Department of Computer Science & Engineering

Course Title: Artificial Intelligence and Expert Systems Lab

Course Code: CSE 404

Date of Submission: 04/04/2025

Submitted To:

Noor Mairukh Khan Arnob

Lecturer,

Department of CSE, UAP

Submitted By:

Name: Swapna Khanam

Reg no: 21201082

Sec: B2

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 Pulpar mosjid(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

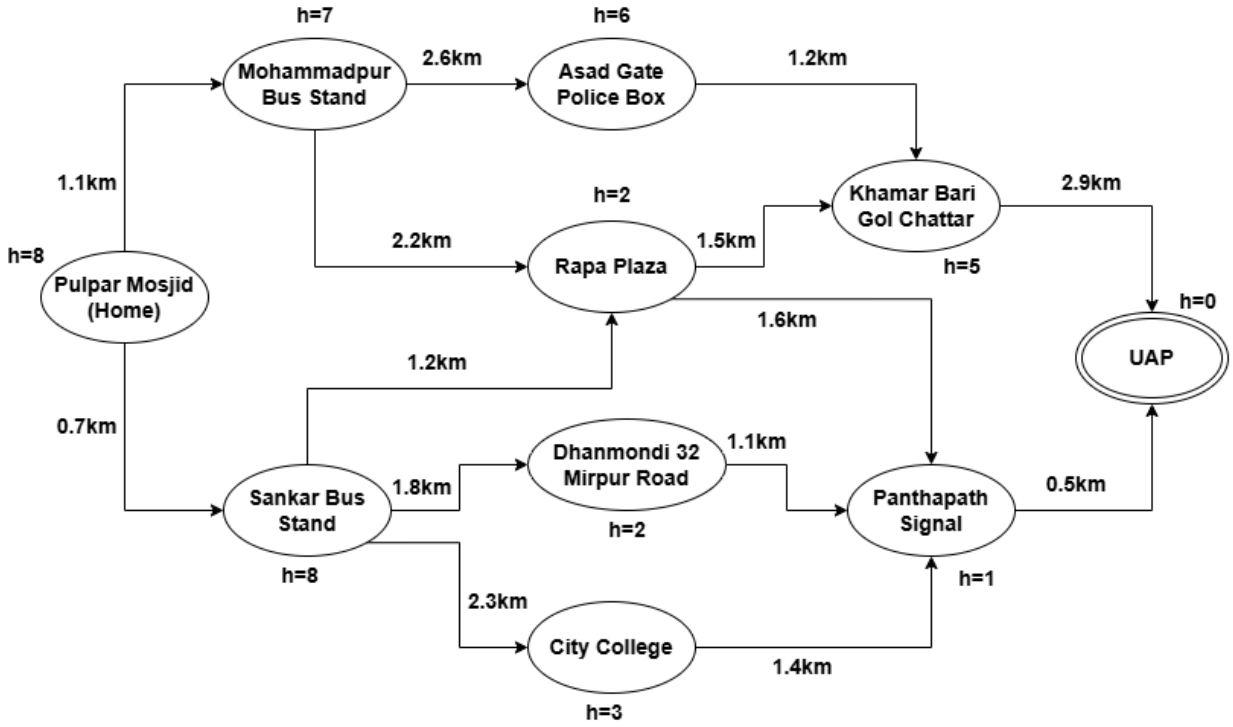
h(n) = Estimated Cost from n-node to goal node

iii) Tools and Languages Used:

- Programming Language: Python
- Tools: PyCharm Professional

iv) Diagram/Figure:

Designed Map:

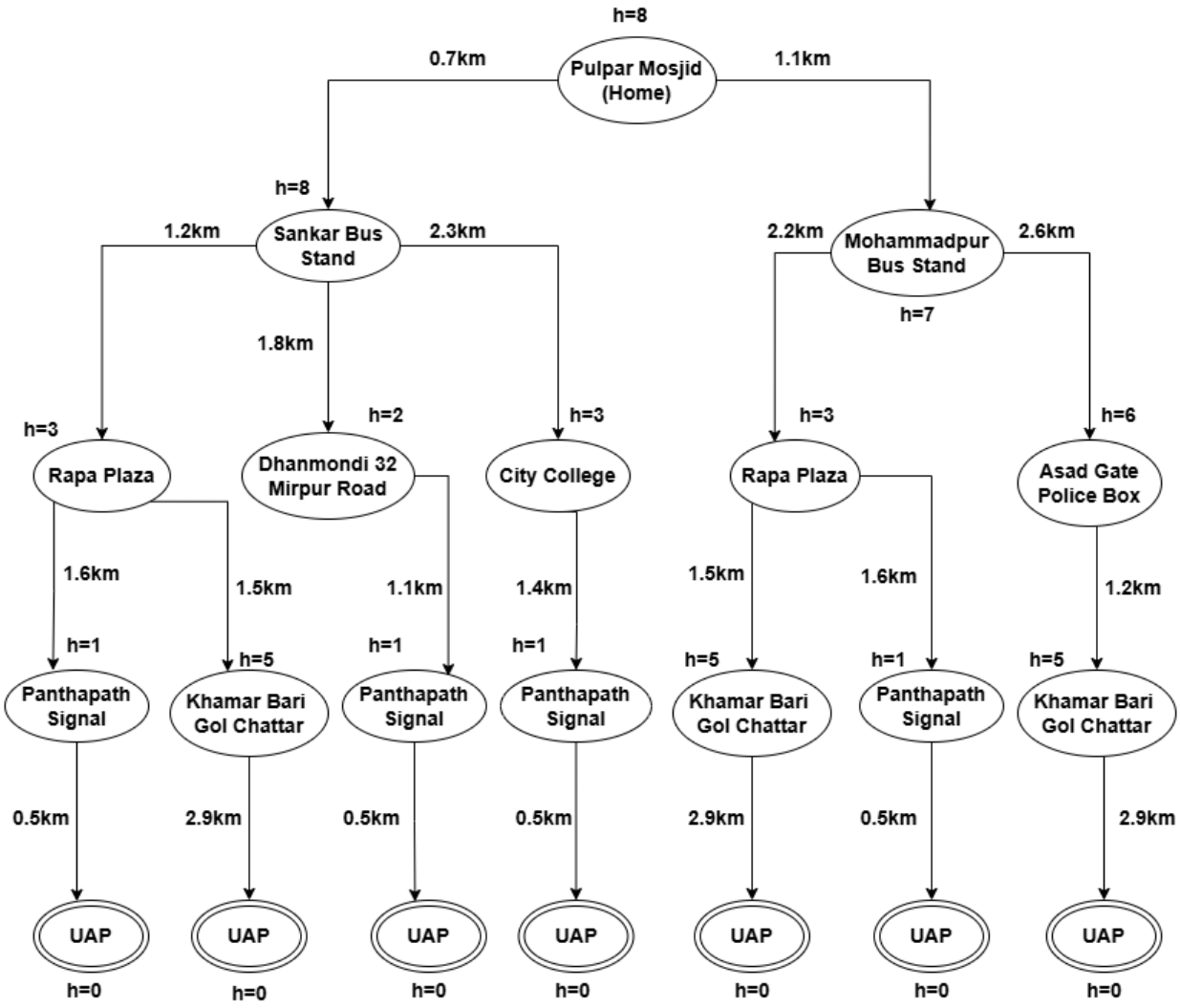


Here, Start Node: Swapna's House(Pulpar Mosjid)

Goal Node: UAP

Cost in Distance: Kilometer(km)

Search tree of designed Map:



v) Sample Input/Output:

Input:

A_Star_Search.py

```python
import heapq

graph = {
    "Pulpar Mosjid (Home)": [("Sankar Bus Stand", 0.7), ("Mohammadpur Bus Stand", 1.1)],
    "Sankar Bus Stand": [("Rapa Plaza", 1.2), ("Dhanmondi 32 Mirpur Road", 1.8), ("City College", 2.3)],
    "Mohammadpur Bus Stand": [("Rapa Plaza", 2.2), ("Asad Gate Police Box", 2.6)],
    "Rapa Plaza": [("Panthapath Signal", 1.6), ("Khamar Bari Gol Chattar", 1.5)],
    "Dhanmondi 32 Mirpur Road": [("Panthapath Signal", 1.1)],
    "City College": [("Panthapath Signal", 1.4)],
    "Asad Gate Police Box": [("Khamar Bari Gol Chattar", 1.2)],
    "Panthapath Signal": [("UAP", 0.5)],
    "Khamar Bari Gol Chattar": [("UAP", 2.9)],
    "UAP": []
}

# Updated admissible and consistent heuristics
heuristics = {
    "Pulpar Mosjid (Home)": 4,
    "Sankar Bus Stand": 3.3,
    "Mohammadpur Bus Stand": 5,
    "Rapa Plaza": 2.1,
    "Dhanmondi 32 Mirpur Road": 2.5,
    "City College": 2.5,
    "Asad Gate Police Box": 4,
    "Panthapath Signal": 0.5,
    "Khamar Bari Gol Chattar": 3,
    "UAP": 0
}

def a_star(start, goal):
    pq = []
    heapq.heappush(pq, __item: (heuristics[start], 0, start, []))  # (f = g + h, g, node, path)
    visited = {}

    while pq:
        f, g, node, path = heapq.heappop(pq)

        if node == goal:
            return path + [node], g

        if node in visited and visited[node] <= g:
            continue

        visited[node] = g

        for neighbor, cost in graph[node]:
            g_new = g + cost
            f_new = g_new + heuristics[neighbor]
            heapq.heappush(pq, __item: (f_new, g_new, neighbor, path + [node]))

    return None, float('inf')

# Run the algorithm
path, cost = a_star( start: "Pulpar Mosjid (Home)", goal: "UAP")
print("Optimal Path:", " -> ".join(path))
print("Optimal Path Cost:", cost)
```

Output:

```
Optimal Path: Pulpar Mosjid (Home) -> Sankar Bus Stand -> Rapa Plaza -> Panthapath Signal -> UAP
Optimal Path Cost: 4.0
```

vi) Conclusion:

By implementing the A* search algorithm, we efficiently determined the most optimal path and the optimal path cost from Pulpar Mosjid (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.