Problem 1609: Even Odd Tree

Problem Information

Difficulty: Medium
Acceptance Rate: 66.84%
Paid Only: No
Tags: Tree, Breadth-First Search, Binary Tree

Problem Description

A binary tree is named **Even-Odd** if it meets the following conditions:

* The root of the binary tree is at level index `0`, its children are at level index `1`, their children are at level index `2`, etc. * For every **even-indexed** level, all nodes at the level have **odd** integer values in **strictly increasing** order (from left to right). * For every **odd-indexed** level, all nodes at the level have **even** integer values in **strictly decreasing** order (from left to right).

Given the `root` of a binary tree, _return_`true` _if the binary tree is**Even-Odd** , otherwise return _`false` _._

Example 1:

![](https://assets.leetcode.com/uploads/2020/09/15/sample_1_1966.png)

Input: root = [1,10,4,3,null,7,9,12,8,6,null,null,2] **Output:** true **Explanation:** The node values on each level are: Level 0: [1] Level 1: [10,4] Level 2: [3,7,9] Level 3: [12,8,6,2] Since levels 0 and 2 are all odd and increasing and levels 1 and 3 are all even and decreasing, the tree is Even-Odd.

Example 2:

![](https://assets.leetcode.com/uploads/2020/09/15/sample_2_1966.png)

Input: root = [5,4,2,3,3,7] **Output:** false **Explanation:** The node values on each level are: Level 0: [5] Level 1: [4,2] Level 2: [3,3,7] Node values in level 2 must be in strictly increasing order, so the tree is not Even-Odd.

Example 3:

![](https://assets.leetcode.com/uploads/2020/09/22/sample_1_333_1966.png)

Input: root = [5,9,1,3,5,7] **Output:** false **Explanation:** Node values in the level 1 should be even integers.

Constraints:

* The number of nodes in the tree is in the range `[1, 105]`. * `1 <= Node.val <= 106`

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:
bool isEvenOddTree(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 boolean isEvenOddTree(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 isEvenOddTree(self, root: Optional[TreeNode]) -> bool:
```