2192 - CSE - 5360 - 001 - AI - 1

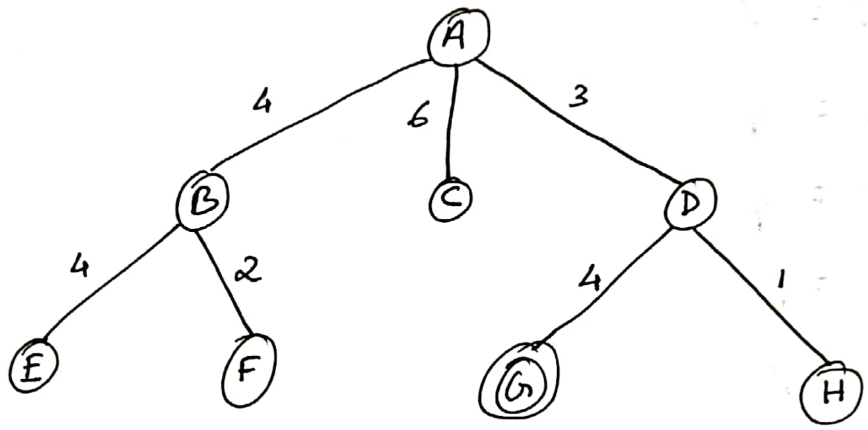
Assignment - 2

NAME : ESHAN DANAYAKAPURA JAGADEESH

UTA ID : 1001667159

NET ID : EXD7159

1).



Start node : A

Goal node : B

The order in which the goal will be visited for

a). Breadth - First - Search :

A B C D E F G

b). Depth - First - Search :

A B E F C D G

c). Iterative - Deepening - Search :

Iteration 1 : A

Iteration 2 : A B C D

Iteration 3 : A B E F C D G

d). Uniform - Cost - Search :

A D B H F C G

$g(A) = 0$

$g(D) = 3$

$g(B) = 4$

$g(H) = 4$

$g(F) = 6$

$g(C) = 6$

$g(G) = 7$

$g(E) = 8$

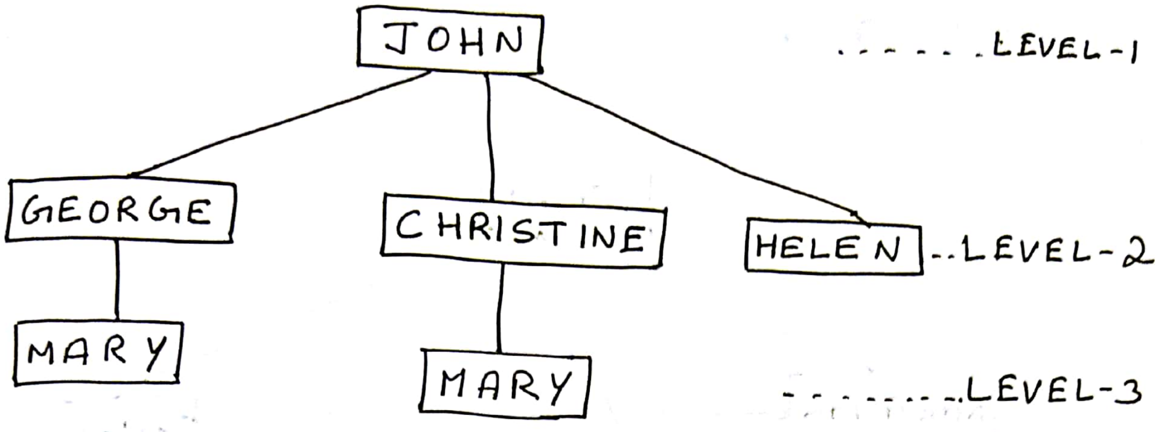
2]. The search strategies guaranteeing to find the correct number of degrees of separation between any 2 people in the graph can be found by :-

