

# \* Laboratory Practice II (Artificial Intelligence) - Group A - Experiment No. - 1

Name:- Kaustubh Shrikant Khabra.

Class:- Third Year Engineering.

Div:- A

ERP. Number:- 38

Department:- Computer Department

College:- AISSMS's IOIT.

Title:-

Depth First Search Algorithm and Breadth First Search Algorithm.

Aim:-

Implement depth first search algorithm and breadth first search algorithm. Use an undirected graph and develop a recursive algorithm for searching all the vertices of a graph or tree data structure.

Objective:-

1. To understand and implement DFS and BFS.
2. To learn and understand use of data structure used in implementation of DFS and BFS.

Theory:-

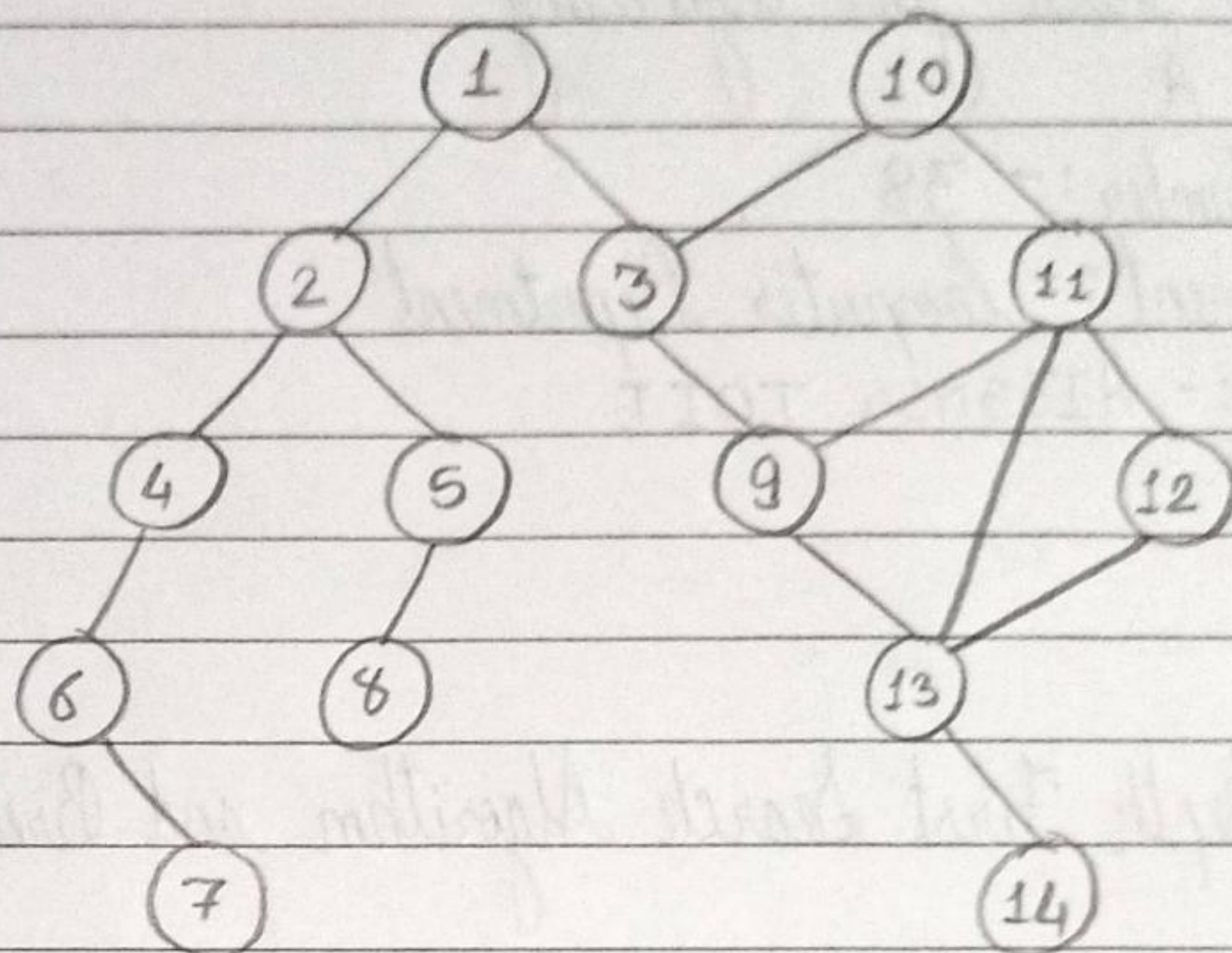
1. Depth First Search-

Depth first search is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root node (selecting any arbitrary node as root node in case of graph) and explores as far as

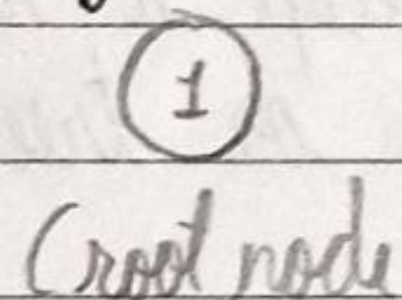


possible along each branch backtracking.

Example:-

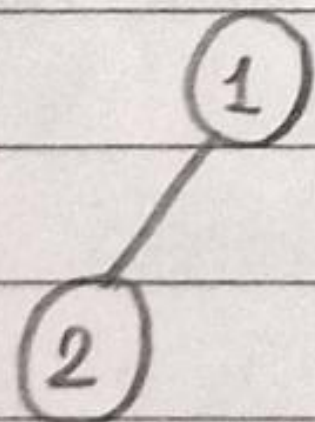


→ 1) Take any arbitrary node as root node (suppose ① as root node)



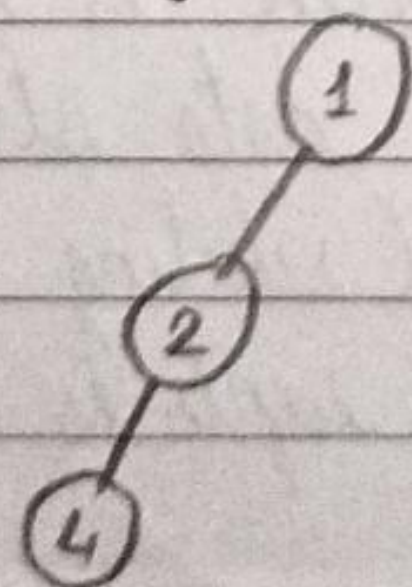
DFS → 1,

2) Explore any one node to 1.



DFS → 1, 2,

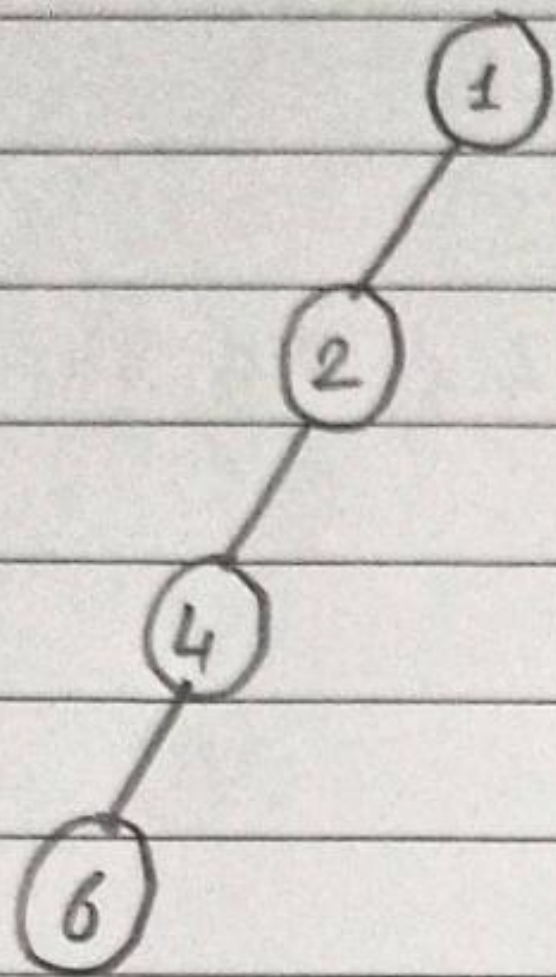
3) Similarly explore any one node connected to 2.



