Problem 1602: Find Nearest Right Node in Binary Tree

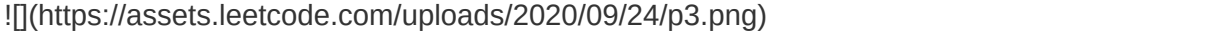
Problem Information

Difficulty: Medium
Acceptance Rate: 75.04%
Paid Only: Yes
Tags: Tree, Breadth-First Search, Binary Tree

Problem Description

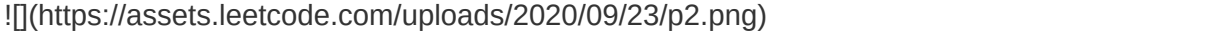
Given the `root` of a binary tree and a node `u` in the tree, return _the**nearest** node on the **same level** that is to the **right** of_ `u` _, or return_ `null` _if_ `u` _is the rightmost node in its level_.

Example 1:



Input: root = [1,2,3,null,4,5,6], u = 4 **Output:** 5 **Explanation:** The nearest node on the same level to the right of node 4 is node 5.

Example 2:



Input: root = [3,null,4,2], u = 2 **Output:** null **Explanation:** There are no nodes to the right of 2.

Constraints:

* The number of nodes in the tree is in the range `[1, 105]`. * `1 <= Node.val <= 105` * All values in the tree are **distinct**. * `u` is a node in the binary tree rooted at `root`.

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:
TreeNode* findNearestRightNode(TreeNode* root, TreeNode* u) {

}
};
```

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 TreeNode findNearestRightNode(TreeNode root, TreeNode u) {
```

```
}
}
```

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 findNearestRightNode(self, root: TreeNode, u: TreeNode) ->
Optional[TreeNode]:
```