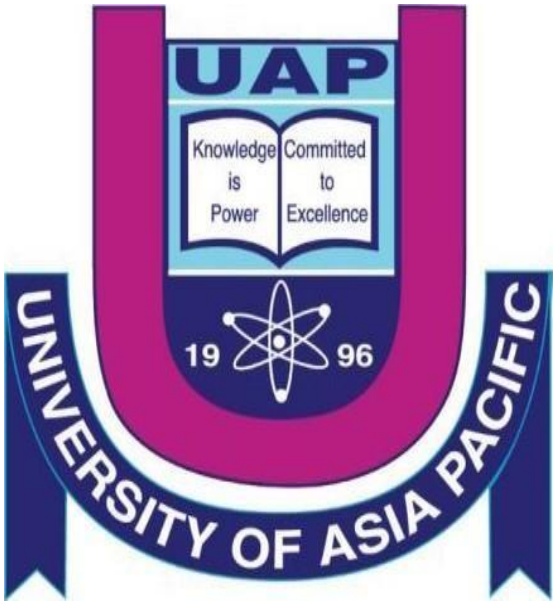


Department of Computer Science & Engineering

Course Title: Artificial Intelligence & Expert System Lab

Course Code: CSE 404

Assignment No: 02

Assignment On, Implementation of a small Address Map (from your home to UAP) using A* Search Algorithm

Submitted To:
Noor Mairukh Khan Arnob
Lecturer,
Department of CSE, UAP

Submitted By:
Name: Amirul Islam Papon
Reg No: 21201076
Sec: B

Problem Title: Optimal Path Finding from Home to UAP Using A* Search Algorithm

Problem Description: The objective of this assignment was to implement the A* search algorithm in Python to find the optimal path from a starting location ("Home") to a destination ("UAP") based on a provided graph of locations and their distances.

