Traversal Technique:-

1. BFS(Level Order Traversal)
2. DFS(Depth First Search)

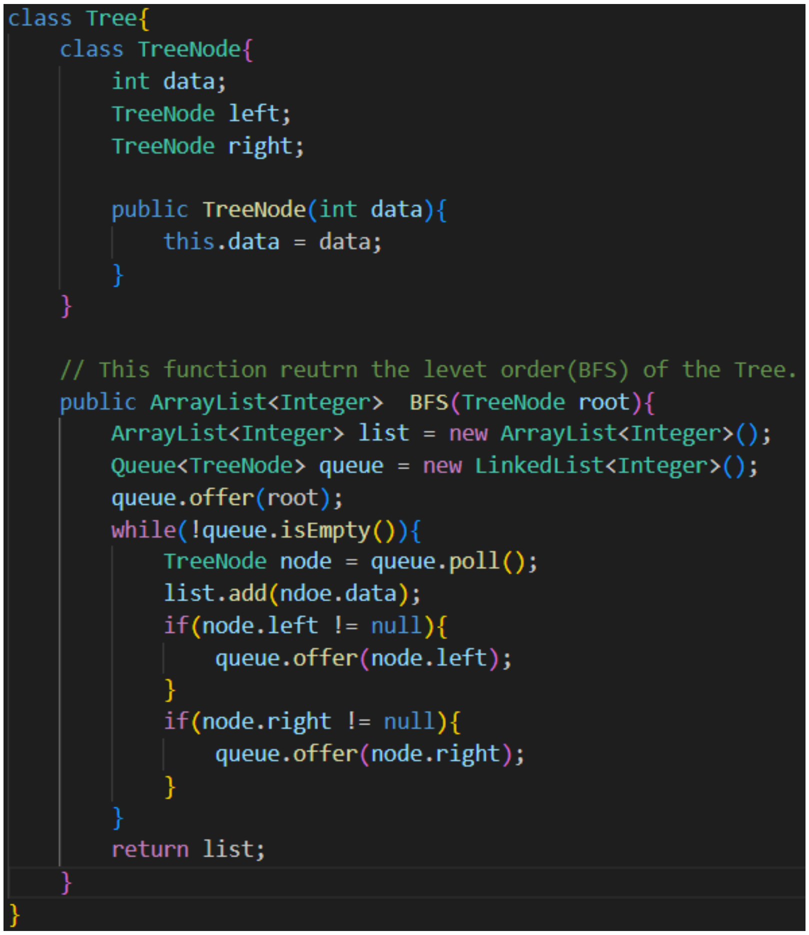
Level Order Traversal (Breadth First Search or BFS) of Binary Tree: The **Level Order Traversal** technique is defined as a method to traverse a Tree such that all nodes present in the same level are traversed completely before traversing the next level.

How does Level Order Traversal work?

The main idea of level order traversal is to traverse all the nodes of a lower level before moving to any of the nodes of a higher level. This can be done in any of the following ways:

- the naive one (finding the height of the tree traversing each level and printing the nodes of that level)
- efficiently using a queue.

