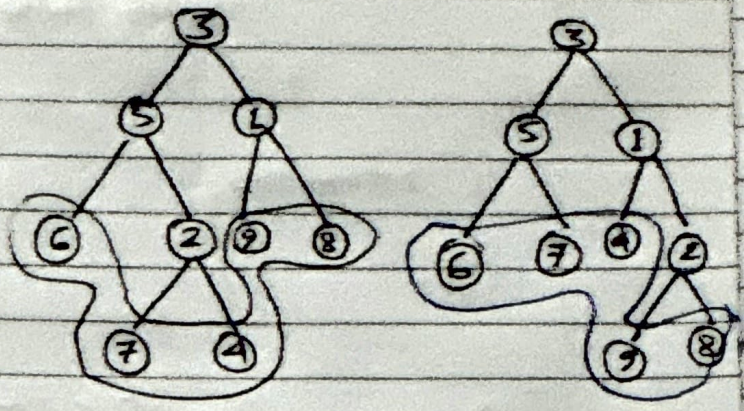
LEETCODE | 872 |

DATE

Leaf - Similar Trees



Approach DFS on BT

1) Base condition

```
if(root1 == null &&
        root2 == null){
    return true
}
```

2) Create a recursive function as below

```
void dFSTree(TreeNode node, Queue<Integer> queue){
    if(node == null)
        return;

    // leaf condition
    if(node.left == null && node.right == null){
        queue.offer(node.val);
    }

    dFSTree(node.left, queue);      // left node
    dFSTree(node.right, queue);     // right node

}
```

3) Populate queues by calling above recursion call and say Queue q_1 & q_2 are populated

4) Pop each queue & compare

A) `if(q1.size() != q2.size()) → return false`

B)
```
while(!q1.isEmpty() && !q2.isEmpty()){
    int node1 = q1.poll();
    int node2 = q2.poll();
```

classmate

PAGE

DATE

```
if (node1 != node2){
        return false;
    }
return true;
}
```

Queue q1 → [6, 7, 4, 9, 8]

Queue q2 → [6, 7, 4, 9, 8]

A) Return false if q1.size() != q2.size();

B) poll() each Queue
& return false if they don't match.