CSE 404

Artificial Intelligence and Expert Systems Lab

Project Report - 2: Optimal Pathfinding using A* Search Algorithm.

Submission Date: 23rd August, 2025

Submitted by

Name: Noor Mohammed Priom

Registration ID: 22101134

Section: C-2

Semester: 4-1

Session: Spring 2025

Submitted to

Bidita Sarkar Diba

Lecturer

Department of Computer Science & Engineering

University of Asia Pacific

Contents

I. Problem Statement

I have to make a graph which represents the paths from my home to my University. Then the graph must be solved by finding the optimal path using A* Search Algorithm. Many times we face such instances where there exist several routes from a source to a destination. Choosing the shortest path is vital in saving time, money, or distance.

- This problem I can model as a graph, in which:
- Nodes are the places (locations, intersections, landmarks)
- Edges are links between nodes (roads, paths) which will hold weights or path cost.

Here, I need to use the A* (A-Star) search algorithm to find the lowest cost trip from Home to UAP.

II. Purpose & Goal

- To figure out all the paths between my home & University.
- To model the road network from Home to UAP as a weighted directed graph.
- Implement the A* search algorithm in Python.
- To employ heuristics so as to direct the search to the goal efficiently.
- To find the optimal route and overall cost of traveling from my home to my University.

III. Graph Representation

There are many paths from my home to my University. Such as-

1. Home ➝ Mohammed_Pur ➝ Khamar_Bari ➝ Farm_Gate ➝ UAP.
2. Home ➝ Shre-e_bangla_road ➝ Modhu_Bazar ➝ Abahoni_Math ➝ 8_No_Bridge ➝ Kola_Baghan ➝ Panthopoth ➝ UAP
3. Home ➝ Tenary_Mor ➝ Jhigatola ➝ Science_Lab ➝ Panthopoth ➝ UAP.

How I've represented the graph-

- Each Node of the graph will represent a location or place on the way to my University.
- Each edge will represent the road or path from source node to destination node. The edge will direct from the source node to the destination node. That means edges of the graph will interlink all the nodes of the graph.
- Each node will hold a weight, which represents the distance between source node to destination node.
- Therefore the Graph will be a directed weighted graph.