DFS → 1, 2, 4,

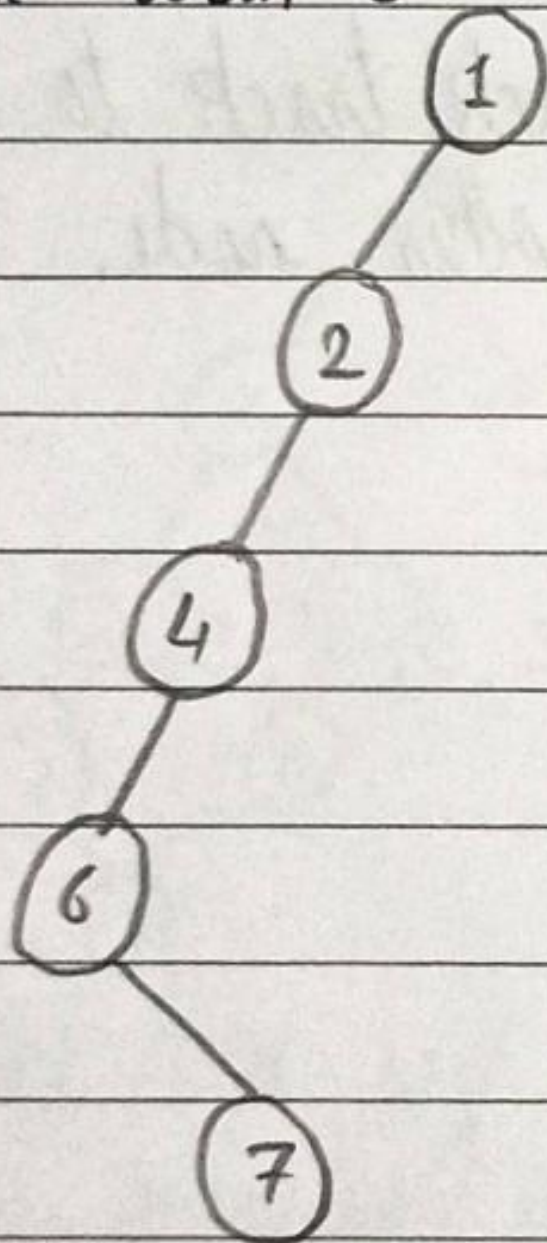


4) Repeat same with 4,



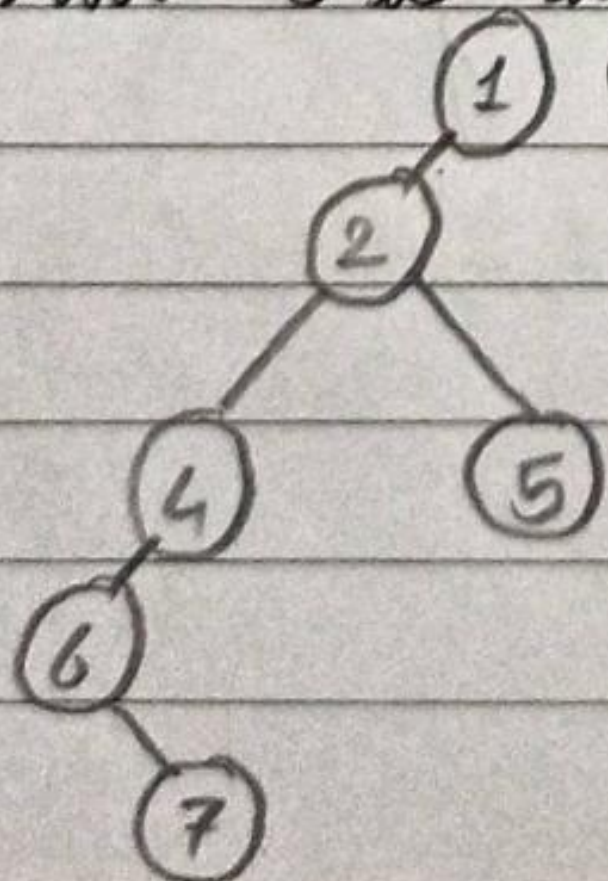
DFS = 1, 2, 4, 6,

5) Same with 6 -



DFS = 1, 2, 4, 6, 7,

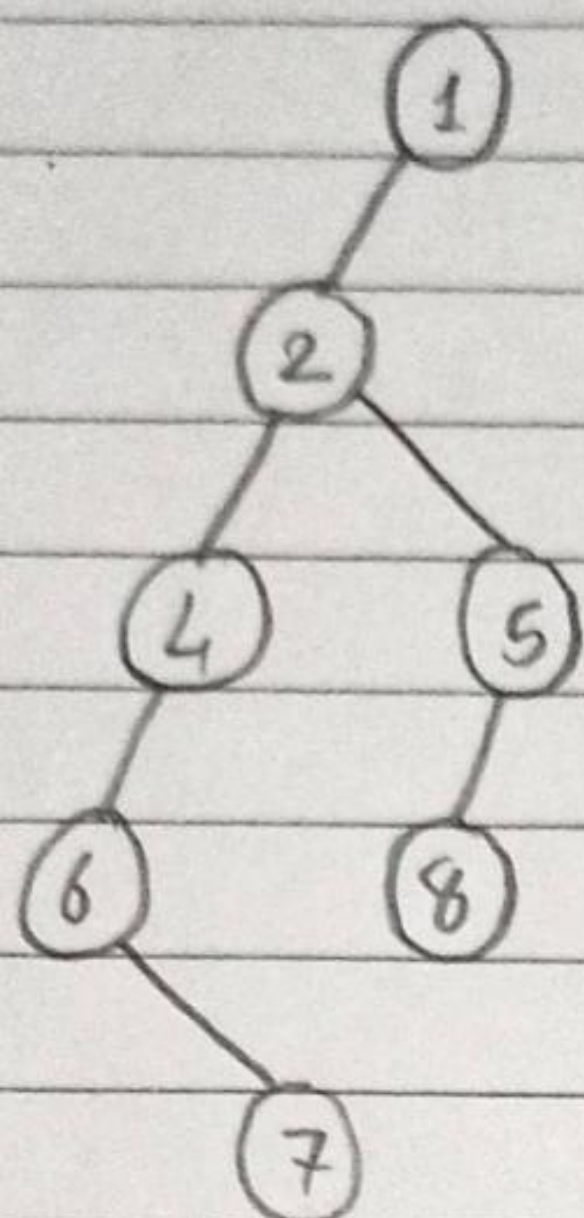
6) As 7 don't have any connection, back track to 6. 6 also don't have any other connection, back track to 4. Same with 4, back track to 2. 2 has 5 as another connection...



DFS = 1, 2, 4, 6, 7, 5.

