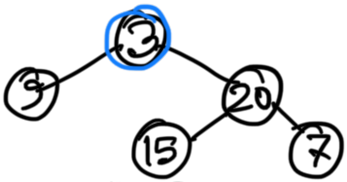
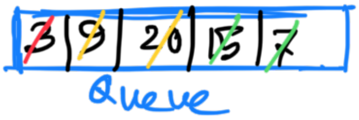
Binary Tree Level Order Traversal

Leetcode - 102



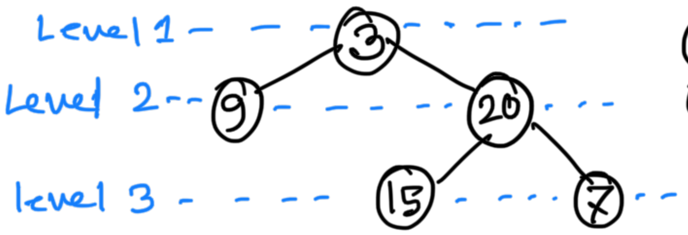


output: [[3], [9,20], [15,7]]

[3],
[9,20],
[15, 7],

```
public List<List<Integer>> levelOrder(TreeNode root){
    if(root == null) return new ArrayList<>();
    List<List<Int>> res = new ArrayList<>();
    Deque<TreeNode> q = new ArrayDeque<>();
    q.add(root);
    while (! q.isEmpty()){
        List<Integer> subList = new ArrayList<>();
        int qLen = q.size();
          for (int i=0; i<qLen ; i++){
              TreeNode node = q.poll();
              if(node != null){
                   sublist.add (node);
                   if(node.left != null)
                       q.add( node.left);
                   if(node.right != nuu)
                       q.add (node.right);
              }
          }
        result.add (subList);
    }
 return result;
```

Simulation



① push left, push right

② poll untill the q is empty

* To do. this: we have to loop through the q by level order.

push	pop	Queue	qLen	Sublist	Result
3		3	1	empty	empty
9, 20	3	9, 20	1	[3]	empty
				→	[[3]]
	9	20	2	[9]	[[3]]
15, 7	20	15, 7	2	[9, 20]	[[3]]
				→	[[3], [9,20]]
	15	7	2	[15]	[[3], [9, 20]]
	7	empty	2	[15, 7]	[[3], [9,20]]
				→	[[3], [9, 20], [15, 7]]