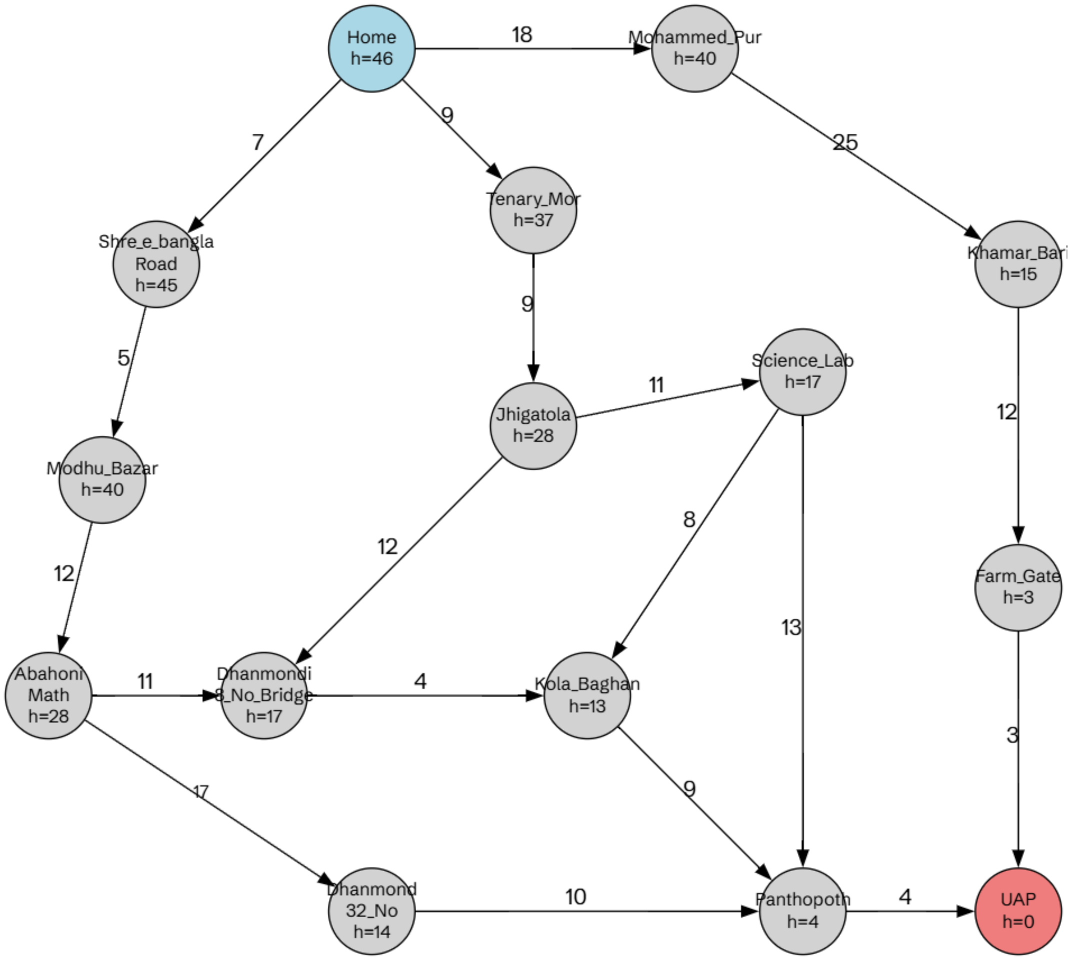

Figure 1: Designed Graph

IV. Heuristic Estimation Calculation

There are two ways to calculate the heuristic value -

1. Straight Line Cost.
2. Minimum Cost.

In both ways the graph heuristic won't be overestimated and will be admissible & consistent. Because **Actual cost** $\geq$ **Straight Line Cost** & **Actual cost** $\geq$ **Minimum Cost.** I've calculated both-

Minimum Cost for all the nodes

UAP: h(UAP) = 0 (goal node)

Panthopoth: Panthopoth → UAP: h(Panthopoth) = 4

32_No: 32_No → Panthopoth → UAP: h(32_No)=10 + 4 = 14

Abahoni_Math:
Abahoni_Math → 32_No → Panthopoth → UAP: h(Abahoni_Math)=17 + 10 + 4 = 31
Abahoni_Math → 8_No_Bridge → Kola_Baghan → Panthopoth → UAP: h(Abahoni_Math)=11 + 4 + 9 + 4 = 28
Minimum is 28.

Modhu_Bazar
Modhu_Bazar → Abahoni_Math → 32_No → Panthopoth → UAP: h(Modhu_Bazar)=12 + 17 + 10 + 4 = 43
Modhu_Bazar → Abahoni_Math → 8_No_Bridge → Kola_Baghan → Panthopoth → UAP (4): 12 + 11 + 4 + 9 + 4 = 40
Minimum is 40.

Shre-e_bangla_road
Shre-e_bangla_road → Modhu_Bazar → Abahoni_Math → 32_No → Panthopoth → UAP: h(Shre-e_bangla_road)=5 + 12 + 17 + 10 + 4 = 48
Shre-e_bangla_road → Modhu_Bazar → Abahoni_Math → 8_No_Bridge → Kola_Baghan → Panthopoth → UAP: h(Shre-e_bangla_road)= 5 + 12 + 11 + 4 + 9 + 4 = 45
Minimum is 45.

Tenary_Mor
Tenary_Mor → Jhigatola → Science_Lab → Panthopoth (13) → UAP: h(Tenary_Mor)=9 + 11 + 13 + 4 = 37
Tenary_Mor → Jhigatola → Science_Lab → Kola_Baghan → Panthopoth → UAP: h(Tenary_Mor)= 9 + 11 + 8 + 9 + 4 = 41
Tenary_Mor → Jhigatola → 8_No_Bridge → Kola_Baghan → Panthopoth → UAP: h(Tenary_Mor)=9 + 12 + 4 + 9 + 4 = 38
Minimum is 37.

Jhigatola
Jhigatola → Science_Lab → Panthopoth → UAP: h(Jhigatola)=11 + 13 + 4 = 28
Jhigatola → Science_Lab → Kola_Baghan → Panthopoth → UAP: h(Jhigatola)=11 + 8 + 9 + 4 = 32
Jhigatola → 8_No_Bridge → Kola_Baghan → Panthopoth → UAP : h(Jhigatola)=12 + 4 + 9 + 4 = 29
Minimum is 28.