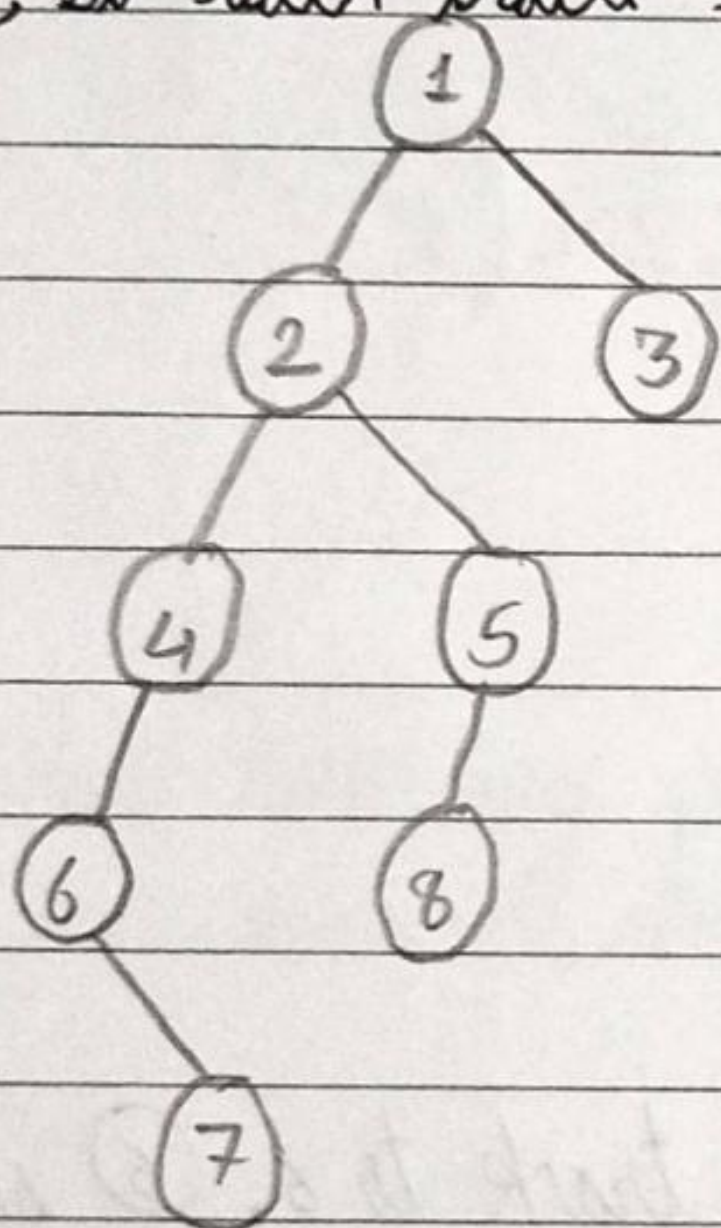


7) 5 have 8 as connection



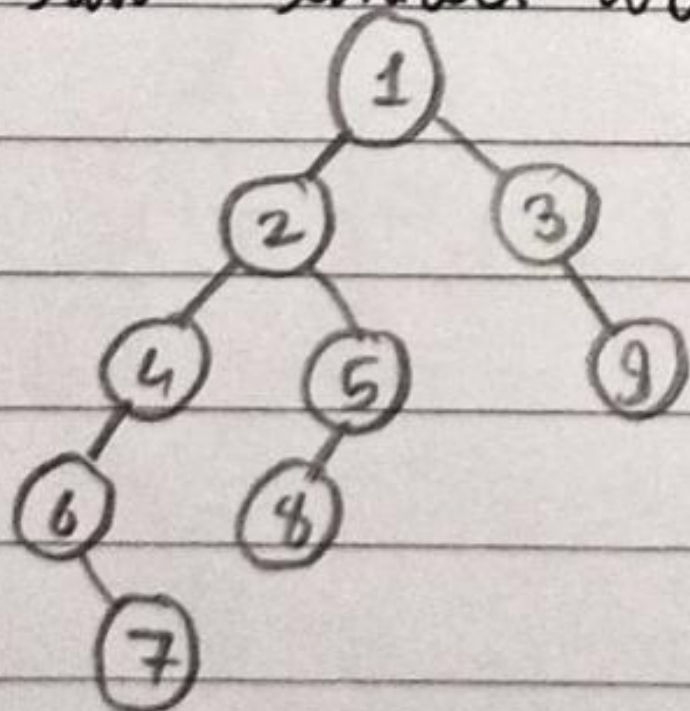
① FS = 1, 2, 4, 6, 7, 5, 8,

8) 8 dont have any other connection, so back track to 5, same with 5 and 2, so back track to 1, 1 have 3 as other node.



① FS = 1, 2, 4, 6, 7, 5, 8, 3,

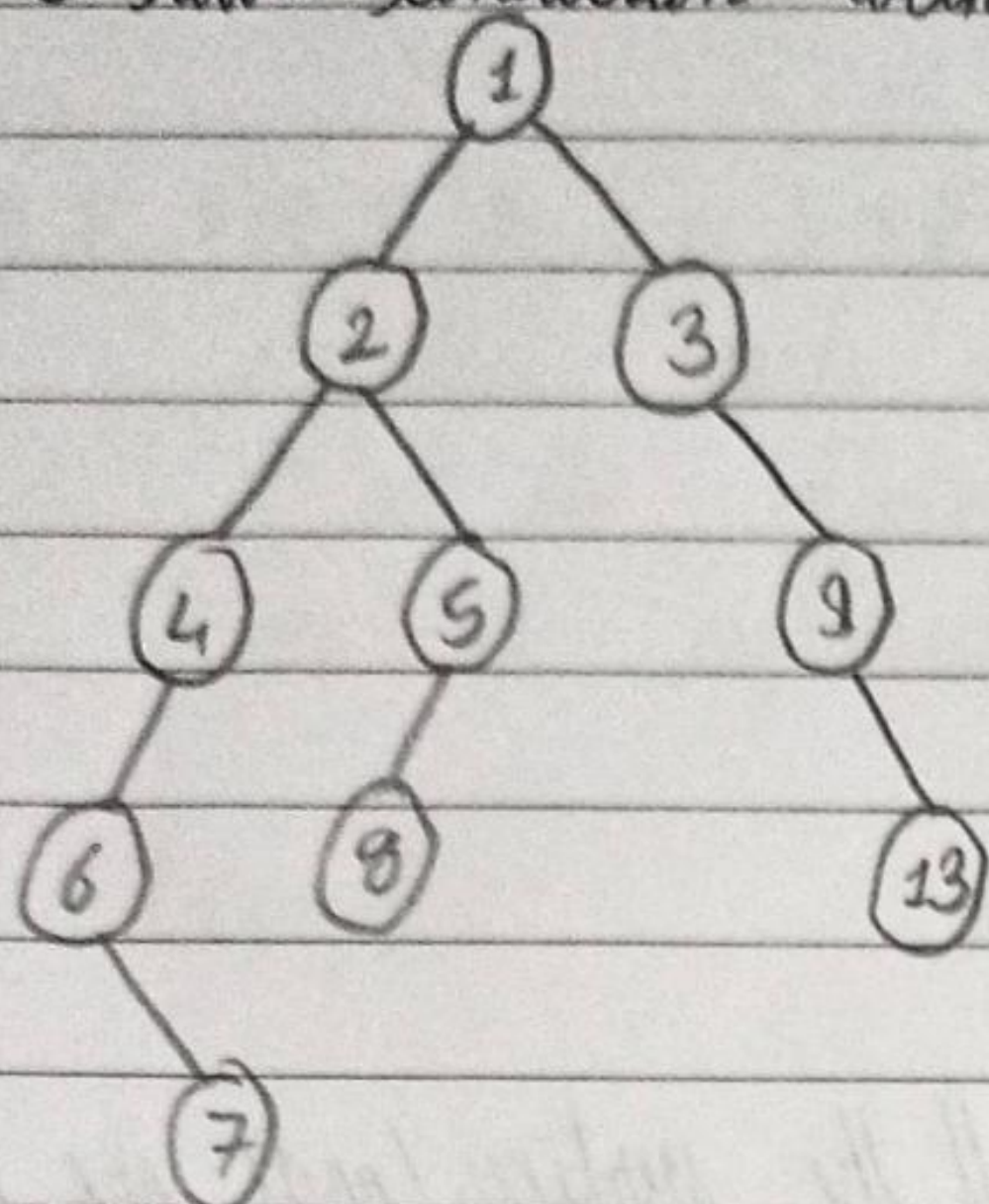
9) 3 have connect with 9 and 10, explore any one -



