

UNIVERSITY OF ASIA PACIFIC

Department of Computer Science of Engineering



Project-1:

Technical Report

Course code: CSE(404)

Course Title: Artificial Intelligence and Expert System Lab

Submitted By:

Anik Chandra

ID : 18101070

Sec: B

Submitted To:

Dr. Nasima Begum

Assistant Professor

CSE, UAP

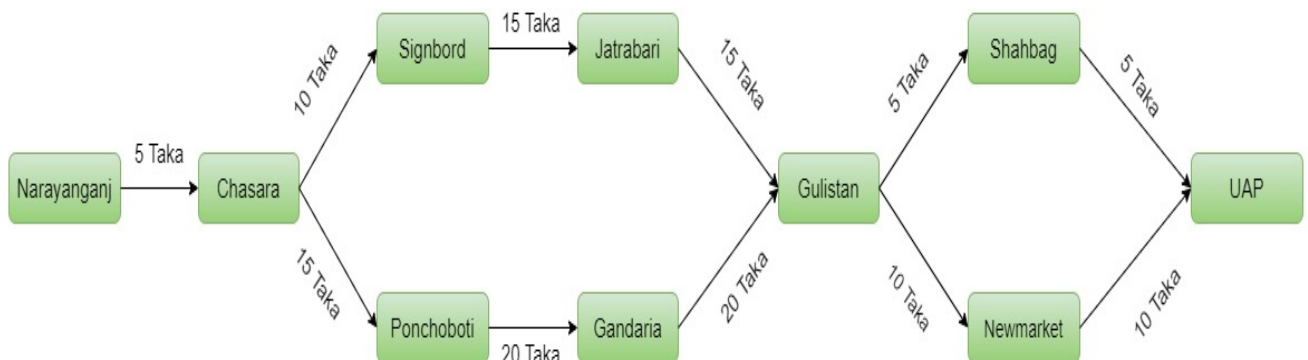
Index

Serial No.	Content Name	Page No.
1	Objective	3
2	Designed Map	3-4
3	Search tree of the map	4-6
4	Implementation	7-10
5	Result analysis	10-14
6	conclusion	15

Objective: In this project the main idea is implementation of a small map (My own home to UAP) using A* Search Algorithms. And in this project we try to find an optimal result or path from my own home to UAP. In this case the start node home(Narayanganj) and goal node (UAP).

Design Map:

My own home(Narayanganj) to UAP road map design are



For implementation we convert location to alphabetic letters:

