Nishant Chaudhary
1001563781

Task 3:

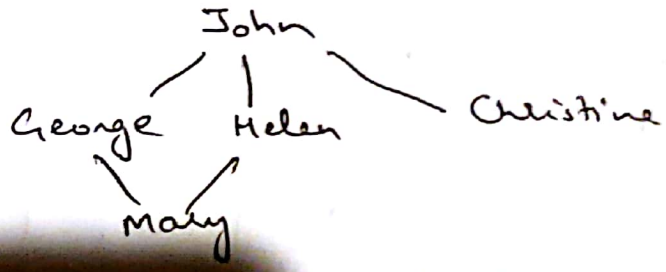
i) BFS → Yes, as it is complete & finite node connectivity. Also cost per step is 1.

Iterative Deepening → Yes, it is complete too & it also reduces the search memory utilization once found the Goal

Uniform cost search → Yes, It will be complete when the stepcost ≥ ∈.

DFS → No, if we have no track of visited nodes it might end up in an infinite loop. Also it might end with an wrong answer (non-optimized one).

ii) No, there isn't a one to one correspondence between the nodes.

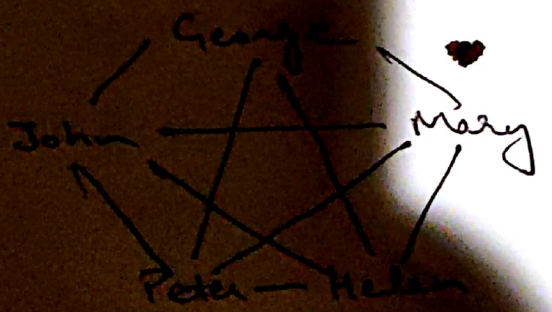


Also Peter is not a node in the search tree.

iii) George — Mary — Helen — John — Christine

George has a 4 degree of separation from Christine & Christine has a 4 degree of separation from George.

iv)



All of the 5 people have 1 degree of separation between them.

v) For this we need to keep track of the node addresses that have been visited & expanded. Say we called it 'visited'. Now before we put anything in our fringe we'll check whether this node address is in

the 'visited'. If it is, then we don't push that node into the fringe.
This way we won't exceed the limit.

Task 6:

a) Every node holds a size of 1 Kb in memory, so we can't have more than 50 nodes. But none of the search algorithms can guarantee to achieve 50 Kb of memory.

BFS → $O(b^{d+1})$
DFS → $O(bm)$
Iterative → $O(bd)$
Uniform → $O(b^{(C^*/\epsilon)})$

} space property

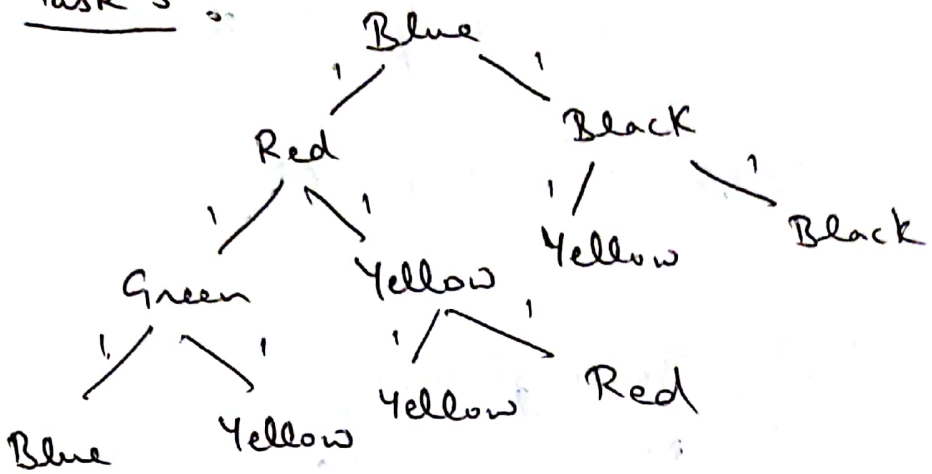
Nodes generated

BFS → $4^{(209)}$
DFS → ∞
Iterative → 832
Uniform → $4^{(209)}$

} No of nodes generated.

b) We can't have more than 832 nodes. According to the calculations shown above only Iterative Deepening search can solve the problem as it generates between 400 & 832 nodes so it won't exceed 832 Kb.

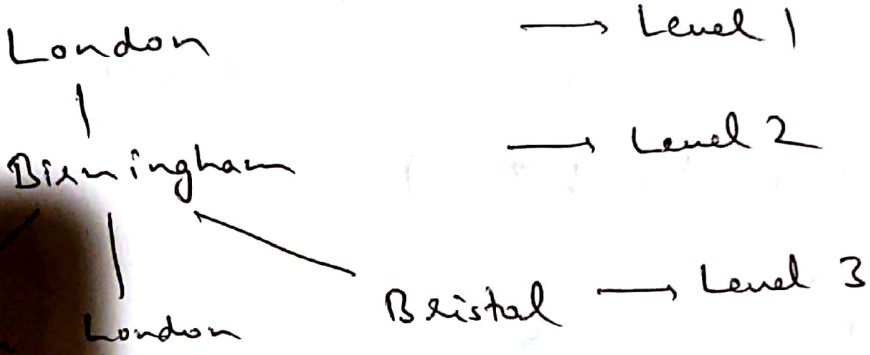
Task 5 :



h(red) = 3 h(blue) = 1 h(green) = 2

h(yellow) = 4 h(black) = 0

Task 2 :



Breadth First →

Dresden, Leipzig, Berlin, Nuremberg, Magdeburg

Level 1 Level 2 Level 3

Depth First →

Dresden, Berlin, Hamburg, Luebeck, Bremen

Level 1 Level 2 Level 3 Level 4

IDS →

Limit 0 → Dresden

Limit 1 → Dresden, Leipzig, Berlin

Limit 2 → Dresden, Leipzig, Nuremburg, Magdeburg, Berlin

Limit 3 → ~~Dresden, Leipzig, Nuremburg, Munich~~, Stuttgart

Limit 4 → ~~Dresden, Leipzig, Nuremburg, Munich~~

Uniform cost →

Dresden(0), Leipzig(119), Berling(204),
Magdeburg(119+125), Nuremburg(119+263)

Task 4:

Smallest cost from G to all other nodes

A → 17 B → 14 C → 10 D → 12

E → 7 F → 4 G → 0

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

Heuristic 1:

Non-admissible

Modify → h(A), h(B), h(D), h(F), h(G)

Modified → 17 14 12 4 0

Heuristic 2:

Non-admissible

Modify → h(A), h(B), h(C), h(D), h(E), h(F), h(G)

Modified → 17 14 10 12 7 4 0

Heuristic 3:

Non-admissible

Modify → h(G)

↓

Modified → 0

Heuristic 4:

Admissible

Heuristic 5:

Admissible