This how I’ve Calculated All the Minimum Cost & Assign them to each node’s heuristic value-

Node(n)	Heuristic Estimation Value h(n)
UAP	0
Panthopoth	4
32_No	14
Abahoni_Math	28
Modhu_Bazar	40
Shre-e_bangla_road	45
Home	46
Mohammed_Pur	40
Khamar_Bari	15
Farm_Gate	3
Tenary_Mor	37
Jhigatola	28
Science_Lab	17
Kola_Baghan	13
8_No_Bridge	17

Straight Line Cost for all the nodes

I’ve used google map’s distance measuring tool for finding out the straight line cost & Assign them to each node’s heuristic value-

Node(n)	Heuristic Estimation Value h(n)
UAP	0
Panthopoth	4
32_No	13
Abahoni_Math	20
Modhu_Bazar	26
Shre-e_bangla_road	28
Home	33

Mohammed_Pur	29
Khamar_Bari	5
Farm_Gate	3
Tenary_Mor	19
Jhigatola	28
Science_Lab	17
Kola_Baghan	11
8_No_Bridge	15

V. Admissibility and Consistency

Because **Actual cost ≥ Straight Line Cost** & **Actual cost ≥ Minimum Cost.** So the graph will be always-

1. Admissible
2. Consistant

❖ **Admissibility Test for Minimum Cost-**

Panthopoth

Path: Panthopoth → UAP = 4

True cost = 4, h(Panthopoth) = 4 → OK

32_No

Path: 32_No → Panthopoth (10) → UAP (4) = 14

True cost = 14, h(32_No) = 14 → OK

Abahoni_Math

Route 1: Abahoni → 32_No (17) → Panthopoth (10) → UAP (4) = 31

Route 2 (better): Abahoni → 8_No_Bridge (11) → Kola_Baghan (4) → Panthopoth (9) → UAP (4) = 11 + 4 + 9 + 4 = 28

True cost = 28, h(Abahoni_Math) = 28 → OK

Science_Lab

Route 1 (better): Science_Lab → Panthopoth (13) → UAP (4) = 17

Route 2: Science_Lab → Kola_Baghan (8) → Panthopoth (9) → UAP (4) = 21

True cost = 17, h(Science_Lab) = 17 → OK

Kola_Baghan
Kola_Baghan → Panthopoth (9) → UAP (4) = 13
True cost = 13, h(Kola_Baghan) = 13 → OK

Jhigatola
Route 1 (better): Jhigatola → Science_Lab (11) → Panthopoth (13) → UAP (4)
= 11 + 13 + 4 = 28
Route 2: Jhigatola → 8_No_Bridge (12) → Kola_Baghan (4) → Panthopoth (9) → UAP (4) = 29
True cost = 28, h(Jhigatola) = 28 → OK

❖ **Admissibility Test for Straightline Cost-**
Panthopoth
Panthopoth → UAP = 4
True cost = 4; h(Panthopoth)=4 → OK

32_No
32_No → Panthopoth (10) → UAP (4) = 14
True cost = 14; h(32_No)=13 $\leq$ 14 → OK

Abahoni_Math
Abahoni_Math → 8_No_Bridge (11) → Kola_Baghan (4) → Panthopoth (9) → UAP (4)
= 11+4+9+4 = 28
True cost = 28; h(Abahoni_Math)=20 $\leq$ 28 → OK

Modhu_Bazar
Modhu_Bazar → Abahoni (12) → ... → UAP (28 from Abahoni) = 12+28 = 40
True cost = 40; h(Modhu_Bazar)=26 $\leq$ 40 → OK

Shre-e_bangla_road
Shre-e_bangla_road → Modhu_Bazar (5) → ... → UAP (40 from Modhu) = 45
True cost = 45; h(Shre-e_bangla_road)=28 $\leq$ 45 → OK

Mohammed_Pur
Mohammed_Pur → Khamar_Bari (25) → Farm_Gate (12) → UAP (3) = 40
True cost = 40; h(Mohammed_Pur)=29 $\leq$ 40 → OK

