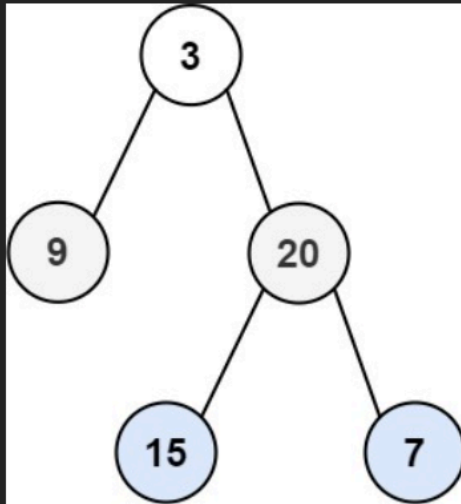


## 102. Binary Tree Level Order Traversal

### Problem Statement

Given the `root` of a binary tree, return *the level order traversal of its nodes' values*. (i.e., from left to right, level by level).

#### Example 1:



**Input:** `root = [3,9,20,null,null,15,7]`

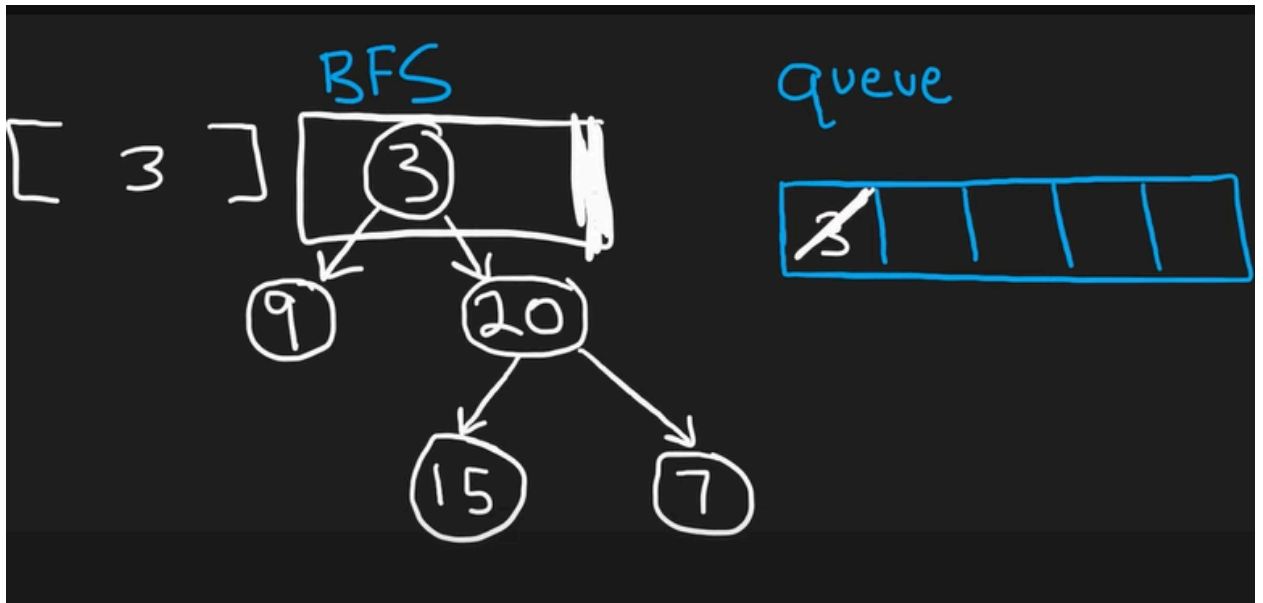
**Output:** `[[3], [9,20], [15,7]]`

1.