→ Breadth - First - Search

→ Iterative - Deepening - Search

→ Uniform Cost Search

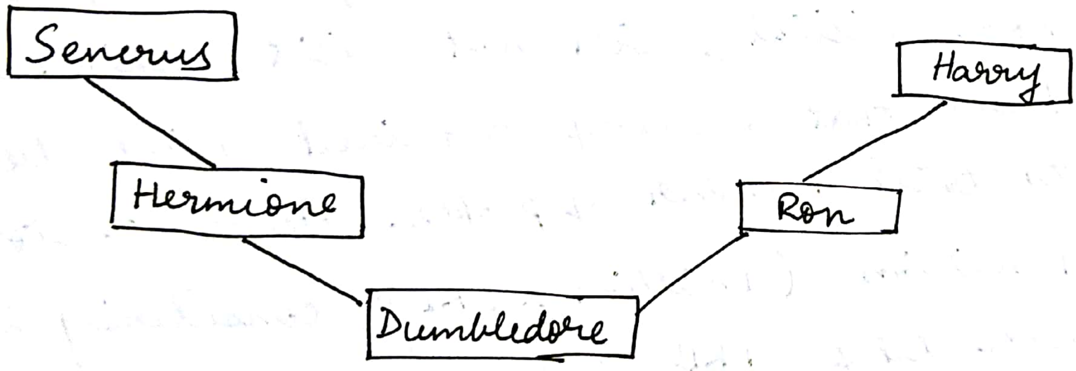
ii). First - three - levels of the search tree



→ With John as the starting point, the first 3 levels of the search tree is drawn.

→ Both George and Christine knows Mary. Hence one to one correspondance between nodes and vertices in SNG does not exist.

iii).



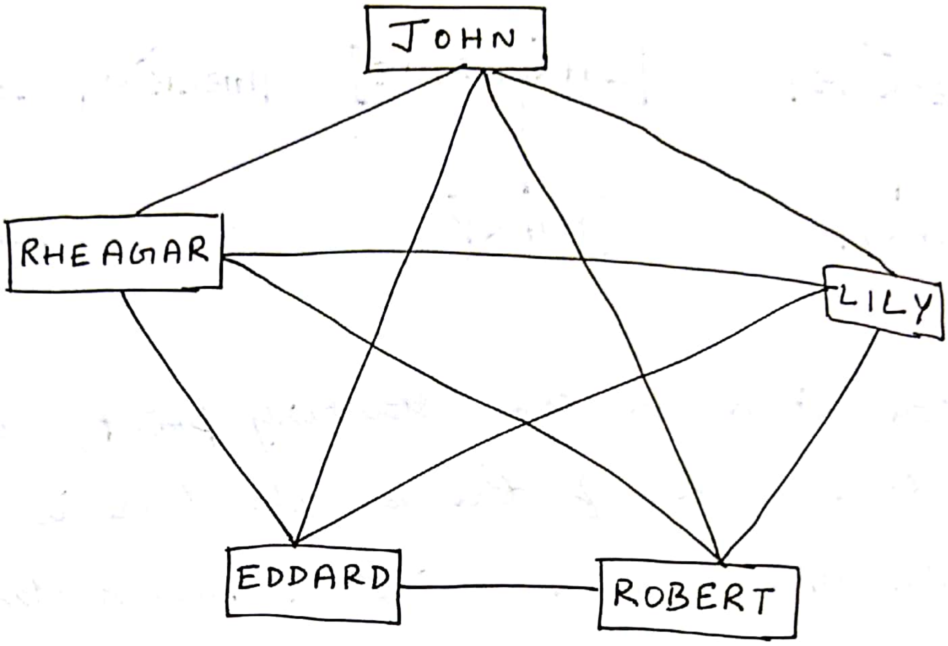
~~Johny~~

iv).

→ SNG containing exactly 5 people where atleast 2 people have 4 degrees of separation between them.

P. T. O

iv). SNG containing exactly 5 people, where all people have 1 degree of separation b/w them

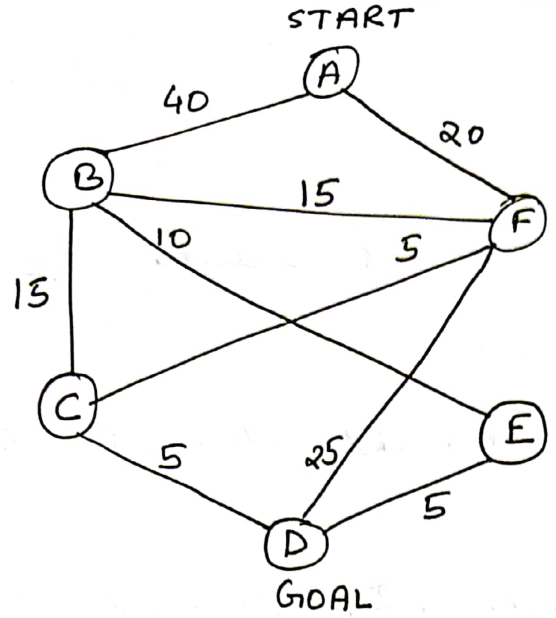


v). In the implementation of the Breadth-First search tree ~~by~~ by making sure node visited, will not visit again. We know that memory required will be equal to total number of people. Hence to store 1 million (1 million × 1 KB) considering every node takes 1 KB of memory will not exceed 1 million KB memory.