Narayanganj -- A

Chasara -- B

Signbord -- C

Ponchoboti -- D

Jatrabari -- E

Gandaria -- F

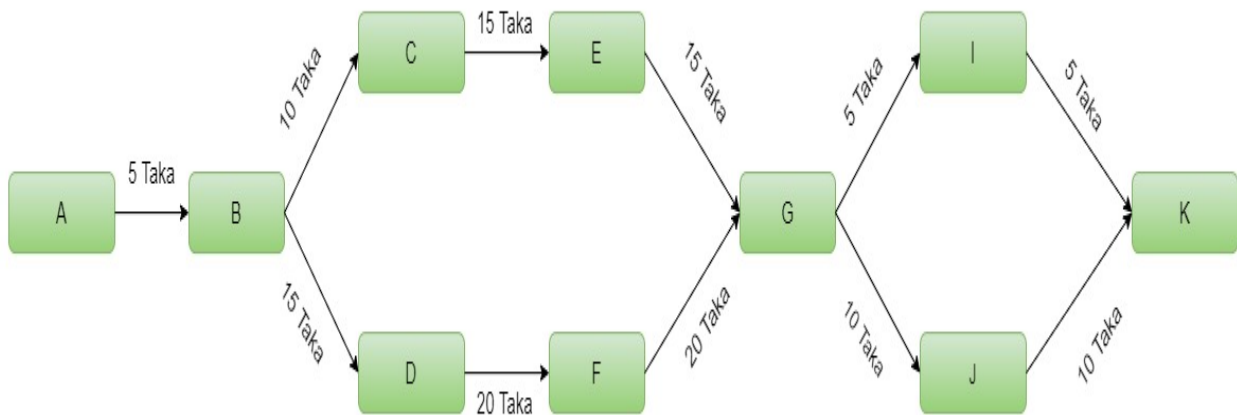
Shahbag -- I

Newmarket -- J

UAP -- K

Gulistan -- G

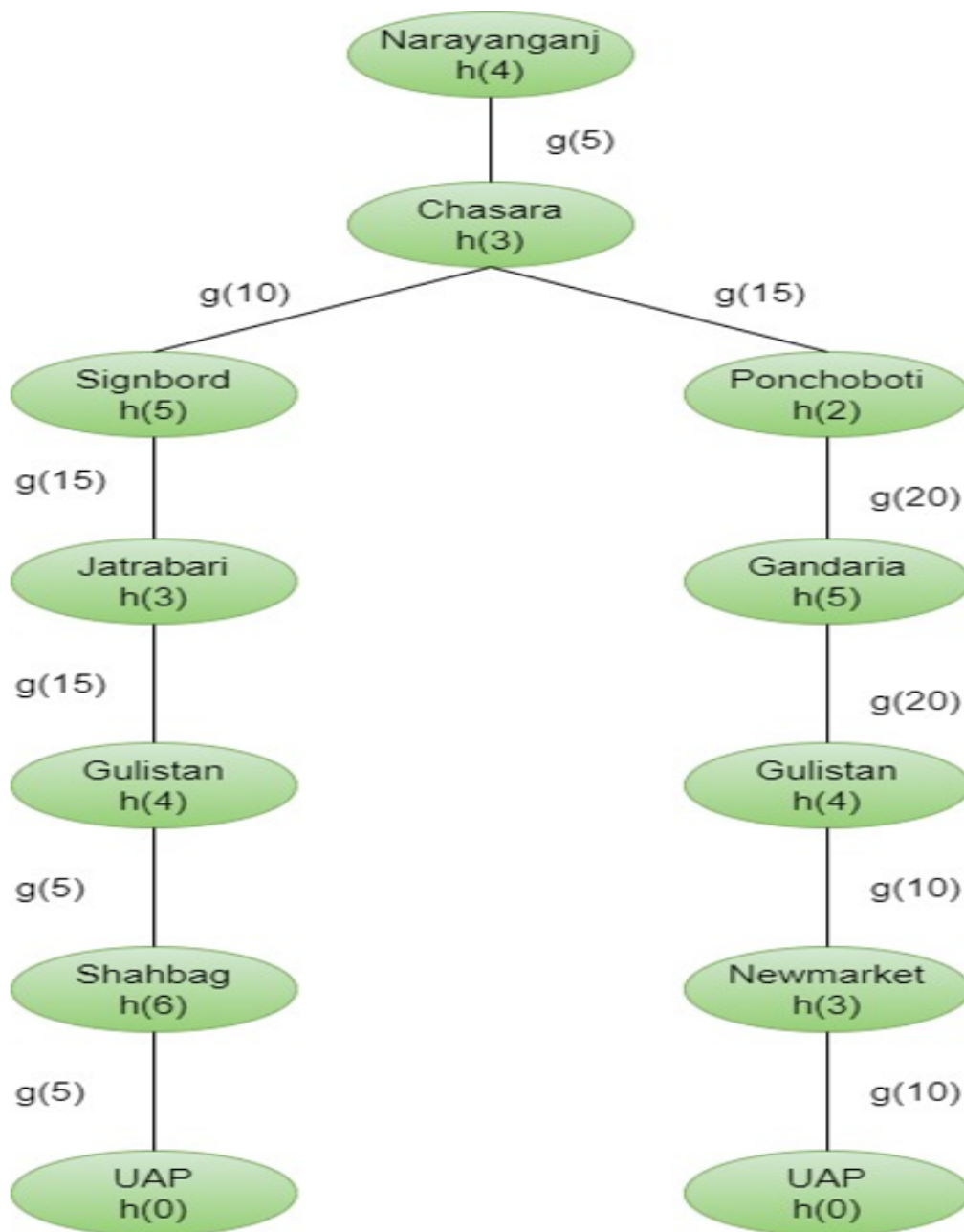
My own home(Narayanganj) to UAP road map design alphabetic letters are:



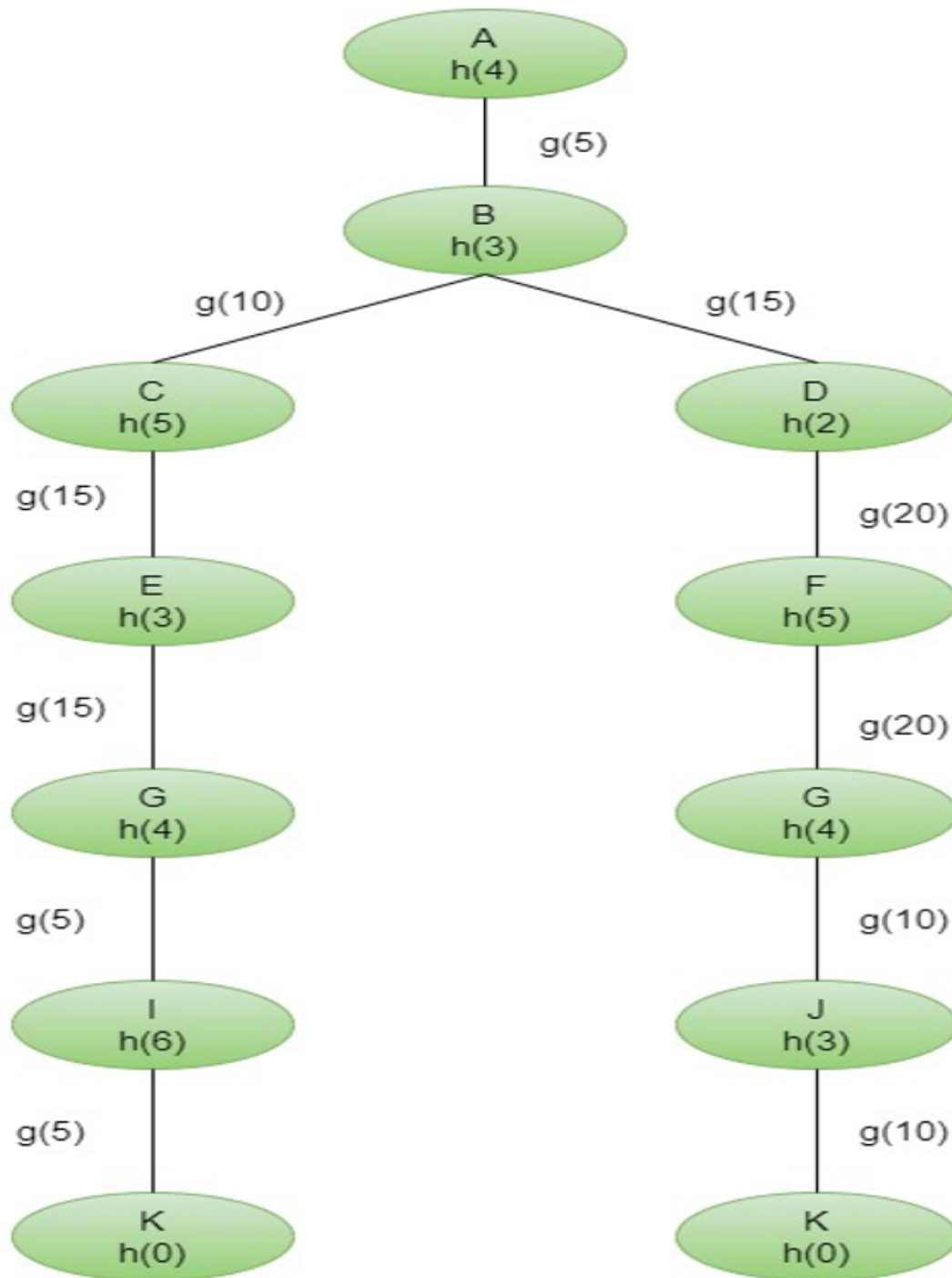
Search tree of the map:

Now convert State Space Graph to Search Tree according to location or alphabetic letters based. There $h(n)$ heuristic values and $g(n)$ path cost.

Location based Search Tree:



Alphabetic Letters based Search Tree:



Implementation :

For implementation I used Python programming language and Excel dataset and Colab notebooks.

In this part I discuss some points of my implementation and code.

First I discuss how to read an excel dataset in Colab. In this case i use python pandas libraries

First import pandas libraries and read the file “pd.read_csv()”.and put the excel content path in this function.

```
✓ [1] 1 import pandas as pd
      2 pd.set_option('mode.chained_assignment', None)
      3 df = pd.read_csv('/content/Input.csv')
      4 df
```

	0	1	2	3
0	A	B	5	3
1	B	C	10	5
2	B	D	15	2
3	C	E	15	3
4	D	F	20	5
5	E	G	15	4
6	F	G	20	4
7	G	I	5	6
8	G	J	10	3
9	I	K	5	0
10	I	K	10	0

Then i rename the data-frame ['parent','child','g(n)','h(n)] for implementation part the visualize is

```
✓ [5] 1 df.columns = ['Parent', 'Child', 'g(n)', 'h(n)']
    2 df['g(n)'] = df['g(n)'].astype('int32')
    3 df['h(n)'] = df['h(n)'].astype('int32')
    4 df
```

	Parent	Child	g(n)	h(n)
0	A	B	5	3
1	B	C	10	5
2	B	D	15	2
3	C	E	15	3
4	D	F	20	5
5	E	G	15	4
6	F	G	20	4
7	G	I	5	6
8	G	J	10	3
9	I	K	5	0
10	I	K	10	0

Then i implement a main logic or function for find a start node to goal node and their 'parent','child','g(n)','f(n)','cost'.In this part show that the which way to reach goal node(UAP).

