

①

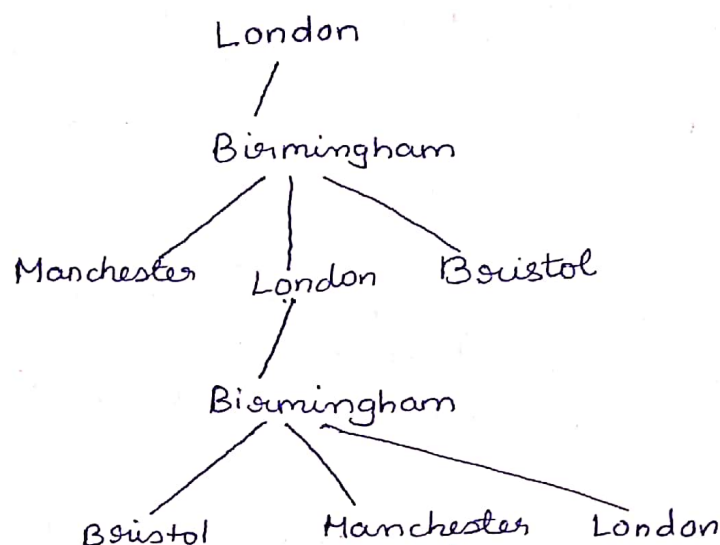
Name: Venkata Sai Manoj

Twarakavi

ID: 1001637682

→ Task 2

Solo:-



\* BFS - [Dresden, Leipzig, Berlin, Magdeburg, Nuremberg]

- 1) Dresden
- 2) Leipzig, Berlin
- 3) Berlin, Magdeburg, Nuremberg
- 4) Magdeburg, Nuremberg, Hamburg
- 5) Nuremberg, Hamburg, Hannover

\* DFS - [Dresden, Leipzig, Magdeburg, Hannover, Bremen]

- 1) Dresden
- 2) Leipzig, Berlin
- 3) Magdeburg, Nuremberg, Berlin
- 4) Hannover, Nuremberg, Berlin
- 5) Bremen, Hamburg, Kassel, Nuremberg, Berlin

\* IDS - [Dresden, Leipzig, Magdeburg, Nuremberg, Berlin]

- 1) Dresden
- 2) ~~Dresden~~ Leipzig, Berlin
- 3) Magdeburg, Nuremberg, Berlin
- 4) Nuremberg, Berlin
- 5) Berlin

\* UCS - [Dresden<sup>(0)</sup>, Leipzig<sup>(119)</sup>, Berlin<sup>(204)</sup>, Magdeburg<sup>(244)</sup>, Nuremberg<sup>(382)</sup>]

- 1) Dresden<sup>(0)</sup>
- 2) Leipzig<sup>(119)</sup>, Berlin<sup>(204)</sup>
- 3) Berlin<sup>(204)</sup>, Dresden<sup>(238)</sup>, Magdeburg<sup>(244)</sup>, Nuremberg<sup>(382)</sup>
- 4) Dresden<sup>(138)</sup>, Magdeburg<sup>(244)</sup>, Magdeburg<sup>(370)</sup>, Nuremberg<sup>(382)</sup>, Dresden<sup>(408)</sup>, Hamburg<sup>(495)</sup>

②

- 5.) Magdeburg(244), Magdeburg(370), Nuremberg(382), Dresden(408), Hamburg(495)
- 6.) Leipzig(369), Magdeburg(370), Nuremberg(382), Hannover(392), Dresden(408), Berlin(410), Hamburg(495)
- 7.) Magdeburg(370), Nuremberg(382), Hannover(392), Dresden(408), Berlin(410), Hamburg(495)
- 8.) Nuremberg(382), Hannover(392), Dresden(408), Berlin(410), Hamburg(495)

### → Task 3

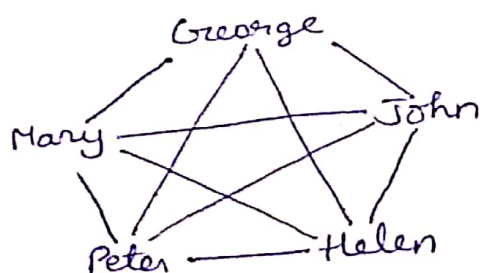
(i) IDS will guarantee correct degree of separation if initial depth is zero and then the increment value is 1. UCS and BFS also gives correct degree of separation if they work the same.

(ii) No, it will not change.

(iii) We cannot achieve one-to-one correspondence between the nodes in the search tree, as the SNG is an undirected graph.

(iv) Peter — Mary — George — John — Helen

(v)

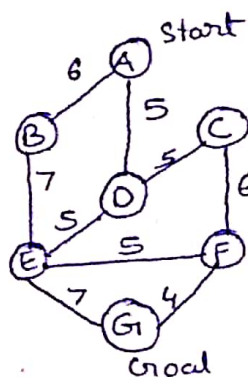


(vi) BFS could be implemented by not generating the successor nodes for those nodes that coincide to states that have already been visited. We could store an initial list of all visited states.

### → Task 4

Best Heuristic:

- $h(A) = 17$
- $h(B) = 14$
- $h(C) = 10$
- $h(D) = 12$
- $h(E) = 7$
- $h(F) = 4$
- $h(G) = 0$



③

H1 :- False

$$h(A) = 17$$

$$h(B) = 14$$

$$h(C) = 5$$

$$h(D) = 0$$

$$h(E) = 5$$

$$h(F) = 4$$

$$h(G) = 0$$

H2 :- False

$$h(A) = 17$$

$$h(B) = 14$$

$$h(C) = 10$$

$$h(D) = 12$$

$$h(E) = 7$$

$$h(F) = 4$$

$$h(G) = 0$$

H3 :- False

$$h(A) = 2$$

$$h(B) = 0$$

$$h(C) = 2$$

$$h(D) = 0$$

$$h(E) = 2$$

$$h(F) = 0$$

$$h(G) = 0$$

H4 :- True

H5 :- True

→ Task 5

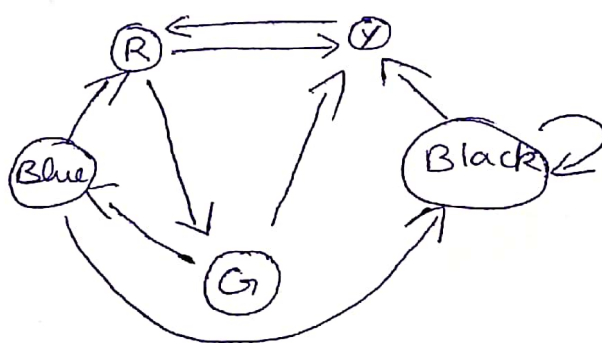
$$h(\text{Yellow}) = 4$$

$$h(\text{Red}) = 3$$

$$h(\text{Green}) = 2$$

$$h(\text{Blue}) = 1$$

$$h(\text{Black}) = 0$$



→ Task 6

Given;  $b=4$ ,  $d=100$ ,  $m=208$

$$\text{For BFS, space} = b^{d+1} = 4^{100+1}$$

$$\text{For DFS, space} = bm = 832$$

$$\text{For IDS, space} = bd = 400$$

$$\text{For UCS, space} = b^{(C/\epsilon)} = 4^{(C^*/\epsilon)}$$

- None of the methods guarantee that you will need never more than 100KB of memory to store search nodes.
- DFS and IDS methods can guarantee that you will never need more than 1000KB of memory to store search nodes.