Description | Solution | Submissions | Discuss (21)
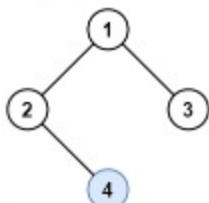
1469. Find All the Lonely Nodes

Easy   👍 9   👎 0   ♡ Add to List   ⬆ Share

In a binary tree, a **lonely** node is a node that is the only child of its parent node. The root of the tree is not lonely because it does not have a parent node.

Given the `root` of a binary tree, return *an array containing the values of all lonely nodes* in the tree. Return the list **in any order**.

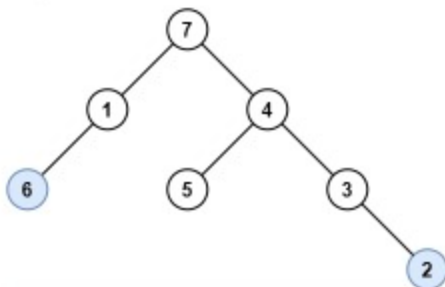**Example 1:**


```
Input: root = [1,2,3,null,4]
Output: [4]
Explanation: Light blue node is the only lonely node.
Node 1 is the root and is not lonely.
Nodes 2 and 3 have the same parent and are not lonely.
```
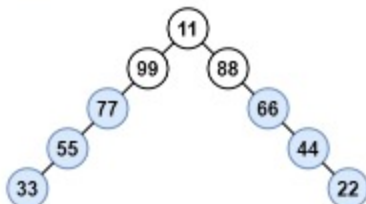
Example 2:


```
Input: root = [7,1,4,6,null,5,3,null,null,null,null,null,2]
Output: [6,2]
Explanation: Light blue nodes are lonely nodes.
Please remember that order doesn't matter, [2,6] is also an acceptable answer.
```

Example 3:


```
Input: root = [11,99,88,77,null,null,66,55,null,null,44,33,null,null,22]
Output: [77,55,33,66,44,22]
Explanation: Nodes 99 and 88 share the same parent. Node 11 is the root.
All other nodes are lonely.
```

Example 4:

```
Input: root = [197]
Output: []
```

Example 5:

```
Input: root = [31,null,78,null,28]
Output: [78,28]
```

Constraints:

- The number of nodes in the `tree` is in the range `[1, 1000]`.
- Each node's value is between `[1, 10^6]`.

Accepted 226 | Submissions 268

Seen this question in a real interview before?   Yes   No

Contributor

Companies 🔒 ⓘ

Related Topics

Similar Questions

Show Hint 1

Show Hint 2

```cpp
/**
 * 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:
    vector<int> getLonelyNodes(TreeNode* root)
    {

    }
};
```

Console ▾   Contribute ⌀

Problems   ✕ Pick One   ‹ Prev 1469/1469 Next ›   ▶ Run Code   Submit