Idea: Avoid expanding paths that are already expensive but expands most promising paths first

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

$g(n)$ actual cost to reach n

$h(n)$ estimated cost from n to goal

$f(n)$ estimated total cost of path through n to goal

```

1 path_cost = 0
2 while True:
3     if header==goal:
4         break
5     sd = sdf[sdf.Parent==header]
6     sd['cost']=sd[['g(n)', 'h(n)']].sum(axis=1)
7
8     sd=sd.sort_values(by='cost',ascending=True).reset_index(drop=True)
9     print(sd)
10    header = sd.loc[0].Child
11    path_cost+=sd.loc[0]['g(n)']#.value
12    path.append(header)

```

```

Parent Child g(n) h(n) cost
0      A    B     5     3     8
Parent Child g(n) h(n) cost
0      B    C    10     5    15
1      B    D    15     2    17
Parent Child g(n) h(n) cost
0      C    E    15     3    18
Parent Child g(n) h(n) cost
0      E    G    15     4    19
Parent Child g(n) h(n) cost
0      G    I     5     6    11
1      G    J    10     3    13
Parent Child g(n) h(n) cost
0      I    K     5     0     5
1      I    K    10     0    10

```

Then print the optimal path or result (form home(Narayanganj) to UAP)

```

1 print('Shortes path are->')
2 for i in range(len(path)-1):
3     print(dic[path[i]], end=' -> ')
4 print(dic[path[len(path) - 1]])
5

```

Shortes path are->
 Narayanganj -> Chasara -> Signboard -> Jatrabari -> Gulistan -> Shahbag -> UAP

Then print the total path cost (form home(Narayanganj) to UAP)

```

1 print('Total path cost is ',path_cost+list(sdf[sdf['Child']==goal]['h(n)'])[0])

```

Total path cost is 55

Result Analysis:

The heuristic value of A,B,C,D,E,F,G,I,J and K will be as follows-

Id = 70

$h(A) = id \% 4 + 2 = 4$

$h(B) = id \% 5 + 3 = 3$

$h(C) = id \% 6 + 1 = 5$

$$h(D) = id \% 5 + 2 = 2$$

$$h(E) = id \% 4 + 1 = 3$$

$$h(F) = id \% 4 + 3 = 5$$

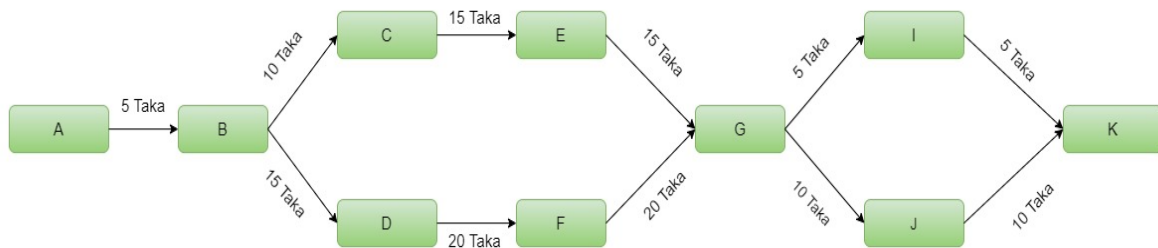
$$h(G) = id \% 5 + 4 = 4$$

$$h(I) = id \% 6 + 2 = 6$$

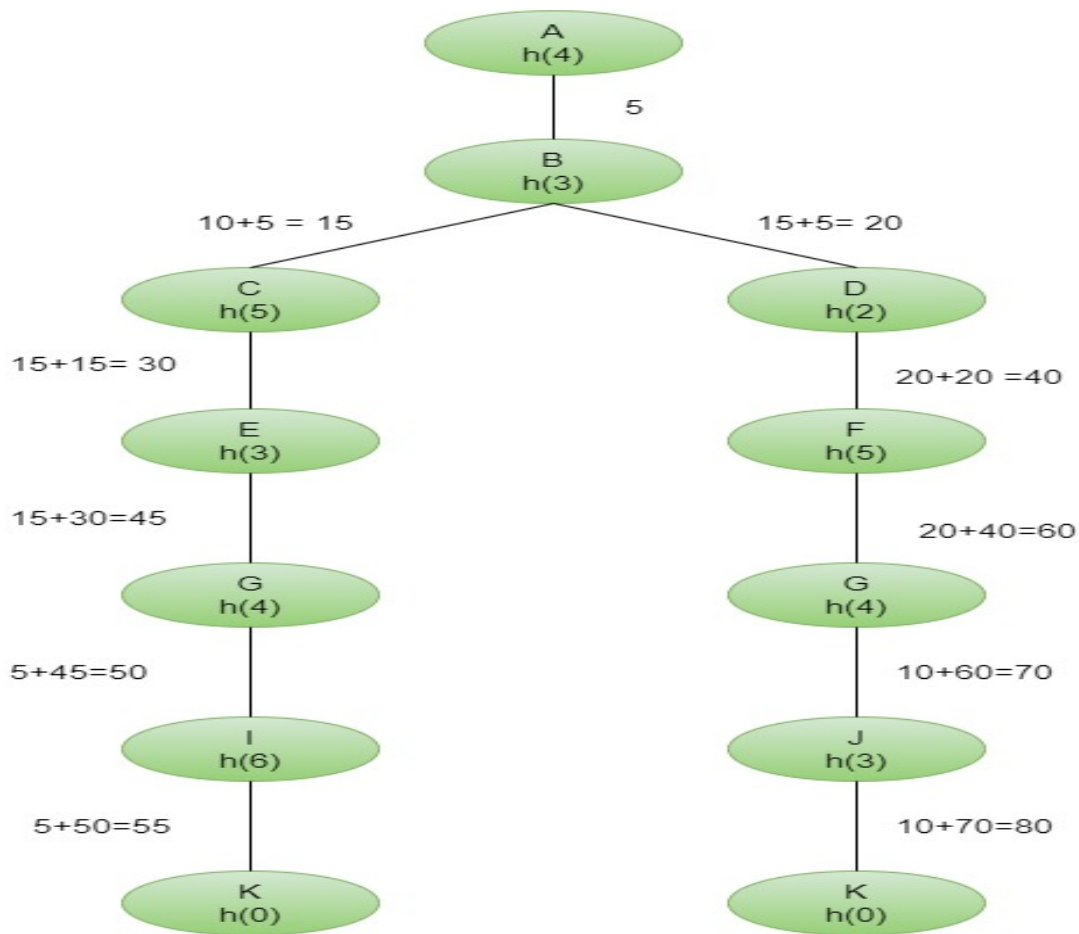
$$h(J) = id \% 5 + 3 = 3$$

$$h(K) = 0$$

The state space graph is:



The Corresponding search tree:



Now we will apply A* search Algorithm on the search tree

Here,

OF = Open fringe

CF = Close fringe

$g(n)$ actual cost to reach n node

$h(n)$ estimated cost from n to goal

$f(n)$ estimated total cost of path through n to goal

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

Iteration	Path	g(n)	h(n)	f(n)	OF	CF
Init	A	0	4	4		
1	A->B	5	3	8	B(8)	A(4)
2	A->B->C A->B->D	10 15	5 2	15 17	C(15) D(17)	A(4) B(8)
3	A->B->C->E A->B->D	15 15	15 2	30 17	E(30) D(17)	A(4) B(8) C(15)
4	A->B->C->E A->B->D->F	15 20	15 20	30 40	E(30) F(40)	A(4) B(8) C(15) D(17)
4	A->B->C->E->G A->B->D->F	15 20	30 20	45 40	G(45) F(40)	A(4)B(8) C(15)D(17) E(30)
5	A->B->C->E->G A->B->D->F->G	15 20	30 40	45 60	G(45) G(60)	A(4)B(8) C(15)D(17) E(30)F(40)
6	A->B->C->E->G ->I A->B->D->F->G	5 20	45 40	50 60	I(50) G(60)	A(4)B(8) C(15)D(17) E(30)F(40) G(45)

7	A->B->C->E->G ->I->K A->B->D->F->G	5 20	50 40	55 60	k(55) G(60)	A(4)B(8) C(15)D(17) E(30)F(40) G(45)I(50)
7	In this iteration we find goal node K(UAP) in the close fringe so it will return the path A->B->C->E->G ->I->K and the optimal cost.				G(60)	A(4)B(8) C(15)D(17) E(30)F(40) G(45)I(50) K(55)

This calculation find an optimal path
A->B->C->E->G->I->K

And the minimum path cost are 55

Conclusion:

In this project we find an optimal way or path A->B->C->E->G->I->K (from Home(Narayanganj) to UAP).

Home(Narayanganj)->Chasara->Signbord->Jatrabari->Gulistan->Shahbag->UAP.

This is the optimal path or minimum path cost(55).

In this project my implementation result and calculation result are the same so I hopefully say that this optimal path is right.