B)

P.T.O

3].



Heuristic 1:

→ h(A) = 50

Not admissable. It should be ≤ 30

→ h(B) = 35

Not admissable. It should be ≤ 15

→ h(C) = 5

It is admissable

→ h(D) = 0

It is admissable

→ h(E) = 45

Not admissable. It should be ≤ 5

→ h(F) = 10

It is admissable.

Heuristic 2:

→ $h(A) = 70$

Not admissable. It should be ≤ 30

→ $h(B) = 70$

Not admissable. It should be ≤ 15

→ $h(C) = 70$

Not admissable. It should be ≤ 5

→ $h(D) = 70$

Not admissable. It should be $= 0$

→ $h(E) = 70$

Not admissable. It should be ≤ 5

→ $h(F) = 70$

Not admissable. It should be ≤ 10

Heuristic 3:

→ $h(A) = 40$

Not admissable. It should be ≤ 30

→ $h(B) = 20$

Not admissable. It should be ≤ 15

→ $h(C) = 5$

It is admissable

→ $h(D) = 0$

It is admissable

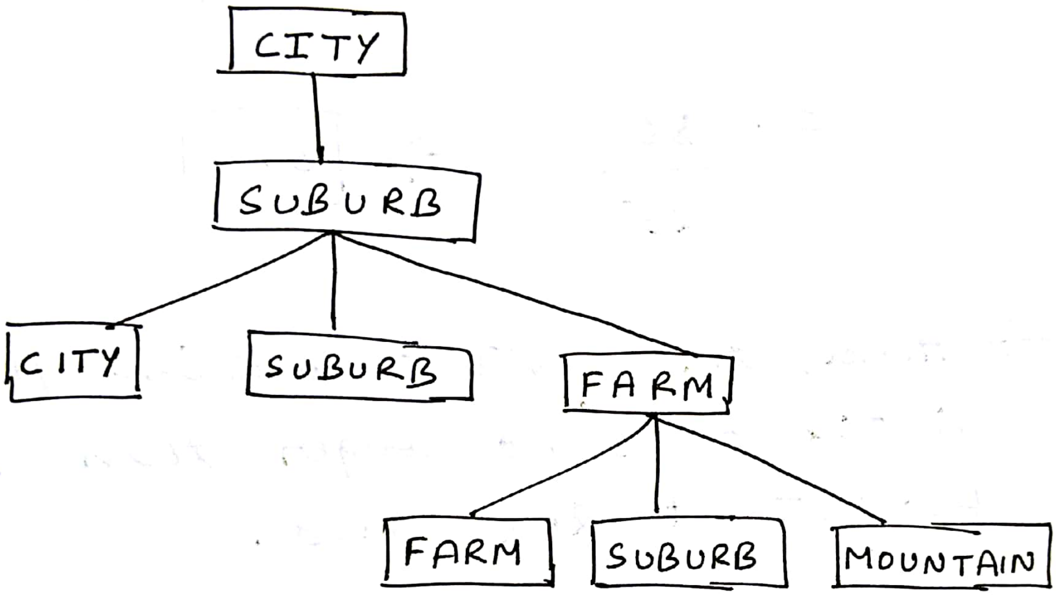
→ $h(E) = 5$

It is admissable

→ $h(F) = 20$

It is not admissable. It should be ≤ 10

4]. Search space where each state can be a city, Suburb, farmland or mountain



The highest possible value for each n in h(n) i.e The best possible admissable heuristic are as follows :

→ h (City) = 3

→ h (Suburb) = 2

→ h (Farm) = 1

→ h (Mountain) = 0

5]. Total possible moves = 80

Possible Blank locations = 25

Average Branching factor =

$$\frac{(\text{Total Possible moves})}{(\text{Possible Locations of Blanks})}$$

$$= \frac{80}{25} \Rightarrow \boxed{3.2}$$

→ Hence, the shortest location for some initial states no longer than 100 and few atmost 208 moves.

→ Taking Branching factor, $b = 3.2$

Depth of least cost solution, $d = 208$

Maximum depth, $m \approx 10^{25}$ states

→ Calculating the space complexity for strategies:

Breadth - First - Search :

$= b^d$

$= 3.2^{208}$ nodes

Depth - First - Search:

$= bm$

$= 3.2 \times 10^{25}$ nodes

Uniform Cost search:

$= b^d$

$= 3.2^{208}$ nodes

Iterative deepening search

$= bd$

$= 3.2 \times 208 \Rightarrow 665.6$ nodes

Storing 1 search node = 1KB of memory

a). None of the strategies can satisfy the condition of 50KB memory to store search nodes

b). Iterative Deepening search space complexity as per the calculation is 665.6KB and hence guarantees that we will never need more than 1200KB of memory to store search nodes.

