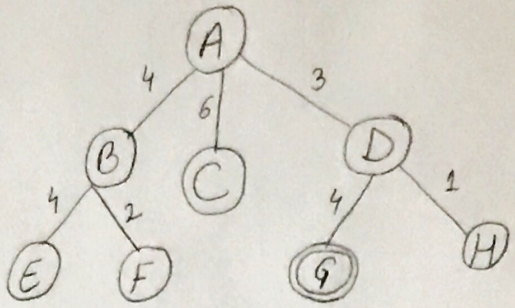
Artificial Intelligence [CSE-5360-002]

Assignment-2

Question-1)



- Breadth First search :-

  A → B → C → D → E → F → G

- Depth - First search :-

  A → B → E → F → C → D → G

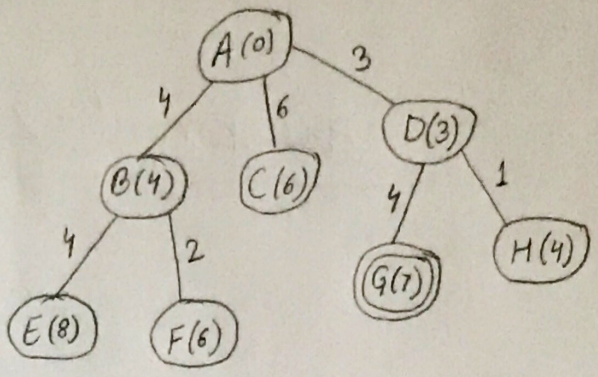
- Iterative Deepening search :-

  First Iteration - A

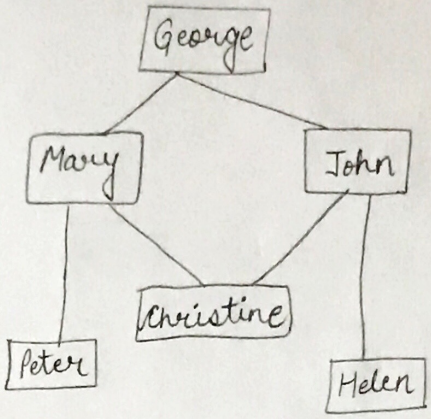
  Second Iteration - A → B → C → D

  Third Iteration - A → B → E → F → C → D → G

• Uniform cost search :-

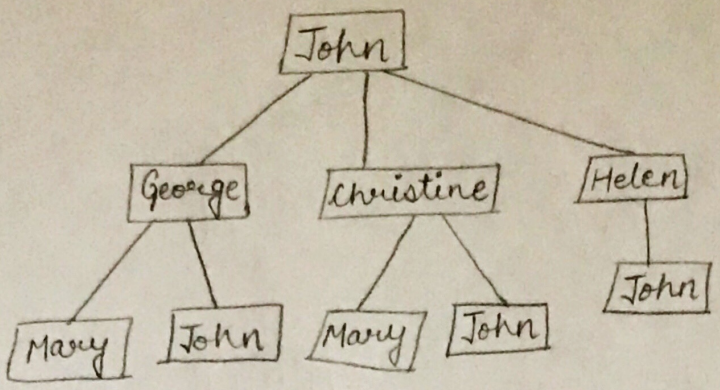


Question - 2)



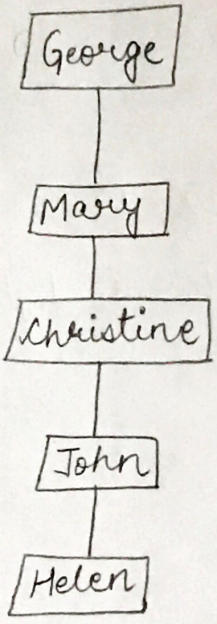
(i) Breadth first search (BFS) and Uniform cost search (UCS) are the algorithms which guarantee finding the correct number of degrees of seperation between any two people in the graph.

(ii)

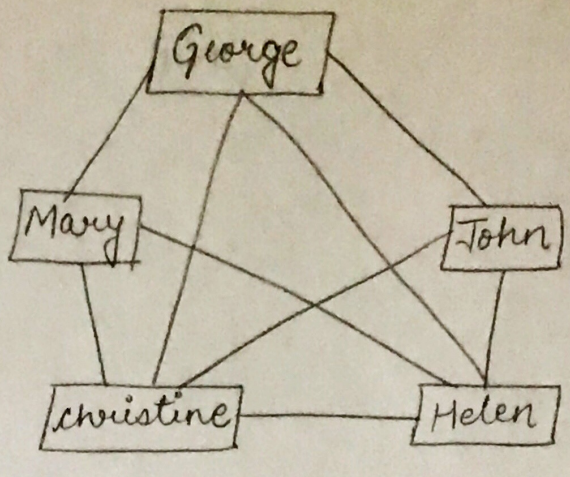


No, there is no one-to-one correspondence between nodes in the search tree and vertices in the SNG because the vertex John in SNG corresponds to multiple nodes in the search tree.

(iii)



(iv)



(v) Every node in the search tree takes 1 Kb of memory. So, 1 million (10^6) people will require 1 million Kb i.e. 10^6 Kb of memory.

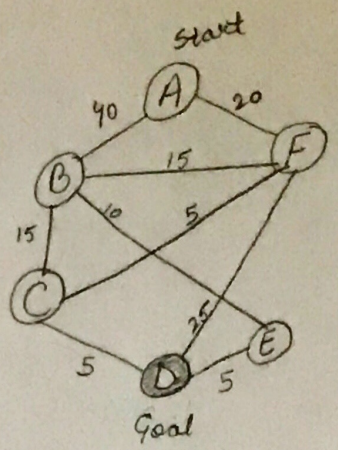
Space Complexity of BFS = b^{d+1}

where, $d = 5$

branch factor (b) = 10

So, it will take atmost 1 gb and will not exceed 1 gb.