① FS = 1, 2, 4, 6, 7, 5, 8, 3, 9

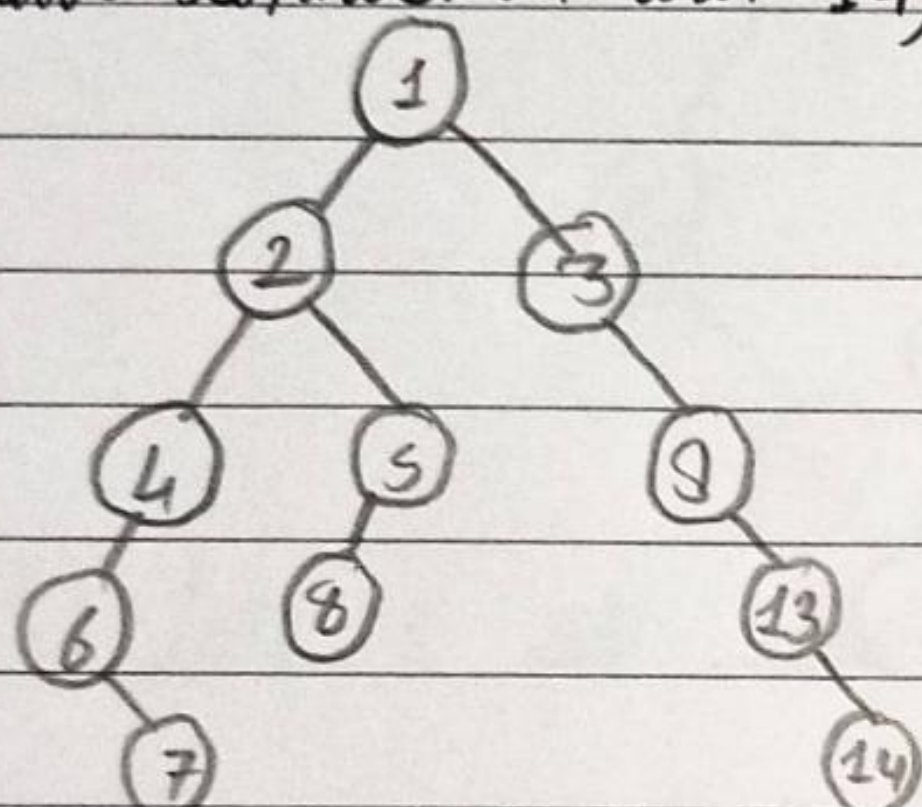


10) 9 have connection with 13 and 11, explore any one -



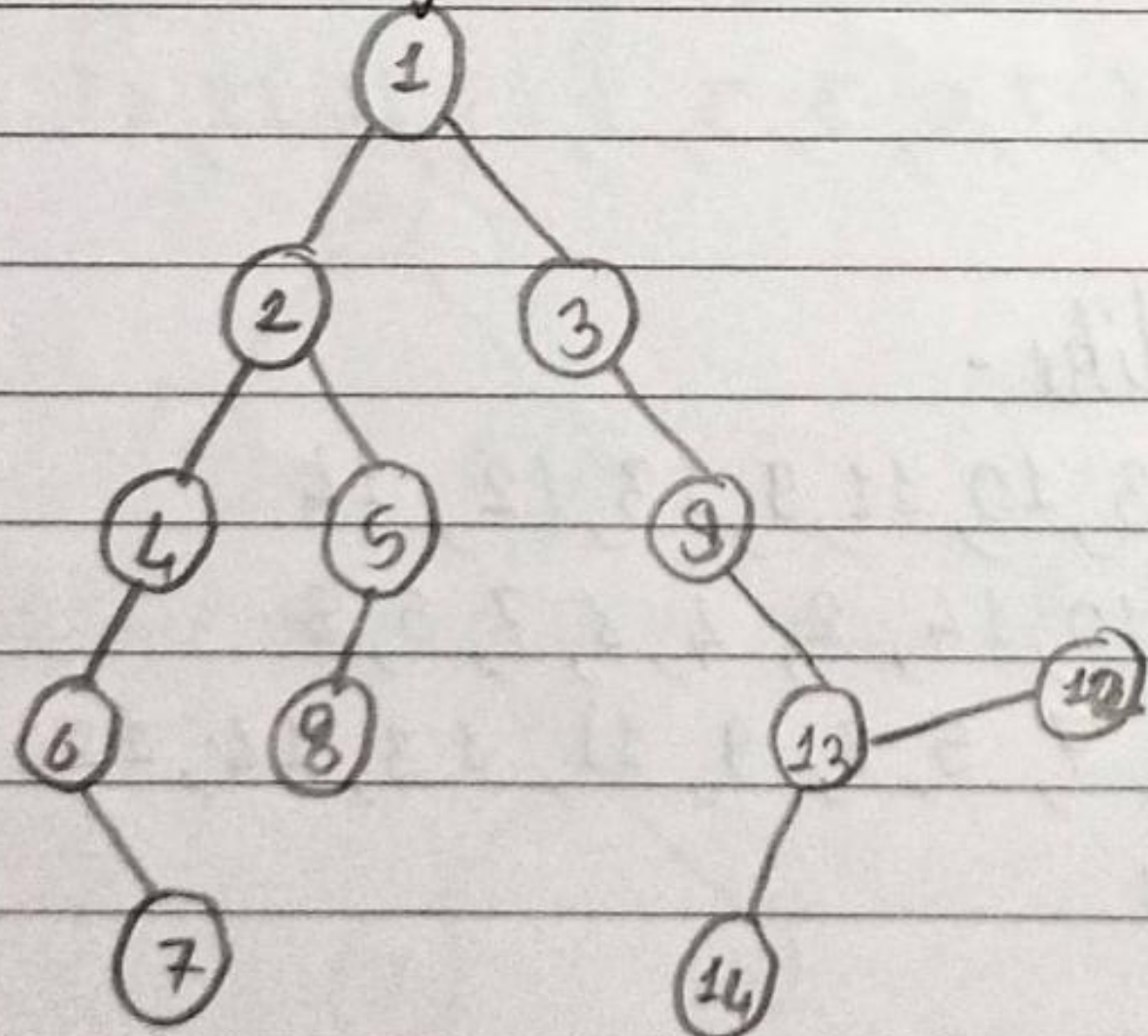
DFS = 1, 2, 4, 6, 7, 5, 8, 3, 9, 13,

11) 13 have connection with 14, 12, 11, explore any one.



DFS = 1, 2, 4, 6, 7, 5, 8, 3, 9, 13, 14,

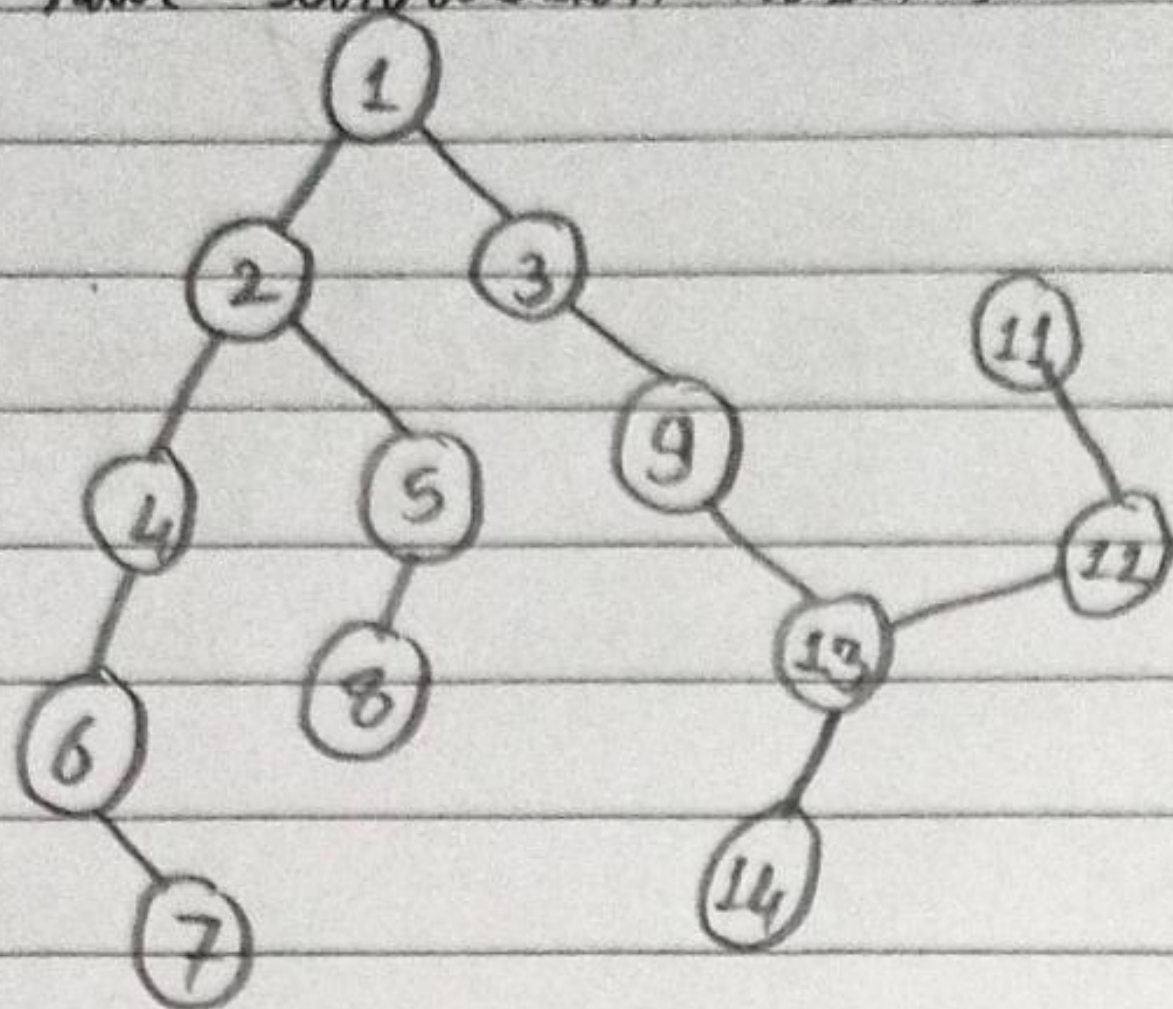
12) 14 dont have any further connection so back track to 13, and explore any one remaining node.