All others will take more memory.

6). Consider $h(n)$ = Euclidean distance

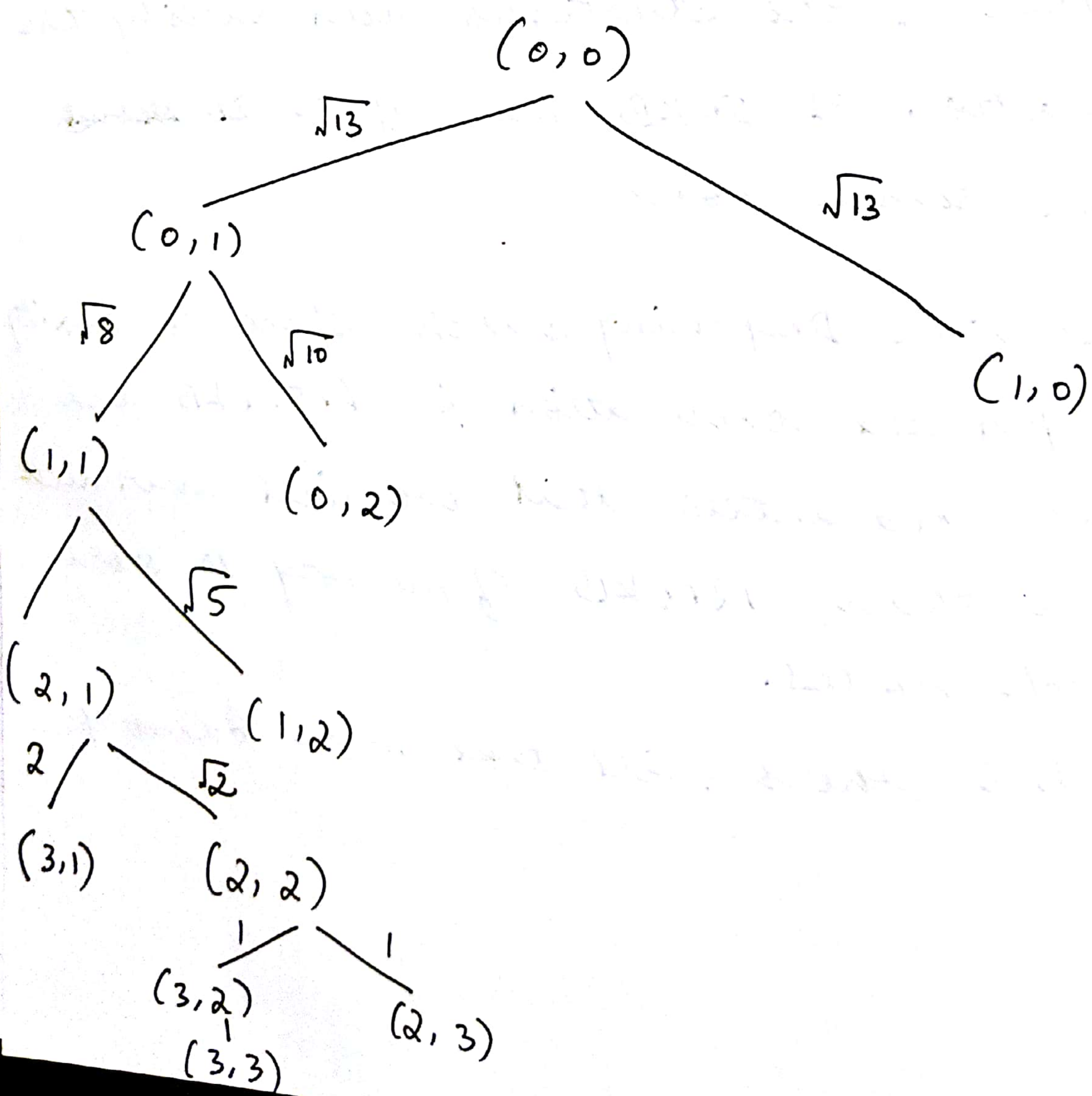
Euclidean distance b/w $(P_1 P_2)$ & (q_1, q_2) is

