

```

1  /* Given a binary tree, return the level order traversal of its nodes' values. (ie,
   from left to right, level by level).
2
3  For example:
4  Given binary tree [3,9,20,null,null,15,7],
5      3
6     /\
7    9  20
8   /\  /\
9  15  7
10 return its level order traversal as:
11 [
12  [3],
13  [9,20],
14  [15,7]
15 ] */
16
17
18 import java.util.LinkedList;
19 import java.util.List;
20 import java.util.Queue;
21
22 class TreeNode {
23     int val;
24     TreeNode left;
25     TreeNode right;
26
27     TreeNode(int x) {
28         val = x;
29     }
30 }
31
32
33 class Solution {
34     public List<List<Integer>> levelOrder(TreeNode root) {
35
36         List<List<Integer>> result = new LinkedList<>();
37
38         if (root == null) {
39             return result;
40         }
41
42         Queue<TreeNode> queue = new LinkedList<>();
43         queue.offer(root);
44
45         while (!queue.isEmpty()) {
46
47             List<Integer> temp = new LinkedList<>();
48
49             int size = queue.size();
50             for (int i = 0; i < size; i++) {
51                 TreeNode tempNode = queue.poll();
52                 temp.add(tempNode.val);
53                 if (tempNode.left != null) {
54                     queue.offer(tempNode.left);
55                 }
56                 if (tempNode.right != null) {
57                     queue.offer(tempNode.right);
58                 }
59             }
60
61             result.add(temp);
62         }
63
64         return result;
65     }
66 }

```