DFS = 1, 2, 4, 6, 7, 5, 8, 3, 9, 13, 14, 12

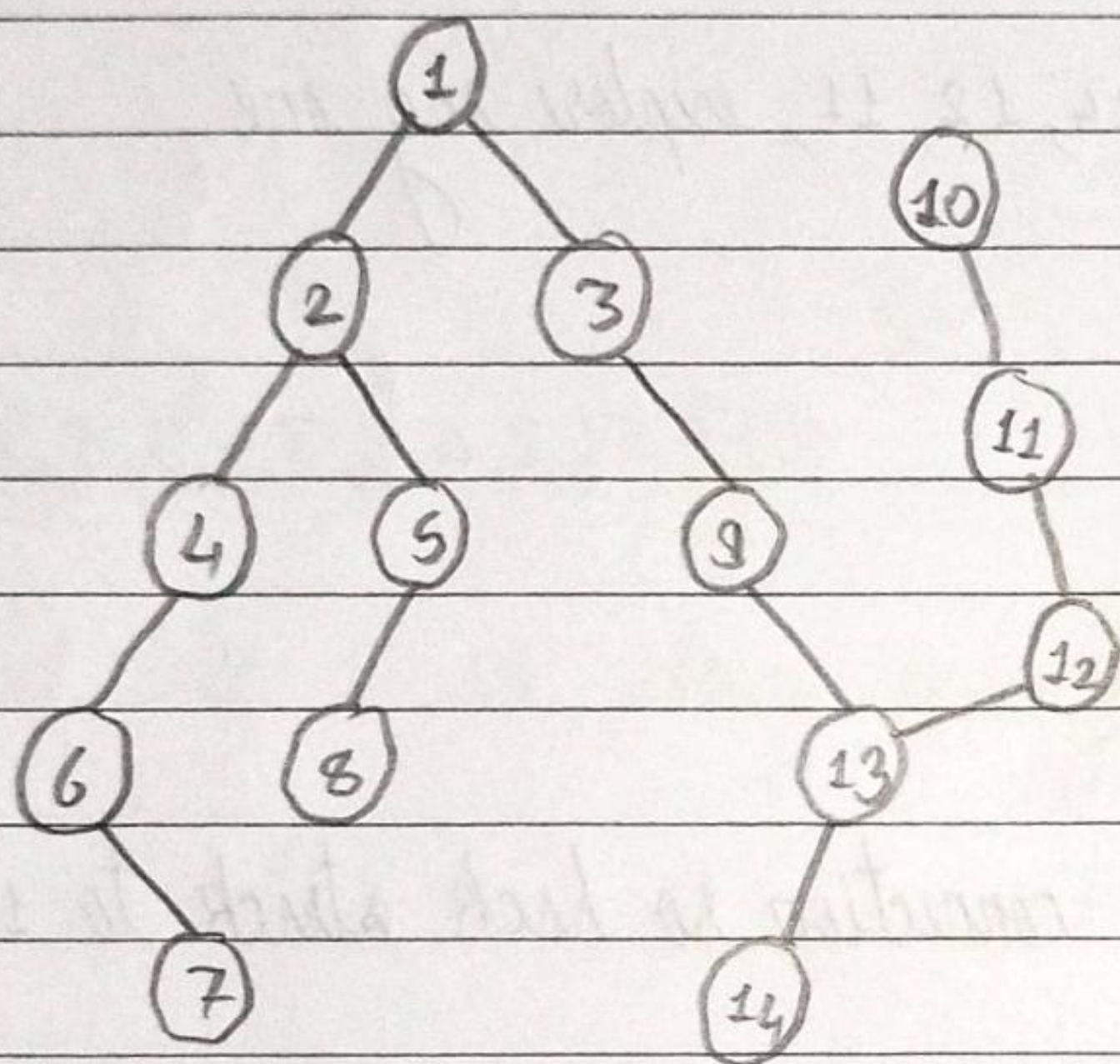


13) 12 have connection with 11.



DFS = 1, 2, 4, 6, 7, 5, 8, 3, 9, 13, 14, 12, 11,

14) 11 have connect with 10 and all the vertices/node are cover.



$\therefore$  DFS = 1, 2, 4, 6, 7, 5, 8, 3, 9, 13, 14, 12, 11, 10

Other DFS can also possible like -

DFS = 1, 2, 5, 8, 4, 6, 7, 3, 10, 11, 9, 13, 12, 14.

or DFS = 1, 3, 9, 13, 12, 11, 10, 14, 2, 4, 6, 7, 5, 8.

or DFS = 10, 3, 1, 2, 4, 6, 7, 5, 8, 9, 11, 13, 14, 12.

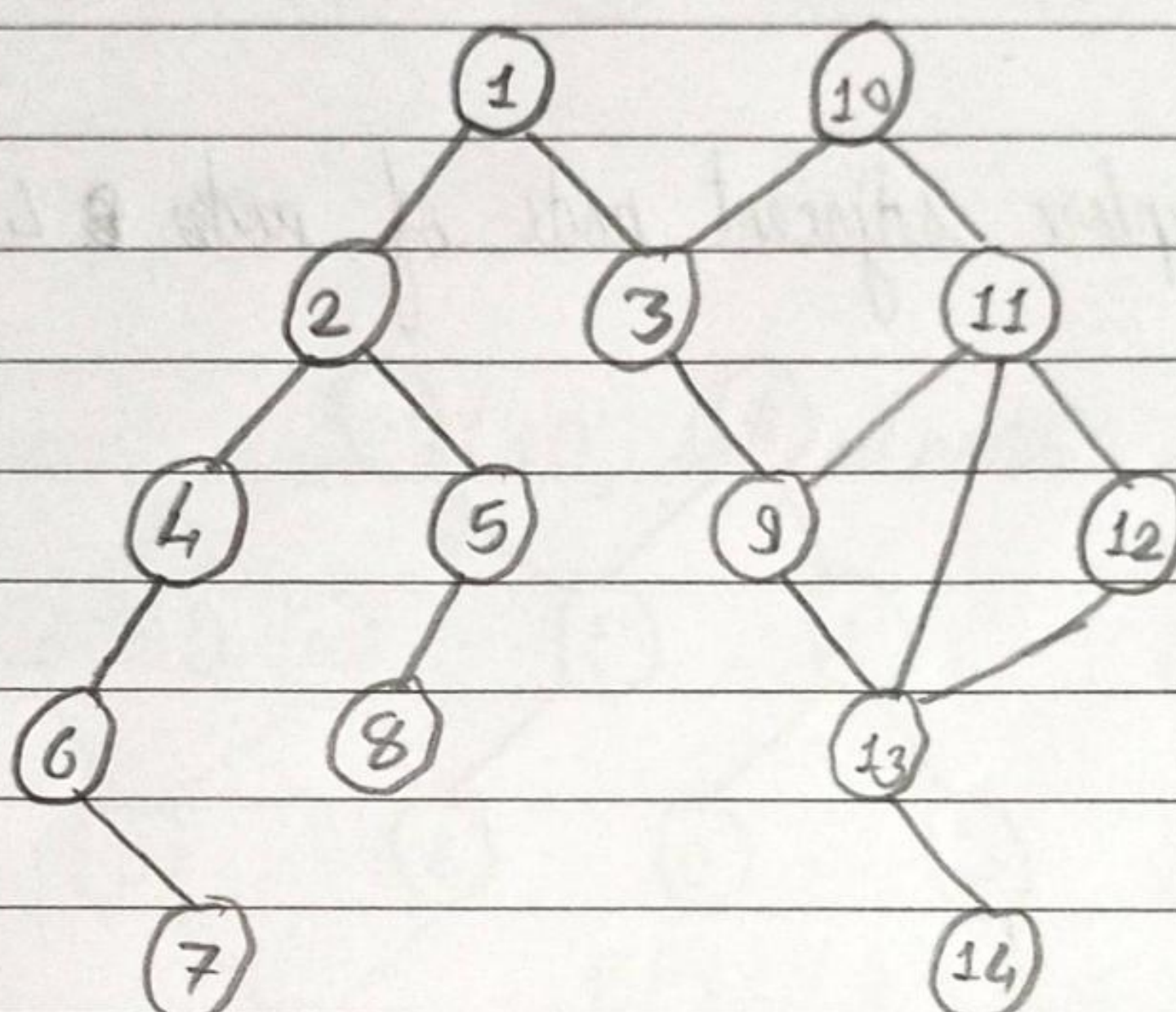
and many more.



## 2. Breadth First Search:-

Breadth first search is an algorithm for traversing or searching tree or graph data structure. The algorithm starts at any one node and exploration is done to all the adjacent nodes connected to starting node then only next node is explored.