Khamar_Bari

Khamar_Bari → Farm_Gate (12) → UAP (3) = 15

True cost = 15; $h(\text{Khamar\_Bari})=5 \leq 15$ → OK

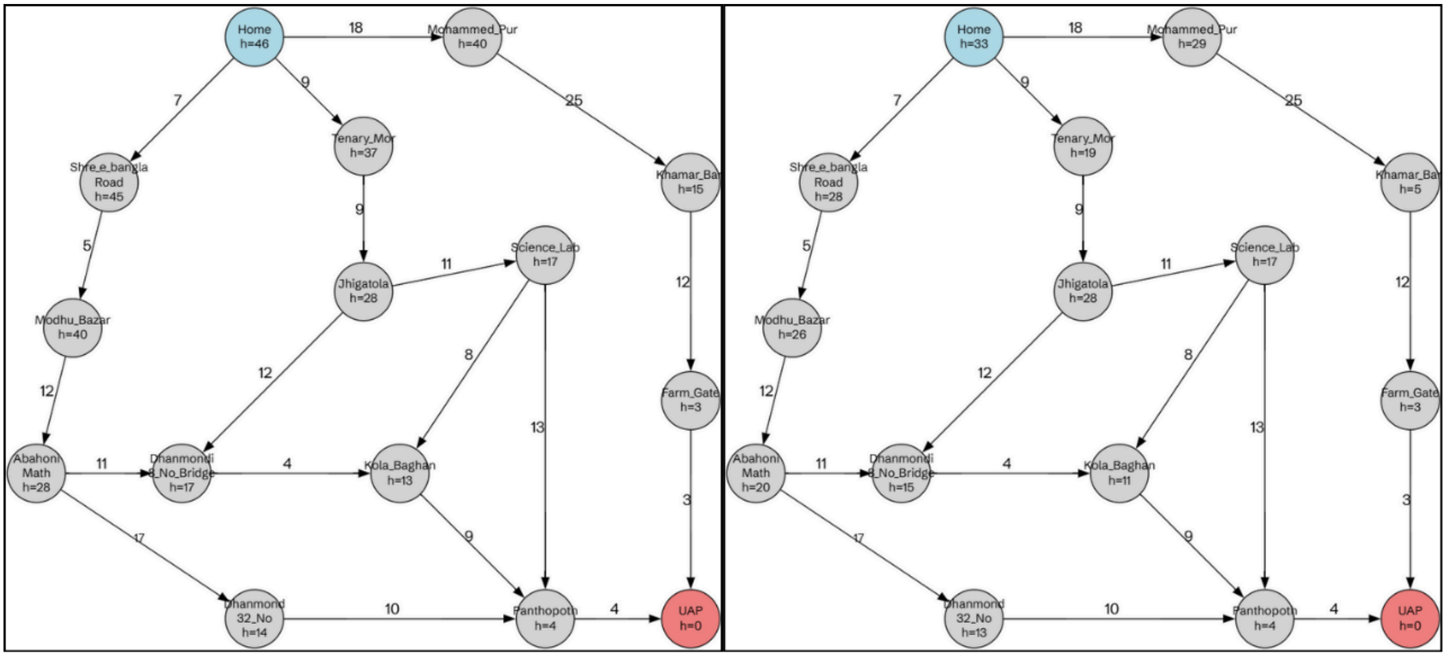

Figure 2: Graph with both heuristic estimation value

❖ **Consistency Test for Minimum Cost-**

Home → Tenary_Mor:

$h(\text{Home})=46 \leq 9+h(\text{Tenary\_Mor})=9+37=46h(\text{Home})=46$

Home → Shre-e_bangla_road:

$46 \leq 7 + 45=52\ 46$

Home → Mohammed_Pur:

$46 \leq 18 + 40=58$

Tenary_Mor → Jhigatola:

$37 \leq 9 + 28 =37$

Jhigatola → Science_Lab:
$28 \leq 11 + 17$=28

Jhigatola → 8_No_Bridge:
$28 \leq 12 + 17 = 29$

Science_Lab → Panthopoth:
$17 \leq 13 + 4$=17

Science_Lab → Kola_Baghan:
$17 \leq 8 + 13 = 21$

Kola_Baghan → Panthopoth:
$13 \leq 9 + 4$=13

Abahoni_Math → 32_No:
$28 \leq 17 + 14$ =31

❖ **Consistency Test for Straightline Cost-**
Home → Shre-e_bangla_road
$33 \leq 7+28=35$

