

5532. Even Odd Tree

[My Submissions \(/contest/weekly-contest-209/problems/even-odd-tree/submissions/\)](/contest/weekly-contest-209/problems/even-odd-tree/submissions/)

[Back to Contest \(/contest/weekly-contest-209/\)](/contest/weekly-contest-209/)

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.

User Accepted: 0

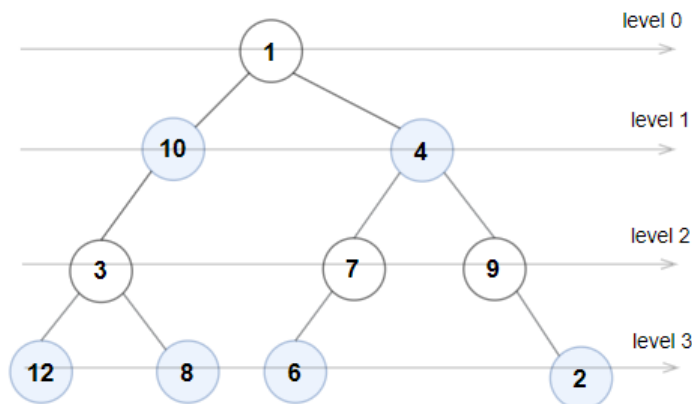
User Tried: 0

Total Accepted: 0

Total Submissions: 0

Difficulty: Medium

Example 1:



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]

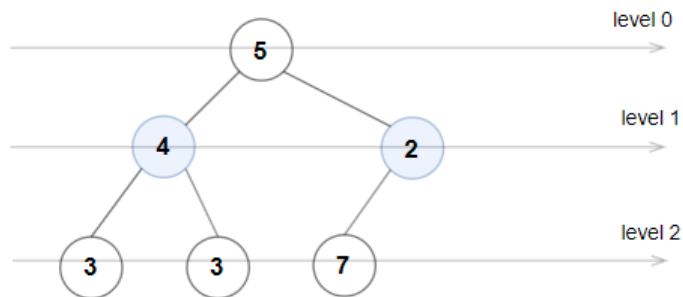
Level 1: [10,4]

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 an Even-Odd tree.

Example 2:



Input: root = [5,4,2,3,3,7]

Output: false

Explanation: The node values on each level are:

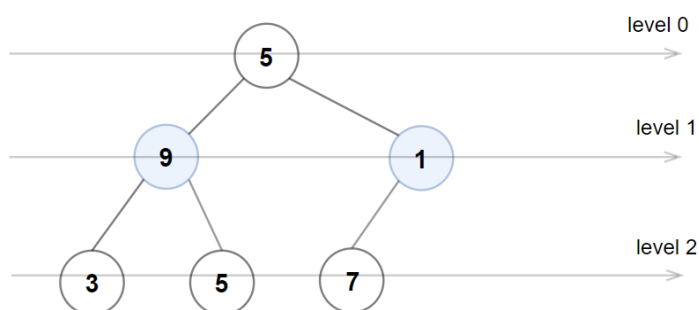
Level 0: [5]

Level 1: [4,2]

Level 2: [3,3,7]

Node values in the level 2 must be in strictly increasing order, so the tree is not Even-Odd Tree.

Example 3:



Input: root = [5,9,1,3,5,7]

Output: false

Explanation: Node values in the level 1 should be even integers.

Example 4:

Input: root = [1]

Output: true

Example 5:

Input: root = [11,8,6,1,3,9,11,30,20,18,16,12,10,4,2,17]

Output: true

Constraints:

- The number of nodes in the tree is in the range $[1, 10^5]$.
- $1 \leq \text{Node.val} \leq 10^6$

JavaScript



```
1 /**
2  * Definition for a binary tree node.
3  * function TreeNode(val, left, right) {
4  *     this.val = (val===undefined ? 0 : val)
5  *     this.left = (left===undefined ? null : left)
6  *     this.right = (right===undefined ? null : right)
7  * }
8  */
9 /**
10  * @param {TreeNode} root
11  * @return {boolean}
12  */
13 const isEvenOddTree = (root) => {
14     let data = levelOrder_BFS(root);
15     let n = data.length;
16     for (let i = 0; i < n; i++) {
17         if (i % 2 == 0) {
18             if (!isAscending(data[i]) || !isAllOdd(data[i])) {
19                 return false;
20             }
21         } else {
22             if (!isDescending(data[i]) || !isAllEven(data[i])) {
23                 return false;
24             }
25         }
26     }
27     return true;
28 };
29
30 const levelOrder_BFS = (root) => {
31     let data = [];
32     getAllLevels(root, 0, data);
33     return data;
34 };
35
36 const getAllLevels = (root, level, data) => {
37     if (!root) return;
38     if (level >= data.length) {
39         let list = [];
40         data.push(list);
41     }
42     data[level].push(root.val);
43     getAllLevels(root.left, level + 1, data);
44     getAllLevels(root.right, level + 1, data);
45 };
46
47 const isAscending = (arr) => {
48     return arr.every((x, i) => {
```

```
49         return i === 0 || x > arr[i - 1];
50     });
51 };
52
53 ▼ const isDescending = (arr) => {
54 ▼     return arr.every((x, i) => {
55         return i === 0 || x < arr[i - 1];
56     });
57 };
58
59 ▼ const isAllOdd = (arr) => {
60 ▼     for (const i of arr) {
61         if (i % 2 === 0) return false;
62     }
63     return true;
64 };
65
66 ▼ const isAllEven = (arr) => {
67 ▼     for (const i of arr) {
68         if (i % 2 === 1) return false;
69     }
70     return true;
71 };
```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)

Submission Result: **Accepted** (/submissions/detail/404188194/) ?

[More Details > \(/submissions/detail/404188194/\)](#)

Share your acceptance!

Copyright © 2020 LeetCode

[Help Center \(/support/\)](#) | [Students \(/students\)](#) | [Terms \(/terms/\)](#) | [Privacy Policy \(/privacy/\)](#)

 [United States \(/region/\)](#)