Example:-

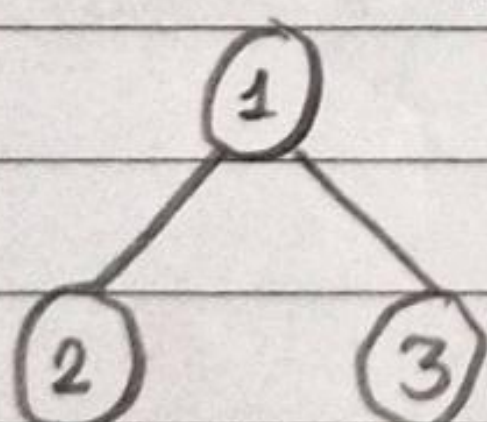


→ 1. Let take 1 as starting vertex/node.

(1)

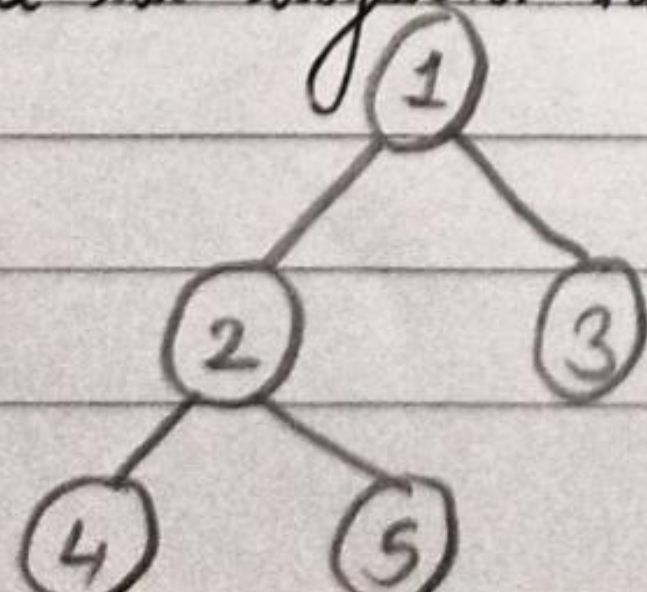
BFS = 1,

2. Explore all adjacent nodes of 1



BFS = 1, 2, 3,

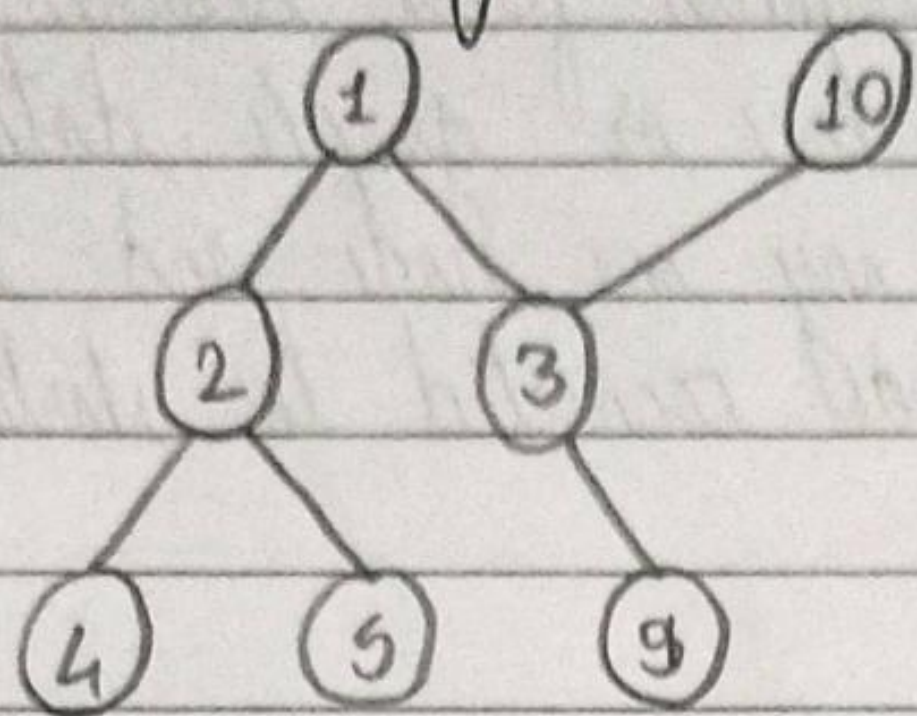
3. Explore all adjacent nodes of 2 or 3,