Shre-e_bangla_road → Modhu_Bazar
$28 \leq 5+26=31$

Modhu_Bazar → Abahoni_Math
$26 \leq 12+20=32$

Abahoni_Math → 32_No
$20 \leq 17+13=30$

32_No → Panthopoth
$13 \leq 10+4=14$

Panthopoth → UAP
$4 \leq 4 + 0$=4

Home → Mohammed_Pur
$33 \leq 18+29=47$

Mohammed_Pur → Khamar_Bari
$29 \leq 25+5=30$

Khamar_Bari → Farm_Gate
$5 \leq 12+3=15$

Farm_Gate → UAP
$3 \leq 3+0=3$

VI. Optimal Path Finding

Using heuristic value as minimum cost-

Node	h(n)	g(n)	f(n)	Open List	Close List
Home	46	0	46	{Tenary_Mor(46), Shre-e_bangla_road(52), Mohammed_Pur(58)}	{Home}
Tenary_Mor	37	9	46	{Jhigatola(46), Shre-e_bangla_road(52), Mohammed_Pur(58)}	{Home, Tenary_Mor}
Jhigatola	28	18	46	{Science_Lab(46), Shre-e_bangla_road(52), Mohammed_Pur(58), 8_No_Bridge(47)}	{Home, Tenary_Mor, Jhigatola}
Science_Lab	17	29	46	{Panthopoth(46), Shre-e_bangla_road(52), Mohammed_Pur(58), 8_No_Bridge(47), Kola_Baghan(50)}	{Home, Tenary_Mor, Jhigatola, Science_Lab}
Panthopoth	4	42	46	{UAP(46), Shre-e_bangla_road(52), Mohammed_Pur(58), 8_No_Bridge(47), Kola_Baghan(50)}	{Home, Tenary_Mor, Jhigatola, Science_Lab, Panthopoth}
UAP	0	46	46	{Shre-e_bangla_road(52), Mohammed_Pur(58), 8_No_Bridge(47), Kola_Baghan(50)}	{Home, Tenary_Mor, Jhigatola, Science_Lab, Panthopoth, UAP}

Table 1: A* Search (Using minimum Cost)

We've reached the goal node. Therefore the A* Optimal path from Home to UAP: Home → Tenary_Mor → Jhigatola → Science_Lab → Panthopoth → UAP. Total Cost: 46

