


1161. Maximum Level Sum of a Binary Tree

Solved 

Medium

 Topics

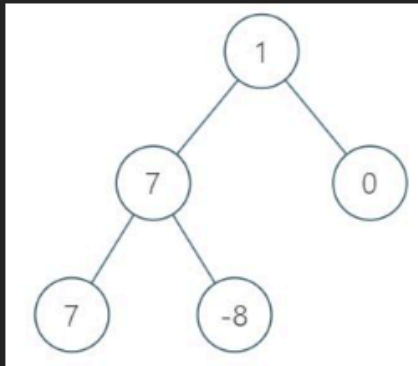
 Companies

 Hint

Given the `root` of a binary tree, the level of its root is `1`, the level of its children is `2`, and so on.

Return the **smallest** level `x` such that the sum of all the values of nodes at level `x` is **maximal**.

Example 1:



Input: `root = [1,7,0,7,-8,null,null]`

Output: `2`

Explanation:

Level 1 sum = 1.

Level 2 sum = 7 + 0 = 7.

Level 3 sum = 7 + -8 = -1.

So we return the level with the maximum sum which is level 2.

Example 2:

Input: `root = [989,null,10250,98693,-89388,null,null,null,-32127]`

Output: `2`

Constraints:

- The number of nodes in the tree is in the range `[1, 104]`.
- `-105 <= Node.val <= 105`

Python:

Definition for a binary tree node.

class TreeNode:

def __init__(self, val=0, left=None, right=None):

```

#     self.val = val
#     self.left = left
#     self.right = right
class Solution:
    def maxLevelSum(self, root: Optional[TreeNode]) -> int:
        idx, Sum=0, -inf
        q=deque()
        q.append(root)
        level=1
        while q:
            qz=len(q)
            curSum=0
            for i in range(qz):
                Node=q.popleft()
                curSum+=Node.val
                if Node.left: q.append(Node.left)
                if Node.right: q.append(Node.right)
            if curSum>Sum:
                idx, Sum=level, curSum
            level+=1
        return idx

```

JavaScript:

```

var maxLevelSum = function(root) {
    let queue = [root];
    let level = 1;
    let answerLevel = 1;
    let maxSum = -Infinity;

    while (queue.length > 0) {
        let size = queue.length;
        let levelSum = 0;

        for (let i = 0; i < size; i++) {
            let node = queue.shift();
            levelSum += node.val;

            if (node.left) queue.push(node.left);
            if (node.right) queue.push(node.right);
        }

        if (levelSum > maxSum) {
            maxSum = levelSum;
            answerLevel = level;
        }
    }
}

```

```

        level++;
    }

    return answerLevel;
};

```

Java:

```

/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode() {}
 *     TreeNode(int val) { this.val = val; }
 *     TreeNode(int val, TreeNode left, TreeNode right) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }
 */
class Solution {
    public int maxLevelSum(TreeNode root) {
        Queue<TreeNode> q = new LinkedList<>();
        q.add(root);
        int max = Integer.MIN_VALUE;
        int ans = 1;
        int curr = 1;
        while (!q.isEmpty()) {
            int c = 0;
            int size = q.size();
            for (int i = 0; i < size; i++) {
                TreeNode temp = q.remove();
                c += temp.val;
                if (temp.left != null)
                    q.add(temp.left);
                if (temp.right != null)
                    q.add(temp.right);
            }
            if (c > max) {
                max = c;
                ans = curr;
            }
        }
    }
}

```

```
        curr += 1;
    }
    return ans;
}
}
```