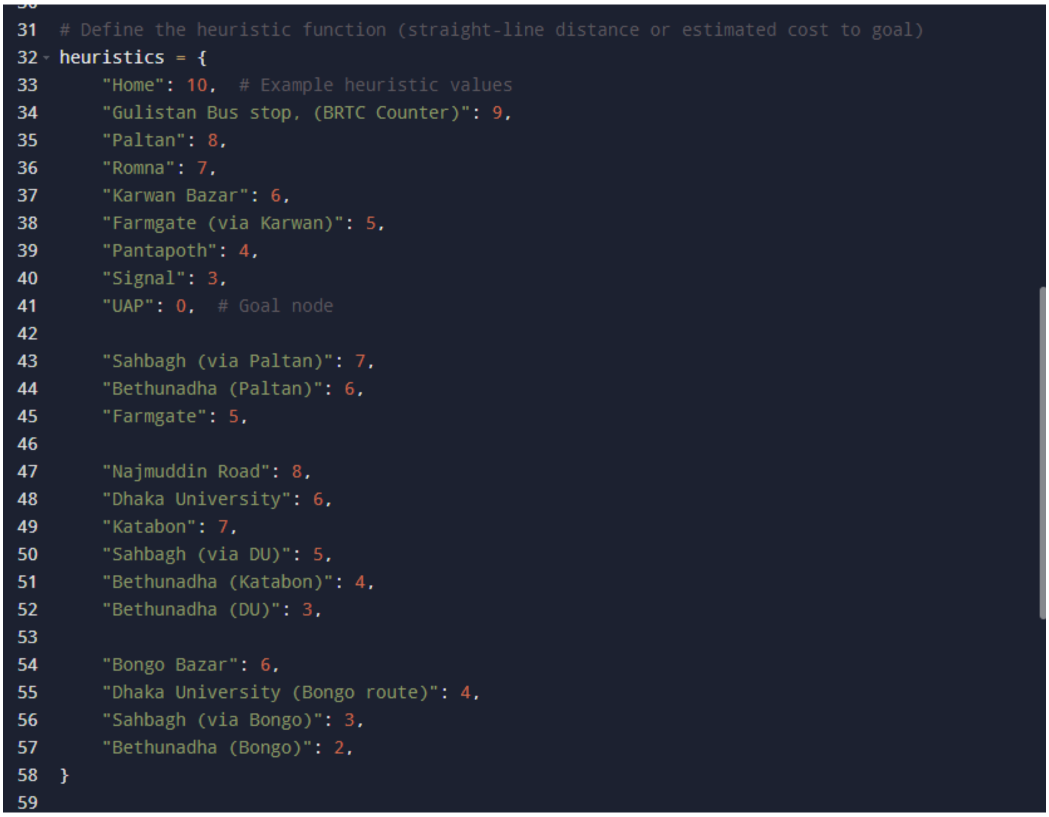


v) Sample Input/Output:

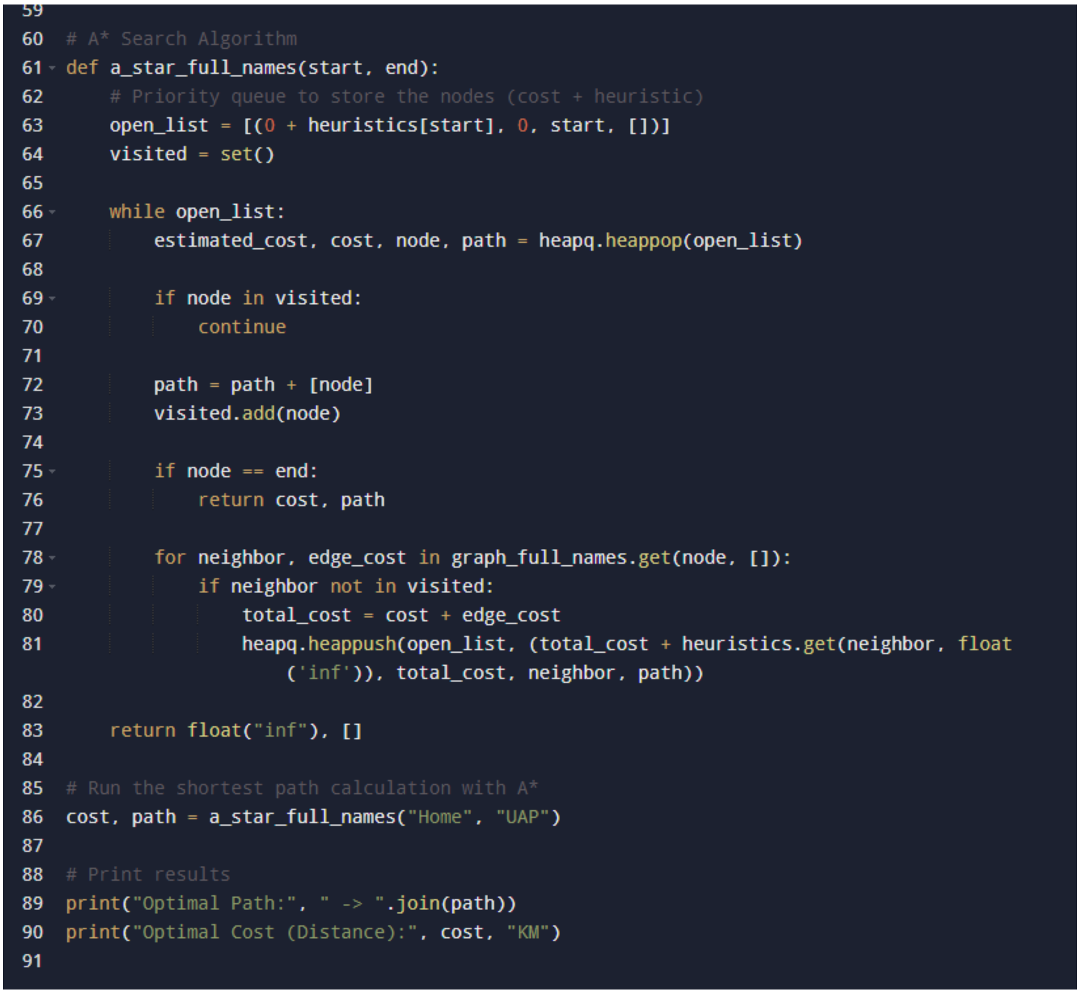
Input:



```python
import heapq

# Define the graph with full names exactly as seen in the tree
graph_full_names = {
    "Home": [("Gulistan Bus stop, (BRTC Counter)", 5.3), ("Najmuddin Road", 1.6), ("Bongo Bazar", 1)],
    "Gulistan Bus stop, (BRTC Counter)": [("Paltan", 1)],
    "Paltan": [("Romna", 3.1), ("Sahbagh (via Paltan)", 2.5)],
    "Romna": [("Karwan Bazar", 3.4)],
    "Karwan Bazar": [("Farmgate (via Karwan)", 1.4)],
    "Farmgate (via Karwan)": [("Pantapoth", 1.2)],
    "Pantapoth": [("Signal", 0.4)],
    "Signal": [("UAP", 0.5)],

    "Sahbagh (via Paltan)": [("Bethunadha (Paltan)", 2.4)],
    "Bethunadha (Paltan)": [("Pantapoth", 0.4), ("Farmgate", 1.2)],
    "Farmgate": [("UAP", 0.27)],

    "Najmuddin Road": [("Dhaka University", 3.1), ("Katabon", 2.0)],
    "Dhaka University": [("Sahbagh (via DU)", 1.1)],
    "Katabon": [("Bethunadha (Katabon)", 2.9)],
    "Sahbagh (via DU)": [("Bethunadha (DU)", 2.4)],
    "Bethunadha (Katabon)": [("Farmgate", 1.2), ("Pantapoth", 0.4)],
    "Bethunadha (DU)": [("Pantapoth", 0.4), ("Farmgate", 1.2)],

    "Bongo Bazar": [("Dhaka University (Bongo route)", 2.7)],
    "Dhaka University (Bongo route)": [("Sahbagh (via Bongo)", 1.1)],
    "Sahbagh (via Bongo)": [("Bethunadha (Bongo)", 2.4)],
    "Bethunadha (Bongo)": [("Pantapoth", 0.4), ("Farmgate", 1.2)]
}