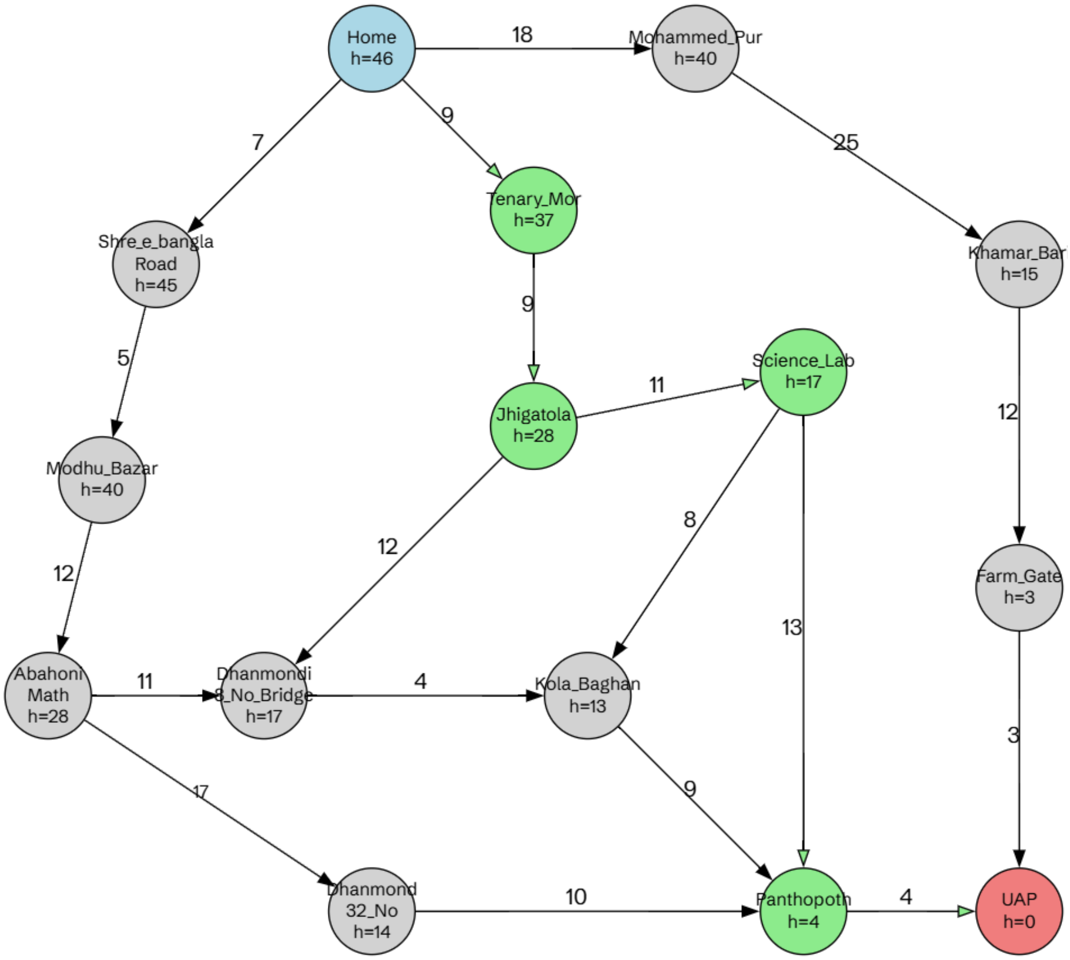

Figure 3: Graph Highlighted Optimal Path

Using heuristic value as straight line cost-

Node	h(n)	g(n)	f(n)	Open List	Close List
Home	33	0	33	{Tenary_Mor(28), Shre-e_bangla_road(35), Mohammed_Pur(47)}	{Home}
Tenary_Mor	19	9	28	{Jhigatola(37), Shre-e_bangla_road(35), Mohammed_Pur(47)}	{Home, Tenary_Mor}
Shre-e_bangla _road	28	7	35	{Jhigatola(37), Modhu_Bazar(33), Mohammed_Pur(47)}	{Home, Tenary_Mor, Shre-e_bangla_road}
Modhu_Bazar	26	12	38	{Jhigatola(37), Abahoni_Math(32), Mohammed_Pur(47)}	{Home, Tenary_Mor, Shre-e_bangla_road, Modhu_Bazar}

Abahoni_Math	20	24	44	{Jhigatola(37), 32_No(37), 8_No_Bridge(39), Mohammed_Pur(47)}	{Home, Tenary_Mor, Shre-e_bangla_road, Modhu_Bazar, Abahoni_Math}
32_No	13	41	54	{Jhigatola(37), 8_No_Bridge(39), Panthopoth(51), Mohammed_Pur(47)}	{Home, Tenary_Mor, Shre-e_bangla_road, Modhu_Bazar, Abahoni_Math, 32_No}
Jhigatola	28	18	46	{Science_Lab(29), 8_No_Bridge(30), Panthopoth(51), Mohammed_Pur(47)}	{Home, Tenary_Mor, Shre-e_bangla_road, Modhu_Bazar, Abahoni_Math, 32_No, Jhigatola}
Science_Lab	17	29	46	{Panthopoth(42), Kola_Baghan(37), 8_No_Bridge(30), Mohammed_Pur(47)}	{Home, Tenary_Mor, Shre-e_bangla_road, Modhu_Bazar, Abahoni_Math, 32_No, Jhigatola, Science_Lab}
Panthopoth	4	42	46	{UAP(46), Kola_Baghan(51), 8_No_Bridge(30), Mohammed_Pur(47)}	{Home, Tenary_Mor, Shre-e_bangla_road, Modhu_Bazar, Abahoni_Math, 32_No, Jhigatola, Science_Lab, Panthopoth}
UAP	0	46	46	{Kola_Baghan(51), 8_No_Bridge(30), Mohammed_Pur(47)}	{Home, Tenary_Mor, Shre-e_bangla_road, Modhu_Bazar, Abahoni_Math, 32_No, Jhigatola, Science_Lab, Panthopoth, UAP}

Table 2: A* Search (Using Straight Line Cost)

We've reached the goal node. Therefore the A* Optimal path from Home to UAP: Home → Tenary_Mor → Jhigatola → Science_Lab → Panthopoth → UAP. Total Cost: 46

