[LeetCode](https://leetcode.com/problems/even-odd-tree/?envType=daily-question&envId=2024-02-29)

<https://github.com/AdityaKonda6/-50DaysOfCoding>

<https://leetcode.com/problems/even-odd-tree/?envType=daily-question&envId=2024-02-29>

<https://www.linkedin.com/in/aditya-adi-konda/>

 Day 10 of [#50dayscodingchallenge](https://www.linkedin.com/feed/hashtag/?keywords=50dayscodingchallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633):  
[#leetcode](https://www.linkedin.com/feed/hashtag/?keywords=leetcode&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcodechallenge](https://www.linkedin.com/feed/hashtag/?keywords=leetcodechallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcodestreak](https://www.linkedin.com/feed/hashtag/?keywords=leetcodestreak&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcode2024](https://www.linkedin.com/feed/hashtag/?keywords=leetcode2024&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcode50day](https://www.linkedin.com/feed/hashtag/?keywords=leetcode50day&highlightedUpdateUrns=urn:li:activity:7166316239483461633)  
   
Just kicked off my coding journey with a fascinating problem - "Successfully solved LeetCode Problem “1609. Even Odd Tree” !”  
   
✨ Task: 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.

Examples:

1. Input: [1,10,4,3,null,7,9,12,8,6,null,null,2]

✅ Output: True (Verified Even-Odd Tree)

2. Input: [5,4,2,3,3,7]

❌ Output: False (Not a valid Even-Odd Tree)

3. Input: [5,9,1,3,5,7]

❌ Output: False (Level 1 values should be even integers)

Let's Connect:

If you find this problem intriguing or have insights to share, let's connect! I'm passionate about problem-solving, algorithmic thinking, and collaborative learning. Feel free to comment or reach out for engaging discussions and knowledge exchange.Unravel the mystery using your coding skills!

[#CodingChallenge](https://www.linkedin.com/feed/hashtag/?keywords=codingchallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#Algorithm](https://www.linkedin.com/feed/hashtag/?keywords=algorithm&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#LinkedInPost](https://www.linkedin.com/feed/hashtag/?keywords=linkedinpost&highlightedUpdateUrns=urn:li:activity:7166316239483461633) #Algorithm #Optimization #DataStructures #CodingChallenge  
  
Excited about the progress and challenges ahead!  
   
Make Sure You Follow My GitHub For Solutions: <https://github.com/AdityaKonda6/-50DaysOfCoding>  
  
  
Happy coding!

**Solution:-**

class Solution {

  public boolean isEvenOddTree(TreeNode root) {

    Queue<TreeNode> q = new LinkedList<>(Arrays.asList(root));

    boolean isEven = true;

    for (; !q.isEmpty(); isEven = !isEven) {

      int prevVal = isEven ? Integer.MIN\_VALUE : Integer.MAX\_VALUE;

      for (int sz = q.size(); sz > 0; --sz) {

        TreeNode node = q.poll();

        if (isEven && (node.val % 2 == 0 || node.val <= prevVal))

          return false; // invalid case on even level

        if (!isEven && (node.val % 2 == 1 || node.val >= prevVal))

          return false; // invalid case on odd level

        prevVal = node.val;

        if (node.left != null)

          q.offer(node.left);

        if (node.right != null)

          q.offer(node.right);

      }

    }

    return true;

  }

}