BFS = 1, 2, 3, 4, 5,

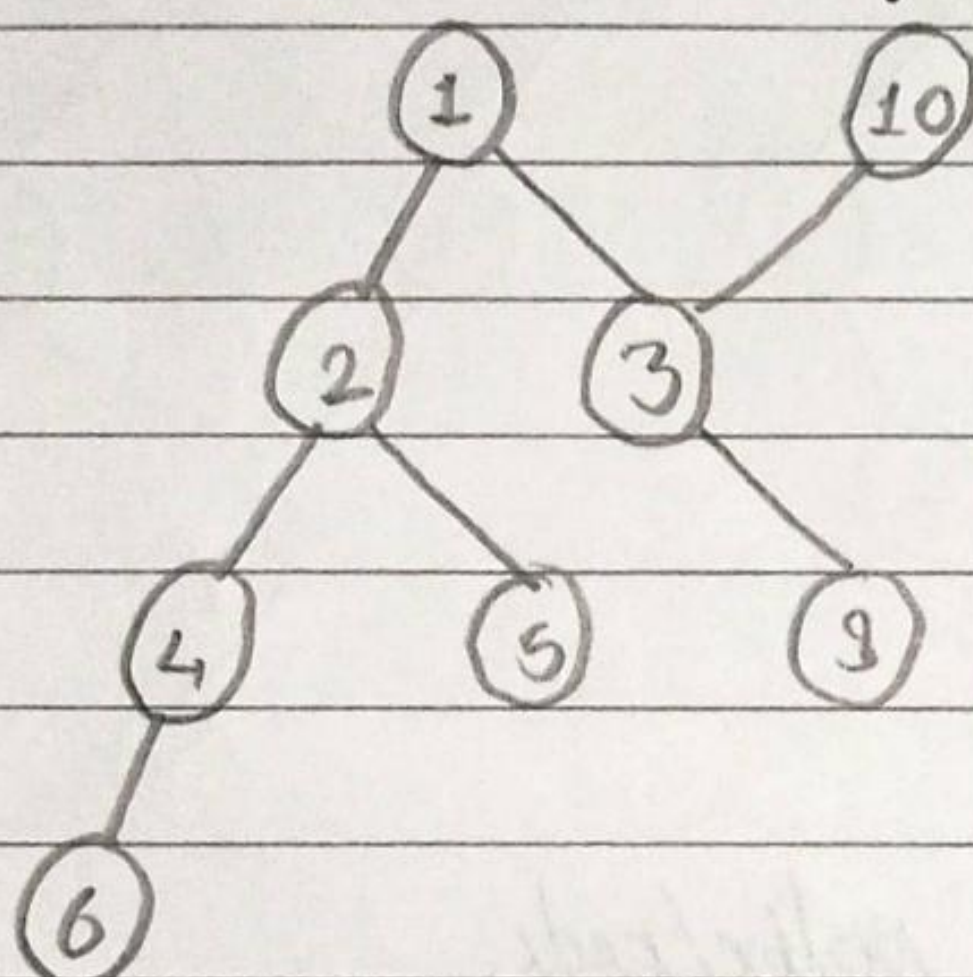


4. Now explore remaining adjacent nodes of 1 i.e. 3.



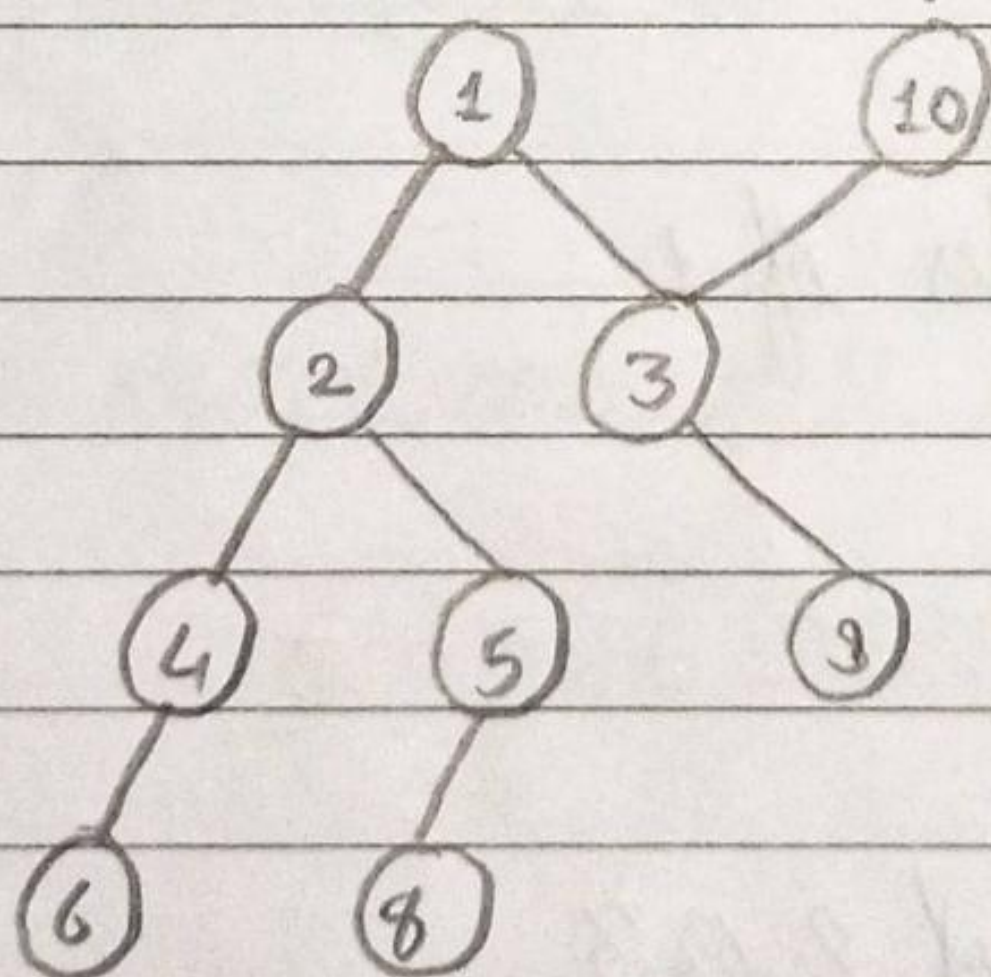
BFS = 1, 2, 3, 4, 5, 9, 10

5. Now explore adjacent node of node 4 and 5



BFS = 1, 2, 3, 4, 5, 9, 10, 6

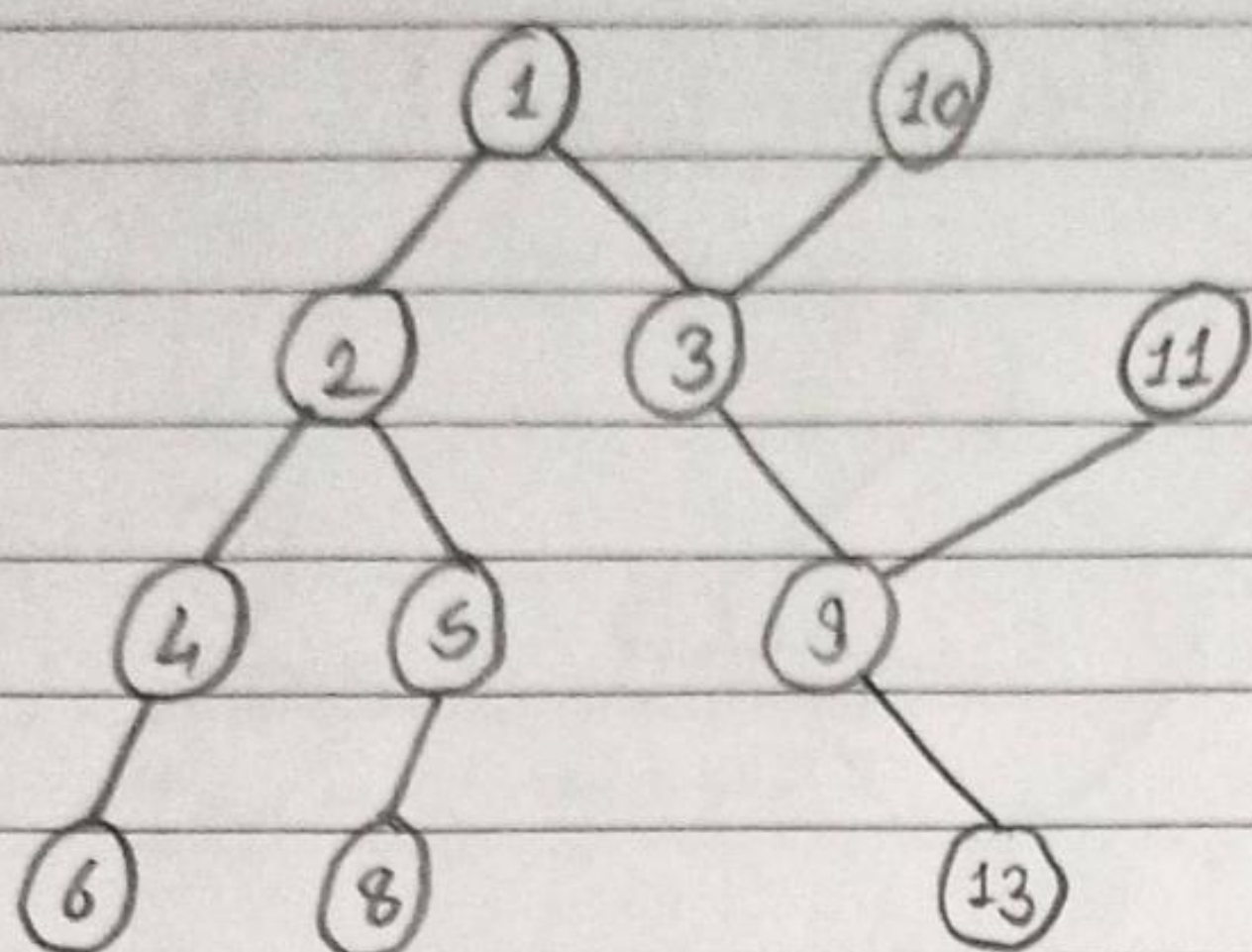
6. Now explore adjacent nodes of 5



BFS = 1, 2, 3, 4, 5, 9, 10, 6, 8,

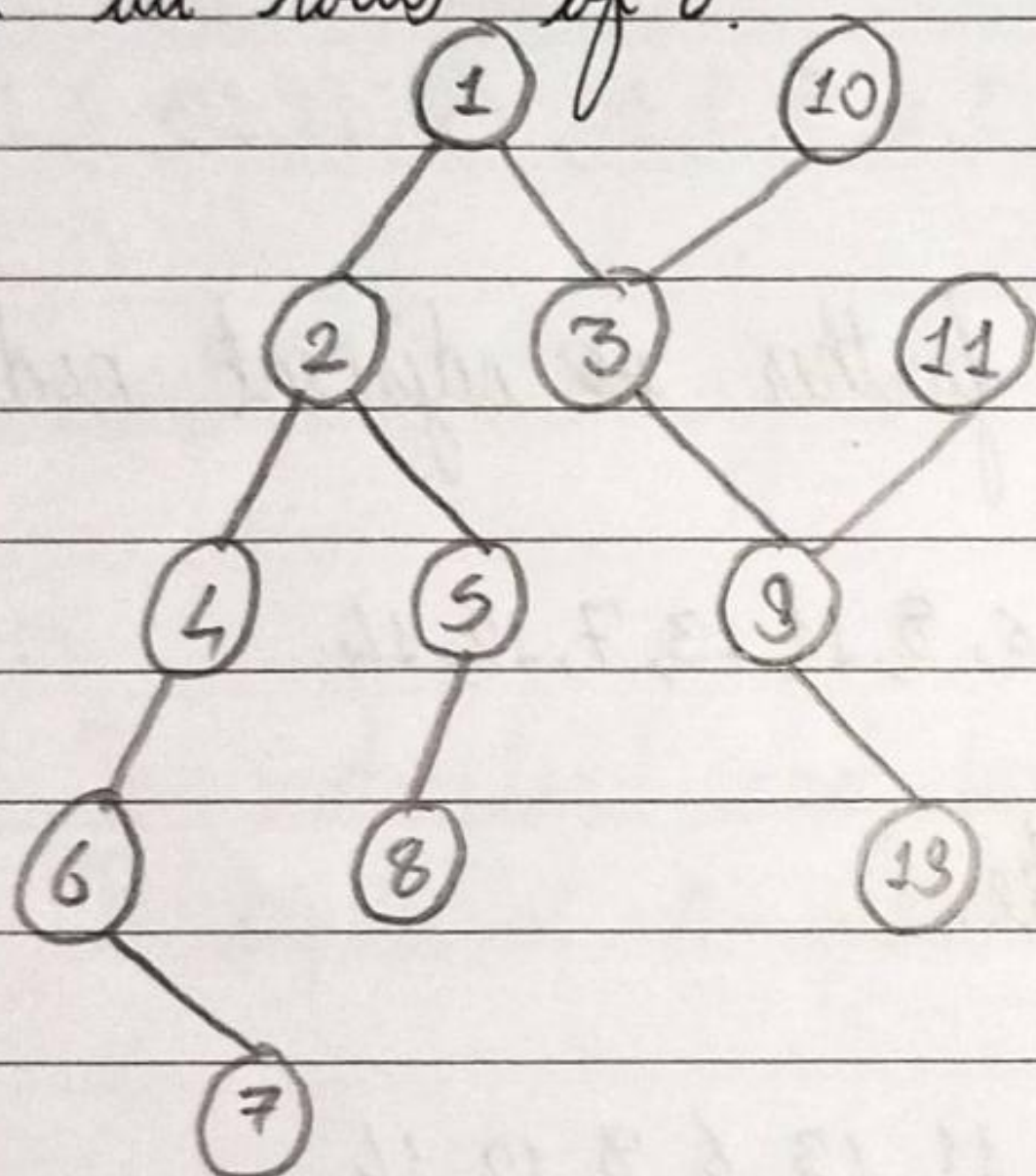


7. Now explore adjacent nodes of 9.



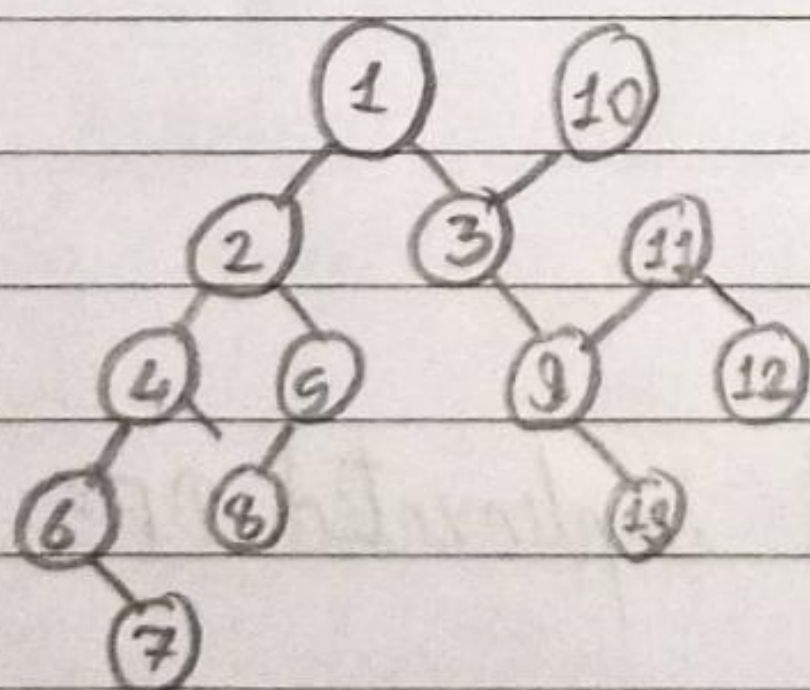
BFS = 1, 2, 3, 4, 5, 9, 10, 6, 8, 11, 13,

8. Now explore adjacent nodes of 10, but all nodes are explored so now explore all nodes of 6.