### Approach

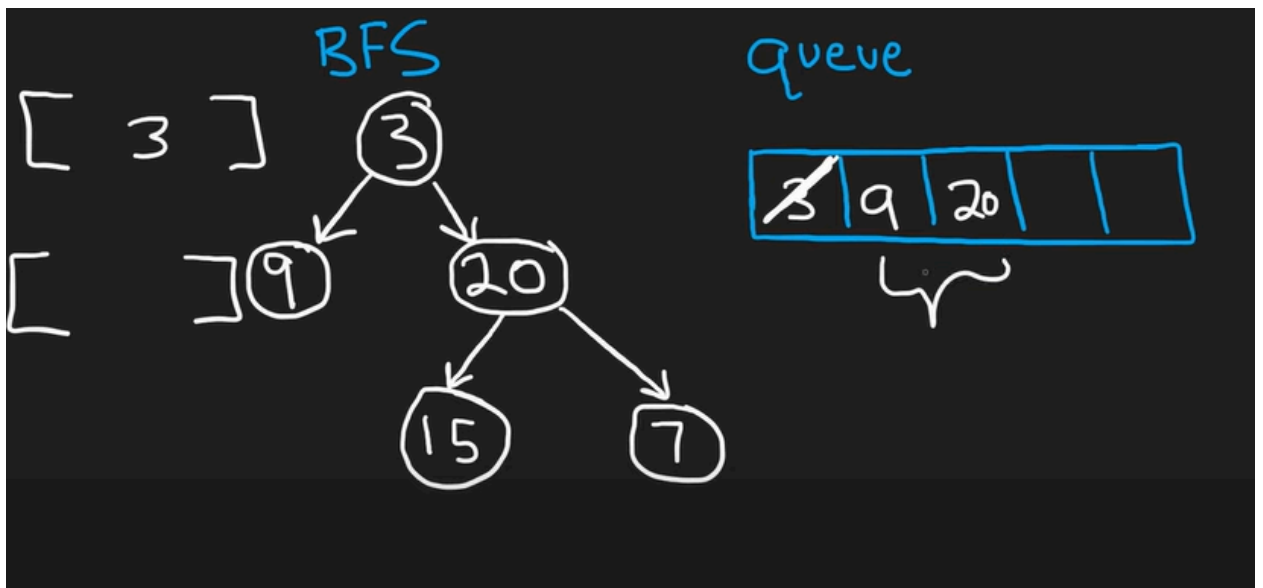
1. So here we'll be applying the breadth first search algorithm using the queue data structure

2. First we'll insert the root in the queue if its present

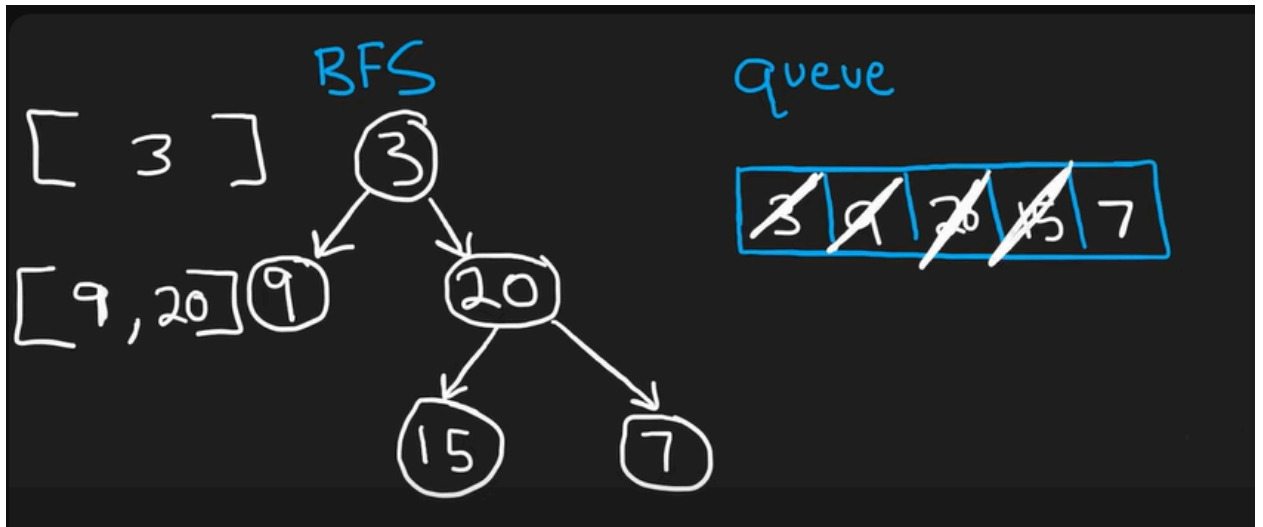


And later pop (add it to a list) and add its children in the queue

3. Now we first pop 9 and add its children to the queue and later pop 20 from the queue and add its children



4. And similarly we do for 15 and 7 too



Python code:

```
from collections import deque
class Solution:
    def levelOrder(self, root: Optional[TreeNode]) -> List[List[int]]:
        res = []
        q = deque()
        q.append(root)

        while q:
            l = len(q)
            level = []
            for i in range(l):
                node = q.popleft()
                if node:
                    level.append(node.val)
                    q.append(node.left)
                    q.append(node.right)
            if level:
                res.append(level)
        return res
```

1.