A* search algorithm formula,

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

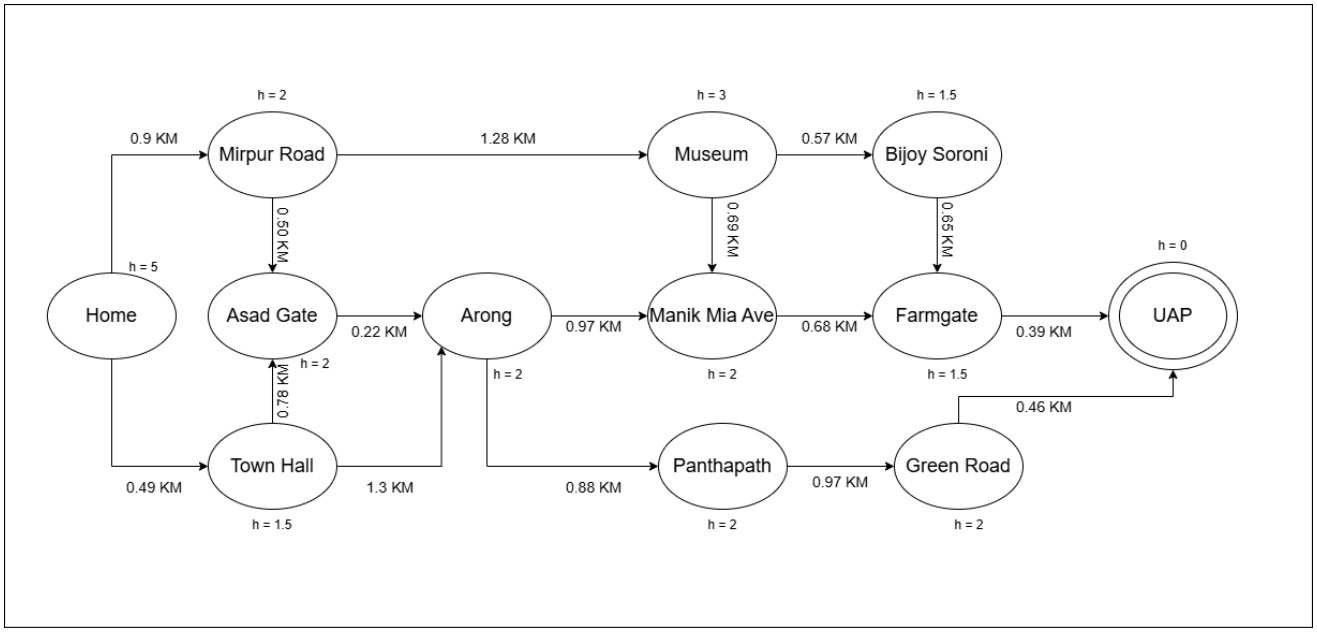
Where,

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

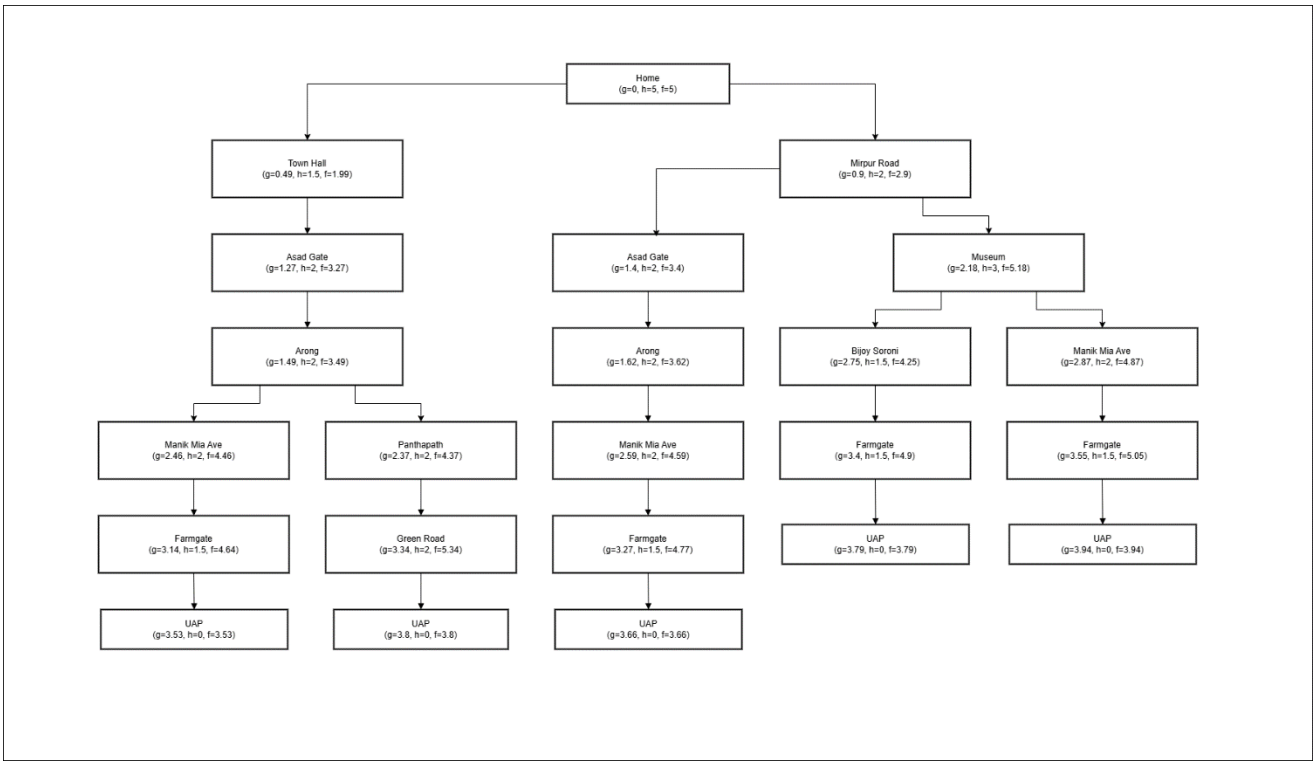
g(n) = Actual Cost from start node to n-node

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

Designed Map:



Search Tree:



Source Code:

A_star_search.py > a_star_search

```python
import heapq
from itertools import count

def a_star_search(graph, heuristics, start, goal):
    counter = count()

    open_set = [(heuristics[start], next(counter), start)]

    came_from = {}

    g_scores = {node: float('inf') for node in graph}
    g_scores[start] = 0

    while open_set:
        _, _, current = heapq.heappop(open_set)

        if current == goal:
            path = []
            while current in came_from:
                path.append(current)
                current = came_from[current]
            path.append(start)
            return path[::-1], g_scores[goal]

        for neighbor, distance in graph[current].items():
            tentative_g = g_scores[current] + distance

            if tentative_g < g_scores[neighbor]:
                came_from[neighbor] = current
                g_scores[neighbor] = tentative_g

                f_score = tentative_g + heuristics[neighbor]

                heapq.heappush(open_set, (f_score, next(counter), neighbor))

    return None, None

```

A_star_search.py > a_star_search

```python
graph = {
    'Home': {'Mirpur Road': 0.9, 'Town Hall': 0.49},
    'Mirpur Road': {'Home': 0.9, 'Museum': 1.28, 'Asad Gate': 0.5},
    'Asad Gate': {'Mirpur Road': 0.5, 'Arong': 0.22, 'Town Hall': 0.78},
    'Town Hall': {'Home': 0.49, 'Arong': 1.3, 'Asad Gate': 0.78},
    'Arong': {'Asad Gate': 0.22, 'Manik Mia Ave': 0.97, 'Town Hall': 1.3, 'Panthapath': 0.88},
    'Museum': {'Mirpur Road': 1.28, 'Manik Mia Ave': 0.69, 'Bijoy Soroni': 0.57},
    'Manik Mia Ave': {'Arong': 0.97, 'Museum': 0.69, 'Farmgate': 0.68},
    'Panthapath': {'Arong': 0.88, 'Green Road': 0.97},
    'Green Road': {'Panthapath': 0.97, 'UAP': 0.46},
    'Bijoy Soroni': {'Museum': 0.57, 'Farmgate': 0.65},
    'Farmgate': {'Bijoy Soroni': 0.65, 'Manik Mia Ave': 0.68, 'UAP': 0.39},
    'UAP': {'Farmgate': 0.39, 'Green Road': 0.46}
}

heuristics = {
    'Home': 5,
    'Mirpur Road': 2,
    'Asad Gate': 2,
    'Town Hall': 1.5,
    'Arong': 2,
    'Museum': 3,
    'Manik Mia Ave': 2,
    'Panthapath': 2,
    'Green Road': 2,
    'Bijoy Soroni': 1.5,
    'Farmgate': 1.5,
    'UAP': 0
}

start_node = 'Home'
goal_node = 'UAP'

path, total_cost = a_star_search(graph, heuristics, start_node, goal_node)

if path:
    print("Optimal Path:", " -> ".join(path))
    print(f"Optimal Cost: {total_cost:.2f} km")
else:
    print("No path found from", start_node, "to", goal_node)

```

Output:

```
[Running] python -u "d:\-Artificial-Intelligence-and-Expert-System-\A_star_search.py"
Optimal Path: Home -> Town Hall -> Asad Gate -> Arong -> Manik Mia Ave -> Farmgate -> UAP
Optimal Cost: 3.53 km

[Done] exited with code=0 in 0.052 seconds
```

Results:

The A* search algorithm successfully found the optimal path from "Home" to "UAP," with the following results:

- **Optimal Path**: "Home" → "Town Hall" → "Asad Gate" → "Arong" → "Manik Mia Ave" → "Farmgate" → "UAP"
- **Total Cost**: 3.53 km

This path and cost were derived by summing the individual edge distances along the route:

- Home to Town Hall: 0.49 km
- Town Hall to Asad Gate: 0.78 km
- Asad Gate to Arong: 0.22 km
- Arong to Manik Mia Ave: 0.97 km
- Manik Mia Ave to Farmgate: 0.68 km
- Farmgate to UAP: 0.39 km
- Total: 0.49 + 0.78 + 0.22 + 0.97 + 0.68 + 0.39 = 3.53 km

Conclusion:

The assignment successfully demonstrated the application of the A* search algorithm to solve a pathfinding problem in a road network. The implementation in Python effectively found the optimal path from "Home" to "UAP" with a total cost of 3.53 km, which matched the expected distance based on the provided graph data. The use of heuristics guided the search efficiently