BFS = 1, 2, 3, 4, 5, 9, 10, 6, 8, 11, 13, 7.

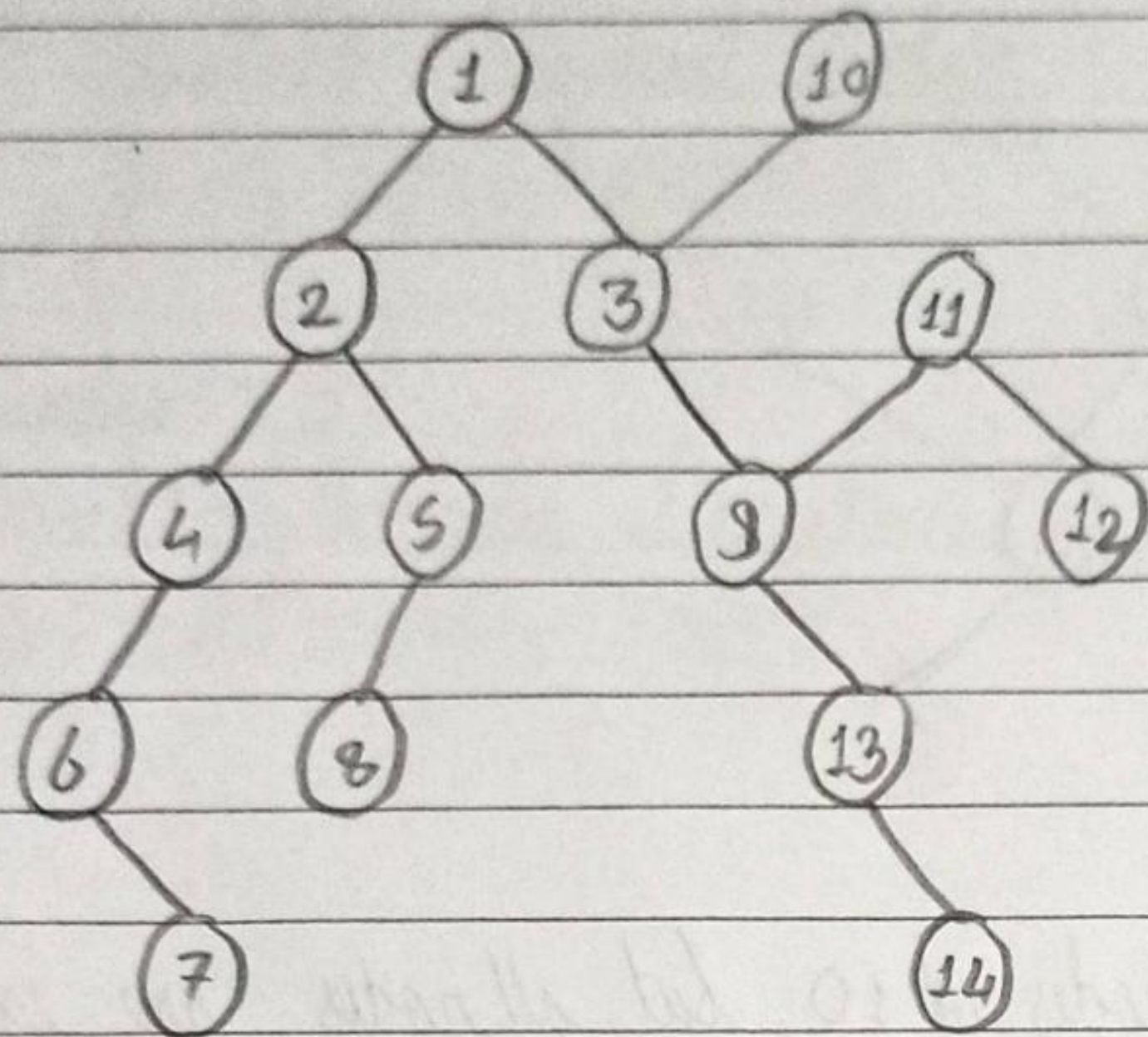
9. Now explore adjacent nodes of 8, but all nodes are explored now explore <sup>all</sup> nodes of 11.



BFS = 1, 2, 3, 4, 5, 9, 10, 6, 8, 11, 13, 7, 12



10. Now explore adjacent nodes of 13.



BFS = 1, 2, 3, 4, 5, 9, 10, 6, 8, 11, 13, 7, 12, 14.

11. 7, 12 and 14 don't have any further adjacent nodes. So BFS is

BFS = 1, 2, 3, 4, 5, 9, 10, 6, 8, 11, 13, 7, 12, 14.

Other BFS can also possible like.

BFS = 1, 3, 10, 9, 4, 5, 11, 13, 6, 8, 12, 14.

BFS = 1, 3, 2, 10, 9, 5, 4, 11, 13, 8, 6, 12, 14.

and many more.

Conclusion :-

We have studied and implemented DFS and BFS on undirected graph.