P-1

Implement depth first search algorithm & Breadth first search algorithm, Use an undirected graph & develop a recursive algorithm for searching all the Vertices of a graph or tree data structure.

Difference between Tree & Graph:-

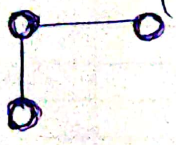
⇒ Every graph is a tree. But every tree may or may not be a graph.

Because in a graph, cycle is allowed but it is not allowed in a tree.

Graph Traversal:-

The process of visiting & exploring a graph for processing is called a graph traversal.

In a graph there are vertex & edges.
(V, e) (vertices)



Visiting → reaching to vertex

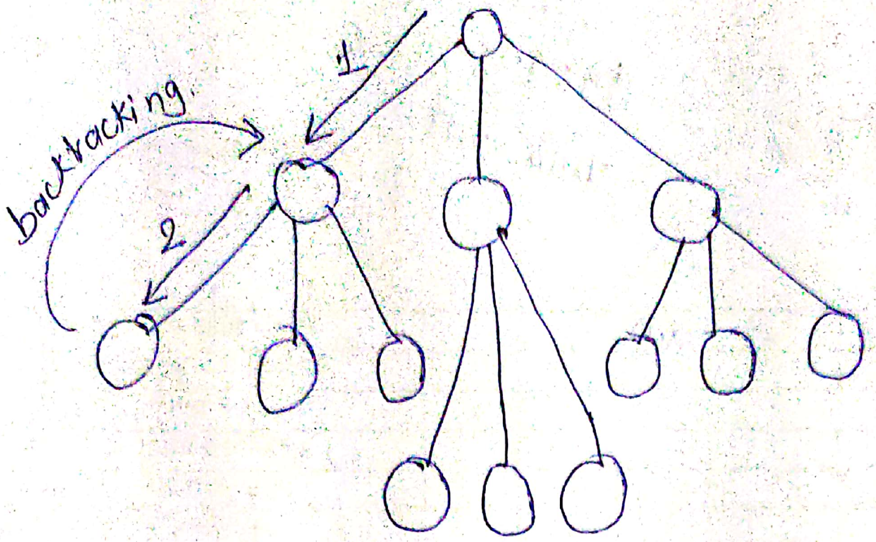
Exploring → checking the neighbours of that vertex.

Depth First Search & Breadth First Search

- In DFS, we goe as deep as Possible down one path before backtracking.
  When deadend found, we backtrack.

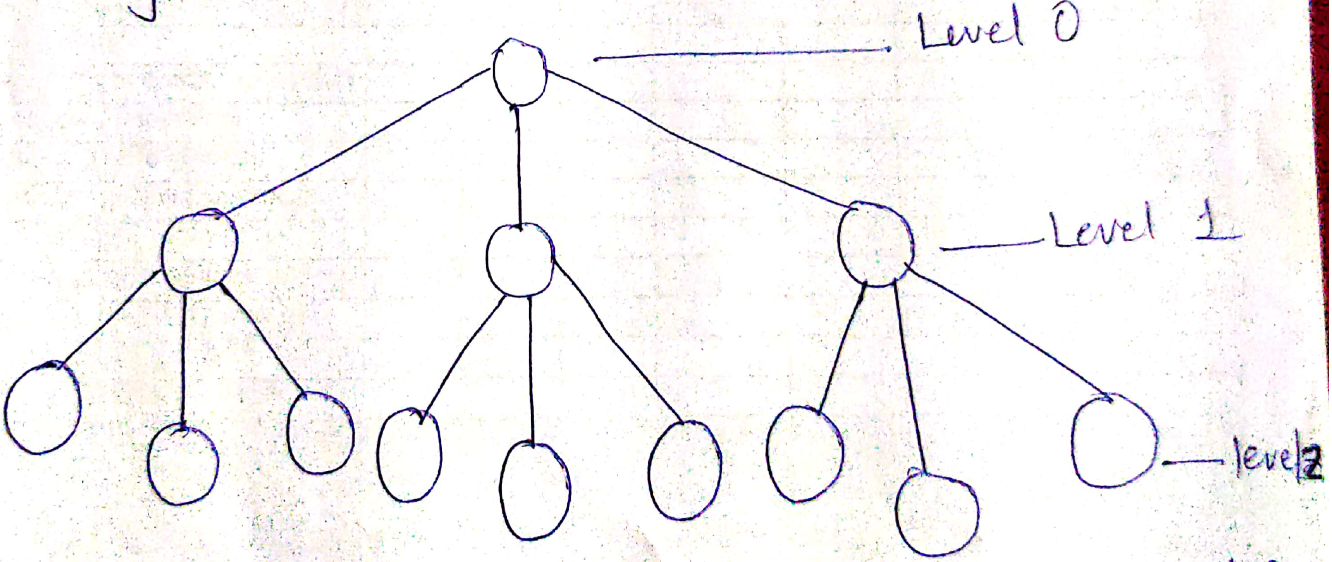
- In BFS, We explore all nodes at the current level before moving to the next level.

DFS Diagram:-



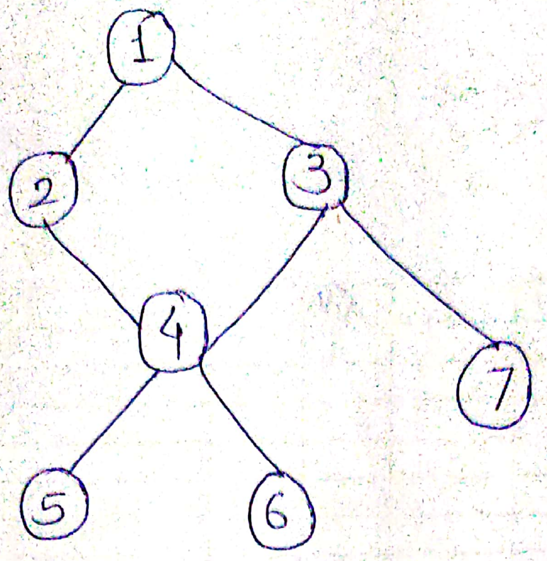
In DFS we choose one direction, till we found dead end. Once we get deadend then we backtrack.

BFS Diagram



In BFS we go level wise. First we cover level 0 then we goto the level 1 & finally level 2.

BFS :- Example



- We use queue data structure.
- We can start from any node.
- ~~Add~~ choose any node to start. Add it in a queue.

~~1~~	

- 1
- Remove 1 from queue & explore it's neighbours. add them in a queue.

~~1~~	~~2~~	3

2
- explore neighbours. Add them in a queue.

~~1~~	~~2~~	~~3~~	4

- 3 explore neighbours. - 7 & 4. 4 is already present so add 7.

~~1~~	~~2~~	~~3~~	4	7

- 4 - explore neighbours & add them in a queue.

~~1~~	~~2~~	~~3~~	~~4~~	7	5	6

- 7. No node is connected.

~~1~~	~~2~~	~~3~~	~~4~~	~~7~~	5	6

- 5. No node is connected.

~~1~~	~~2~~	~~3~~	4	~~7~~	~~5~~	6

- 6 No node is connected.

So BFS order is $\boxed{1234756}$