```

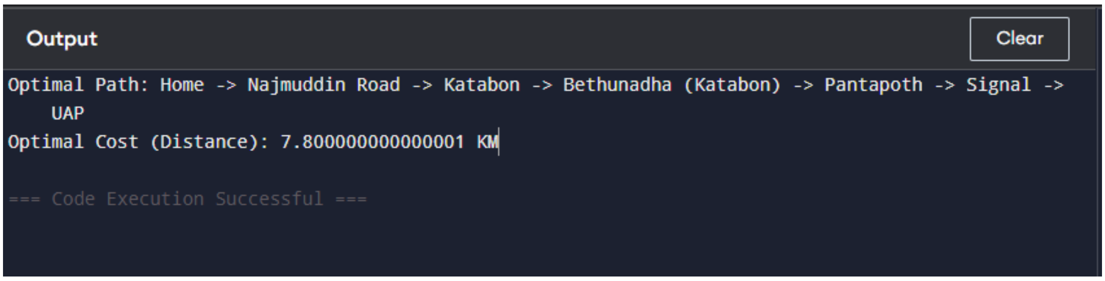


```python
# Define the heuristic function (straight-line distance or estimated cost to goal)
heuristics = {
    "Home": 10,  # Example heuristic values
    "Gulistan Bus stop, (BRTC Counter)": 9,
    "Paltan": 8,
    "Romna": 7,
    "Karwan Bazar": 6,
    "Farmgate (via Karwan)": 5,
    "Pantapoth": 4,
    "Signal": 3,
    "UAP": 0,  # Goal node

    "Sahbagh (via Paltan)": 7,
    "Bethunadha (Paltan)": 6,
    "Farmgate": 5,

    "Najmuddin Road": 8,
    "Dhaka University": 6,
    "Katabon": 7,
    "Sahbagh (via DU)": 5,
    "Bethunadha (Katabon)": 4,
    "Bethunadha (DU)": 3,

    "Bongo Bazar": 6,
    "Dhaka University (Bongo route)": 4,
    "Sahbagh (via Bongo)": 3,
    "Bethunadha (Bongo)": 2,
}

```



```python

# A* Search Algorithm
def a_star_full_names(start, end):
    # Priority queue to store the nodes (cost + heuristic)
    open_list = [(0 + heuristics[start], 0, start, [])]
    visited = set()

    while open_list:
        estimated_cost, cost, node, path = heapq.heappop(open_list)

        if node in visited:
            continue

        path = path + [node]
        visited.add(node)

        if node == end:
            return cost, path

        for neighbor, edge_cost in graph_full_names.get(node, []):
            if neighbor not in visited:
                total_cost = cost + edge_cost
                heapq.heappush(open_list, (total_cost + heuristics.get(neighbor, float('inf')), total_cost, neighbor, path))

    return float("inf"), []

# Run the shortest path calculation with A*
cost, path = a_star_full_names("Home", "UAP")

# Print results
print("Optimal Path:", " -> ".join(path))
print("Optimal Cost (Distance):", cost, "KM")

```

Output:



```
Optimal Path: Home -> Najmuddin Road -> Katabon -> Bethunadha (Katabon) -> Pantapoth -> Signal ->
    UAP
Optimal Cost (Distance): 7.800000000000001 KM

=== Code Execution Successful ===
```

vi) Conclusion:

By implementing the A* search algorithm, we efficiently determined the most optimal path and the optimal path cost from Lutfur Rahnan Ln- Bangshal Road(Home))to UAP, minimizing travel distance. The algorithm effectively balances the actual travel cost (g(n)) with the estimated distance (h(n)), ensuring the shortest possible route while maintaining high computational efficiency.