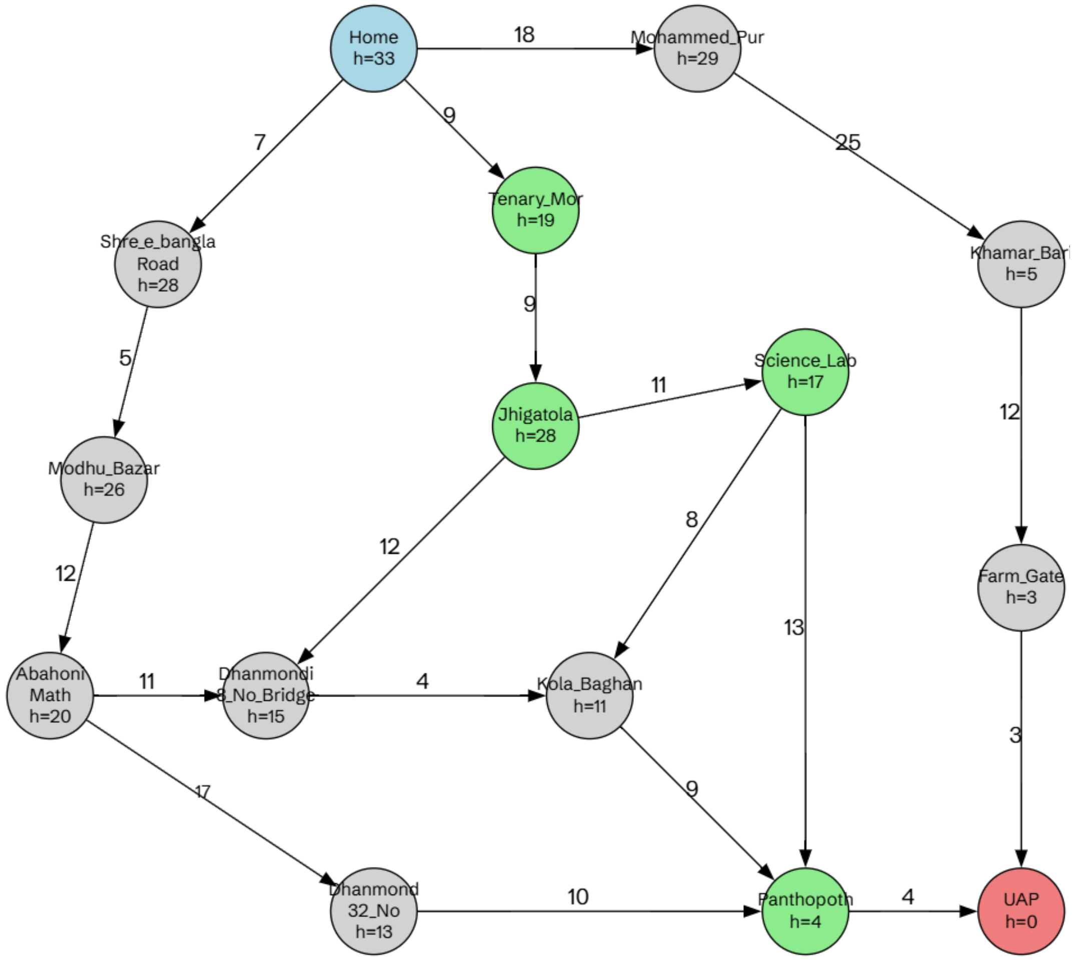

Figure 3: Graph Highlighted Optimal Path

VII. Code Implementation & Github Link

Algorithm for A* search-

Step I: Initialize an **open list** (priority queue) with the start node.

Step II: Initialize an **empty closed list** to store visited nodes.

Step III: While the open list is not empty:

Pick the node with the **lowest f(n)** from the open list.

If this node is the **goal node**, return the path and cost.

Otherwise, expand this node (explore its neighbors).

For each neighbor:

Compute g(n), h(n), and f(n).

If the neighbor is not in the open/closed list → add it.

If it is already there with a higher cost → update with the lower cost.

Move the current node to the **closed list**.

Step IV: If the open list becomes empty → No path exists.

In my python implementation-

- I've used a python dictionary to store the Graph.
- Each key is a node(location).
- Values are sets of adjacent vertices along with the traveling cost.
- The graph is stored as an adjacency list.

GitHub Link-

A* Implementation

VIII. Conclusion

This experiment has successfully demonstrated how the A* search algorithms can be utilized in obtaining the solution for a real-world shortest-path problem.

- The graph model enabled modeling various paths from Home to UAP.
- A* was able to incorporate the actual traveling cost (g) and the heuristic estimate (h) in managing the search.
- Optimal route discovered was: Home → Tenary_Mor → Jhigatola → Science_Lab → Panthopoth → UAP
- Total Cost for Travel = 46

This confirms the fact that A* is a perfect algorithm for pathfinding work for navigation systems, robotics, and decision-making on computers.