

# GRAPH (BFS)

# Problem – Maximum Level Sum of a Binary Tree

Medium

<https://leetcode.com/problems/maximum-level-sum-of-a-binary-tree>

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**.

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

**Output:** 2

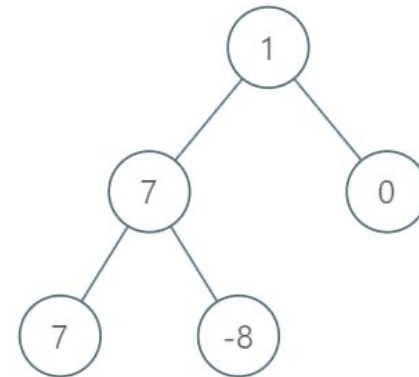
**Explanation:**

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.



# Solution – Maximum Level Sum of a Binary Tree

Medium

<https://leetcode.com/problems/maximum-level-sum-of-a-binary-tree>

```
int maxLevelSum(TreeNode* root) {
    std::queue<TreeNode*> nodes;
    int currentLevel = 0;
    int maxLevel = 1;
    int maxSum = INT_MIN;

    nodes.push(root);

    // traverse the graph
    while(!nodes.empty()) {
        int levelSum = 0;
        int levelSize = nodes.size();
        currentLevel++;

        // sum the values in current level
        for (int i = 0; i < levelSize; ++i) {
            TreeNode* node = nodes.front();
            levelSum += node->val;
            nodes.pop();

            if (node->left) nodes.push(node->left);
            if (node->right) nodes.push(node->right);
        }

        if (levelSum > maxSum) {
            maxLevel = currentLevel;
            maxSum = levelSum;
        }
    }

    return maxLevel;
}
```