Architha Harinath
1001657433

Assignment-2.

1) i) Breadth first search:

A → B → C → D → E → F → G.

ii) Depth first search.

A → B → E → F → C → D → G.

iii) Iterative deepening search:

Depth 0 : A

Depth 1 : output A → B → C → D

Depth 2 : output A → B → E → F → C → D → G

iv) uniform cost search

Step 1 : A(0)
Step 2 : A(0) D(3)
Step 3 : A(0) D(3) B(4)
Step 4 : A(0) D(3) B(4) H(5)
Step 5 : A(0) D(3) B(4) H(5) E(5)
Step 6 : A(0) D(3) B(4) H(5) E(5) C(6)
Step 7 : A(0) D(3) B(4) H(5) E(5) C(6) G(7)

Total cost to reach G is 7

iv) ~~uniform cost search~~

2)

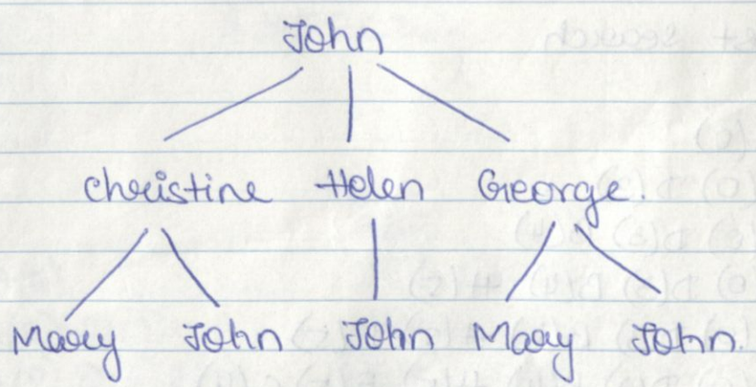
i) In the social Network graph where people are connected to one another the best strategy that would provide the correct number of degree would be Breadth first search and uniform cost search.

Breadth first search does the searching level by level so number of degrees will be known.

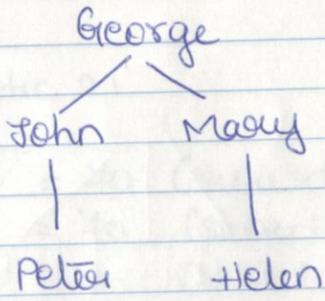
uniform cost search also uses the optimal strategy of finding cost where the search is also level by level.

DFS and Iterative deepening would find because it searches depth wise and cannot calculate degree.

ii)



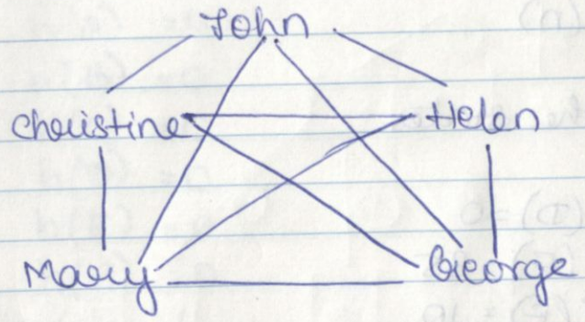
iii)



Helen and peter are having 4 degrees of seperation since

peter → John → George → Mary → Helen and vice versa.

iv)



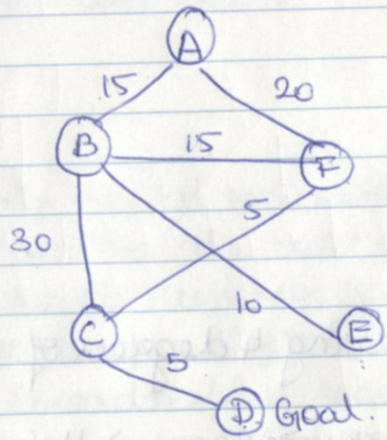
Each node in SNG connects with the other node in SNG with degree 1.

v) For breadth first search algorithm for the memory not to exceed more than 1GB we need to check the fringe for the destination.

If the destination is present in the fringe we need not have to expand all the nodes.

That way we can make sure storage does not exceed 1GB.

3)



For an heuristic to be admissable the condition is

$$h(n) \leq h^*(n).$$

$h^*(n)$ values of all the states

$h^*(A) = 30$	$h^*(D) = 0$
$h^*(B) = 25$	$h^*(E) = 35$
$h^*(C) = 5$	$h^*(F) = 10.$

Heuristic 1

$h(A) = 50$ (should be 30 or less than that)
$h(B) = 35$ (shoud be 25 or less than that)
$h(C) = 5$
$h(D) = 0$
$h(E) = 45$
$h(F) = 10.$

Heuristic 2:

h(A) = 70 (should be less than 30 or 30)
h(B) = 70 (should be 25 or less than 25)
h(C) = 70 (should be 5 or less than 5)
h(D) = 70 (should be 0 because goal heuristic is 0)
h(E) = 70 (should be 35 or less than 35)
h(F) = 70 (should be 10 or less than 10)

Heuristic 3:

h(A) = 0
h(B) = 0
h(C) = 0
h(D) = 0
h(E) = 0
h(F) = 0.

