

Problem List

DescriptionEditorialSubmissionsSolutions

1469. Find All The Lonely Nodes

Premium

Solved

EasyTopicsCompaniesHint

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:

```
graph TD; 1((1)) --- 2((2)); 1 --- 3((3)); 2 --- 4((4)); style 4 fill:#add8e6
```

Input:

Output:

Explanation:

root = [1,2,3,null,4]  
[4]  
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:

```
graph TD; 7((7)) --- 1((1)); 7 --- 4((4)); 1 --- 6((6)); 4 --- 5((5)); 4 --- 3((3)); 3 --- 2((2)); style 6 fill:#add8e6; style 2 fill:#add8e6
```

Input:

Output:

Explanation:

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

Example 3:

```
graph TD; 11((11)) --- 99((99)); 11 --- 88((88)); 99 --- 77((77)); 99 --- 55((55)); 77 --- 33((33)); 77 --- 55; 88 --- 66((66)); 88 --- 44((44)); 66 --- 44; 66 --- 22((22)); style 33 fill:#add8e6; style 55 fill:#add8e6; style 66 fill:#add8e6; style 44 fill:#add8e6; style 22 fill:#add8e6
```

Input:

Output:

Explanation:

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

Constraints:

- The number of nodes in the `tree` is in the range `[1, 1000]`.
- `1 <= Node.val <= 106`

Seen this question in a real interview before?

1/5

Yes

No

Accepted

54.8K

Submissions

65.5K

Acceptance Rate

83.7%

Topics

Companies

Hint 1

Similar Questions

Discussion 9

Copyright © 2024 LeetCode All rights reserved

506

9

☆

📄

🔒

Code

Python3

Auto

```
1 # Definition for a binary tree node.
2 # class TreeNode:
3 #     def __init__(self, val=0, left=None, right=None):
4 #         self.val = val
5 #         self.left = left
6 #         self.right = right
7 class Solution:
8     def getLonelyNodes(self, root: Optional[TreeNode]) -> List[int]:
9         self.lonely_nodes = []
10
11         def dfs(node):
12             # We'll append the child's value only if it is an only child
13             if node.left and not node.right:
14                 self.lonely_nodes.append(node.left.val)
15             if node.right and not node.left:
16                 self.lonely_nodes.append(node.right.val)
17
18             # Then dfs left and/or right, whichever is present
19             if node.left:
20                 dfs(node.left)
21             if node.right:
22                 dfs(node.right)
23
24         dfs(root)
25         return self.lonely_nodes
26
```

Saved

Ln 8, Col 64

Testcase

Test Result

Accepted

Runtime: 43 ms

Case 1

Case 2

Case 3

Input

root = [1,2,3,null,4]