

# Problem 513: Find Bottom Left Tree Value

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 72.01%

**Paid Only:** No

**Tags:** Tree, Depth-First Search, Breadth-First Search, Binary Tree

## Problem Description

Given the `root` of a binary tree, return the leftmost value in the last row of the tree.

**Example 1:**



**Input:** root = [2,1,3] **Output:** 1

**Example 2:**



**Input:** root = [1,2,3,4,null,5,6,null,null,7] **Output:** 7

**Constraints:**

\* The number of nodes in the tree is in the range  $[1, 104]$ . \*  $-231 \leq \text{Node.val} \leq 231 - 1$

## Code Snippets

**C++:**

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
```

```

* int val;
* TreeNode *left;
* TreeNode *right;
* TreeNode() : val(0), left(nullptr), right(nullptr) {}
* TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
* TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
right(right) {}
* };
*/

class Solution {
public:
int findBottomLeftValue(TreeNode* root) {

}
};

```

## 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 findBottomLeftValue(TreeNode root) {

}
}

```

## Python3:

```
# 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 findBottomLeftValue(self, root: Optional[TreeNode]) -> int:
```