Lever order Traversal for Binary and A-array Tree using queue

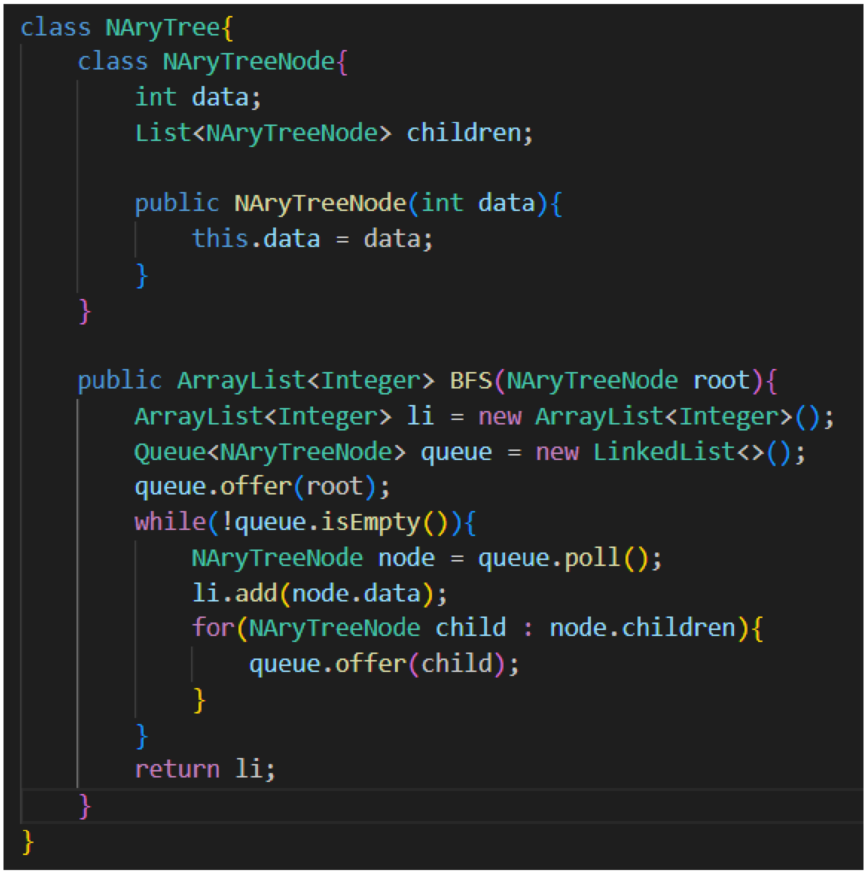


```
class Tree{
    class TreeNode{
        int data;
        TreeNode left;
        TreeNode right;

        public TreeNode(int data){
            this.data = data;
        }
    }

    // This function reutrn the levet order(BFS) of the Tree.
    public ArrayList<Integer>  BFS(TreeNode root){
        ArrayList<Integer> list = new ArrayList<Integer>();
        Queue<TreeNode> queue = new LinkedList<Integer>();
        queue.offer(root);
        while(!queue.isEmpty()){
            TreeNode node = queue.poll();
            list.add(ndoe.data);
            if(node.left != null){
                queue.offer(node.left);
            }
            if(node.right != null){
                queue.offer(node.right);
            }
        }
        return list;
    }
}
```

This Level order(BFS) for the Binary Tree



```java
class NAryTree{
    class NAryTreeNode{
        int data;
        List<NAryTreeNode> children;

        public NAryTreeNode(int data){
            this.data = data;
        }
    }

    public ArrayList<Integer> BFS(NAryTreeNode root){
        ArrayList<Integer> li = new ArrayList<Integer>();
        Queue<NAryTreeNode> queue = new LinkedList<>();
        queue.offer(root);
        while(!queue.isEmpty()){
            NAryTreeNode node = queue.poll();
            li.add(node.data);
            for(NAryTreeNode child : node.children){
                queue.offer(child);
            }
        }
        return li;
    }
}
```

N-Array Tree Level Order Traversal (BFS)

Both the algorithm is write using the Queue

Easy Level Problem - for a tree (Breath First Search)

1. Level Order Traversal (On Binary Tree)
2. 429. N-ary Tree Level Order Traversal (On N-ary Tree)

Depth First Search

DFS is a graph traversal algorithm that explores as far as possible along each branch before backtracking. In the context of trees, it's a systematic way to visit all nodes, starting from the root and going as deep as possible before moving to the next sibling.

Steps:

1. Start at the **root**: Begin with the root node.
2. **Visit node**: Process the current node (e.g., print its value).
3. Recursively **visit children**: For each child of the current node, recursively call DFS on that child.
4. **Backtrack**: Once all children have been visited, return to the parent node.

Basic Problem for Tree (depth first search):

1. Preorder Traversal (Binary Tree)
2. Inorder Traveral (Binary Tree)
3. Postorder Traversal (Binary Tree)