$$\sqrt{(P_1 - q_1)^2 + (P_2 - q_2)^2}$$

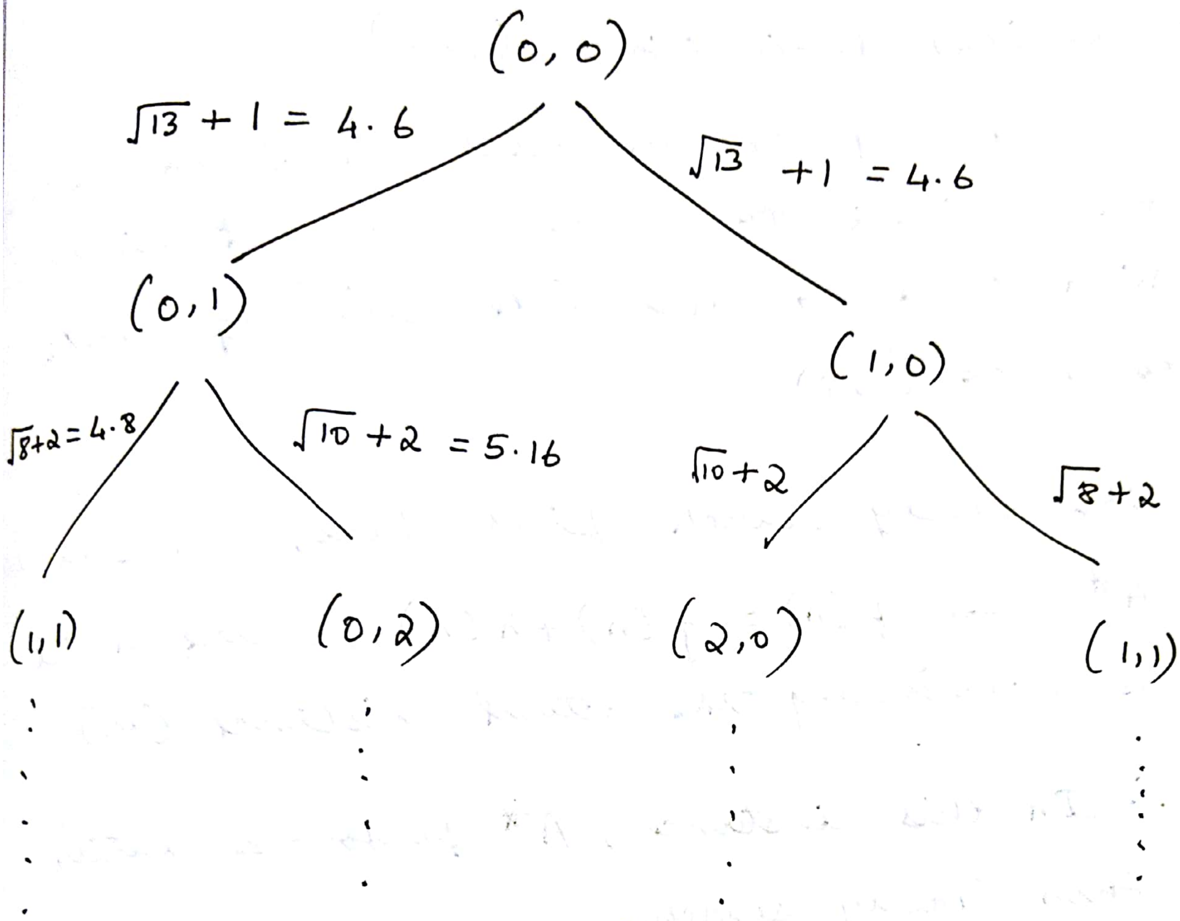
a). Consider Fig 5

→ Consider (0,0) & (0,1) as start and end Greedy search and A* performs the same and the number of nodes visited are the same.

GREEDY SEARCH h(n)



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



→ We can observe from above that the number of nodes visited by the A* strategy is more than Greedy search

→ Hence Greedy search will perform better than A*

→ Hence by considering all the available conditions for figure 5, Greedy search always performs better than or equal as A* depending on the start and end states.

6]. Consider Fig 6.

→ Consider destinations (3,5)

The greedy search uses h(n) to calculate the heuristic to reach (3,5) but fails since there no available driving route to node (3,5)

→ The greedy search fails here, where as the A* uses $f(n) = g(n) + h(n)$, where in it is considering the actual distance (∞)

→ ∴ In this instance, A* performs better than Greedy search.

→ By evaluating all the points,

Sometimes Greedy search will perform better than A* and sometimes both will perform similarly.

∴ Considering all the conditions for Fig 6, Greedy search will perform sometimes better, sometimes worse and sometimes the same as A* depending on the start and end states