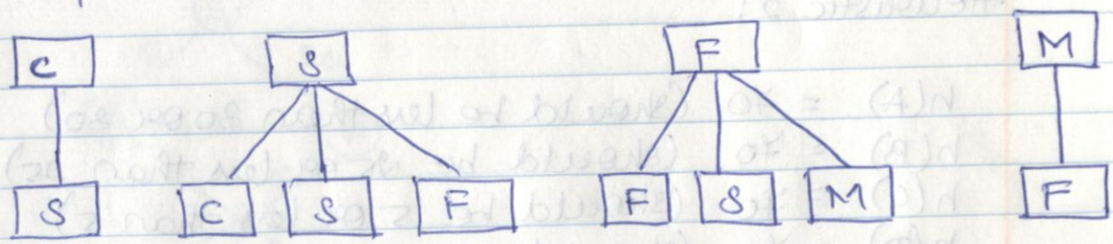
All the heuristic values given is admissable because it is less than $h^*(n)$ value.

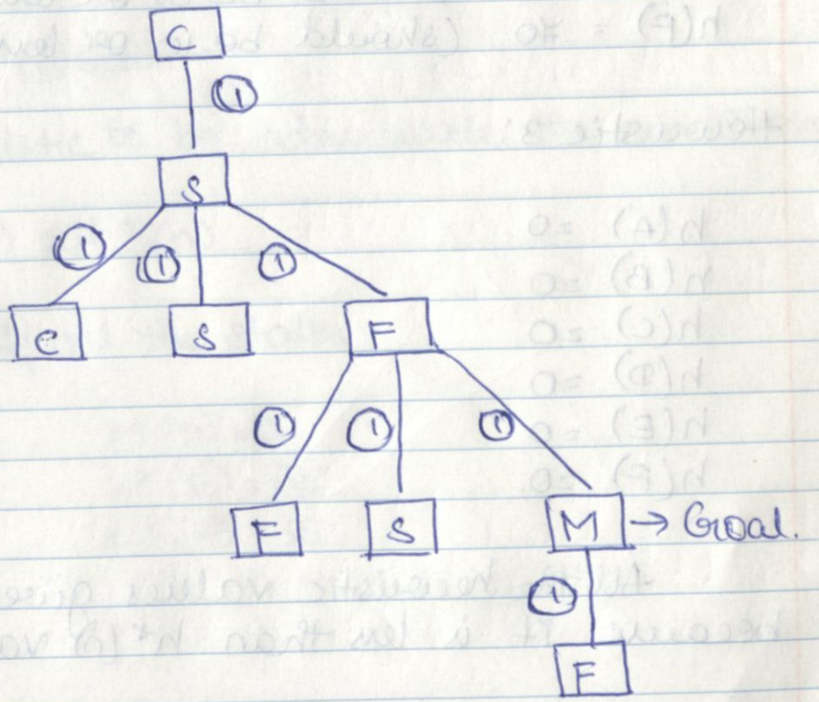
4) Search Space
= {city, suburb, farmland, mountain}

Goal is to reach Mountain.

city is represented C
Suburb is represented S
Farmland is represented F
Mountain is represented M

Rules represented as tree structure





cost to move from one state to other.

C → S → F → M admissible heuristic is 3 or less than that

S → F → M admissible heuristic is 2 or less than 2 for that route.

F → M Admissible heuristic is 1 or less than 1 for the route.

M Goal state always has 0 as heuristic.

5) In a 24 puzzle problem there are nodes that has 2, 3, 4 as branching factor.

corner nodes has branching factor of 2 per corner
so $4 \times 2 = 8$.

The nodes that have 3 brances the branching factor
$3 \times 12 = 36$.

The Rest other nodes in the puzzle have branching factor
$9 \times 4 = 36$.

Average branching factor $= 8 + 36 + 36 / 25$
$= 3.2$

Worst case senario $= 4$

Depth value is given by $100 \leq d \leq 208$.

Space complexity for various uninformed searches

Breadth first search $= O(b^{d+1})$
$= (4)^{209} > 50KB$

Depth first search $= O(b \times m)$
$= 4 \times 208 = 665 > 50KB$

Iterative deepening search $= O(b \times d)$
lowest depth $= 4 \times 100 = 400 > 50KB$
Maximum depth $= 4 \times 208 = 832 > 50KB$

Uniform cost search $= O(b^{\wedge(C^*/\varepsilon)})$

The value will be greater than 50 KB because its exponential of branching factor.

a) From the above calculated space complexity values none of the values are less than or equal to 50 KB. So none of the methods can be used.

b) From the above calculated ~~search~~ space complexity values only Depth first search and iterative deepening search uses less than 1200 KB in all the senarios.

6) For figure 5 if we choose following nodes from edge to center, corner to center or any case A* and greedy algorithm calculates the straight line distance to reach the destination.

Therefore for figure 5 both A* and greedy always performs the same irrespective of start and end nodes.

6) In figure 6 A* chooses straight line distance to traverse from start to end node. But for greedy since it has few disconnected nodes in the middle it will choose the node nearest to the destination. And in some cases where all the nodes are connected greedy searches the same way as A*.

So for figure 6 greedy works worst than or similar to A*