Question - 3)



Calculation of $h^*(n)$:

$h^*(n)$ is the minimum cost to reach goal

$h^*(A) = 20 + 5 + 5 = 30$

$h^*(B) = 15 + 5 = 20$

$h^*(C) = 5$

$h^*(D) = 0$

$h^*(E) = 5$

$h^*(F) = 25$

$h(n) \leq h^*(n)$, this is admissible

Heuristic 1:

$h(A) = 50$	;	$50 > 30$,	$h(A) > h^*(A)$
$h(B) = 35$	;	$35 > 20$,	$h(B) > h^*(B)$
$h(C) = 5$	;	$5 = 5$;	$h(C) = h^*(C)$
$h(D) = 0$	;	$0 = 0$,	$h(D) = h^*(D)$
$h(E) = 45$	;	$45 > 5$,	$h(E) > h^*(E)$
$h(F) = 10$	;	$10 < 25$,	$h(F) < h^*(F)$

Therefor $h(C)$, $h(D)$ and $h(F)$ are admissible.
$h(A)$, $h(B)$, $h(E)$ are not admissible.

To make them admissible we need to change the values to $h(A) = 30$, $h(B) = 20$ and $h(E) = 5$.

Heuristic 2 :-

$h(A) = 70$; $70 > 30$, $h(A) > h^*(A)$

$h(B) = 70$; $70 > 20$, $h(B) > h^*(B)$

$h(C) = 70$; $70 > 5$, $h(C) > h^*(C)$

$h(D) = 70$; $70 > 0$, $h(D) > h^*(D)$

$h(E) = 70$; $70 > 5$, $h(E) > h^*(E)$

$h(F) = 70$; $70 > 25$, $h(F) > h^*(F)$

This heuristic is not admissible because all h values are greater then h^* values.

To make this heuristic admissible we need to change all the h values.

Admissible values are as follows :-

$h(A) = 30$

$h(B) = 20$

$h(C) = 5$

$h(D) = 0$

$h(E) = 5$

$h(F) = 25$

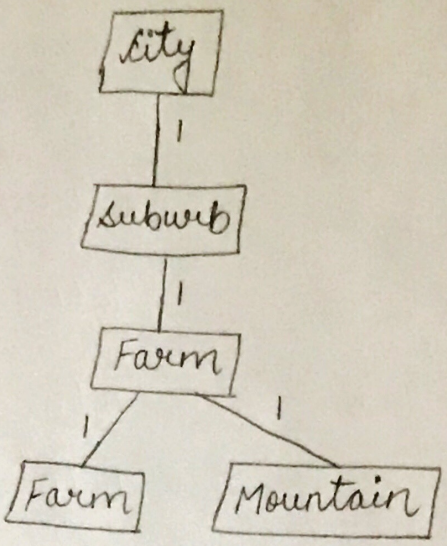
Heuristic - 3 :-

$h(A) = 40$;	$40 > 30$,	$h(A) > h^*(A)$
$h(B) = 20$;	$20 = 20$,	$h(B) = h^*(B)$
$h(C) = 5$;	$5 = 5$,	$h(C) = h^*(C)$
$h(D) = 0$;	$0 = 0$,	$h(D) = h^*(D)$
$h(E) = 5$;	$5 = 5$,	$h(E) = h^*(E)$
$h(F) = 20$;	$20 < 25$,	$h(F) < h^*(E)$

Therefore, $h(B)$, $h(C)$, $h(D)$, $h(E)$ and $h(F)$ are admissible.

$h(A)$ is not admissible. To make it admissible we need to change the value of $h(A)$ from 40 to 30. If $h(A)$ is less than or equal to 30 then $h(A)$ becomes admissible.

Question-4) According to rules successors are,



The best admissible heuristic with goal state as mountain is as follows :-

h (mountain) = 0

h (farm) = 1

h (suburb) = 2

h (city) = 3

Question-5)

The shortest solution is greater than 100 moves for some initial states.

The shortest solution is at most 208 moves for all initial states.

In any implementation storing a search node takes 1000 bytes i.e. 1KB of memory.

For Linear space complexity,

$b = 4$ [up, down, right, left]

$d = 101$ [more than 100 moves]
Best-case

So, according to Iterative Deepening search or DFS

Space complexity = $4 \times 101 \times 1\,KB$
$= 404\ KB$

$404\,KB > 50\,KB$

None of the methods can guarantee that we will never need more than 50KB of memory to store search nodes even with linear space complexity.

(b) Iterative Deepening search and Depth First Search can guarantee that you will never need more than 1200 KB of memory to store search nodes.
We conclude that,
Considering worst case,

$b = 4$, $d = 208$

Space complexity = $4 \times 208 \times 1\,KB$
$= 832\ KB$

$832\,KB < 1200\,KB$

Question - 6)

For figure 5, Greedy search always performs the same as A*, irrespective of the start and end states because both of them have same heuristic values and follow the same path.

For figure 6, Greedy search always performs worse than or the same as A*, depending on the start and end states because in one case where start state is (4,3) and end state is (4,4). A* search goes via (4,2), (3,2), (3,3) and (3,4) but Greedy search picks (5,3) and goes via (6,3), (6,4), (6,5), (5,5) and (4,5).