

6182. Reverse Odd Levels of Binary Tree

[My Submissions \(/contest/weekly-contest-311/problems/reverse-odd-levels-of-binary-tree/submissions/\)](#)
[Back to Contest \(/contest/weekly-contest-311/\)](#)

Given the **root** of a **perfect** binary tree, reverse the node values at each **odd** level of the tree.

- For example, suppose the node values at level 3 are `[2, 1, 3, 4, 7, 11, 29, 18]`, then it should become `[18, 29, 11, 7, 4, 3, 1, 2]`.

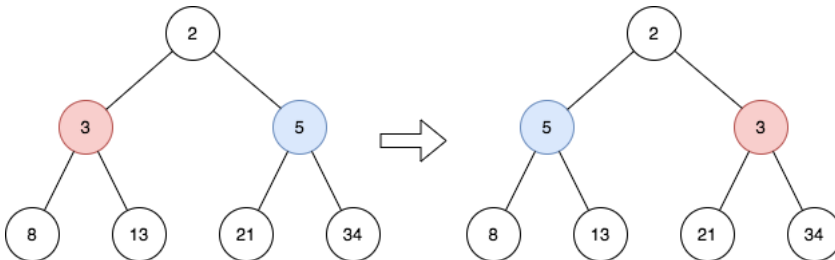
Return *the root of the reversed tree*.

A binary tree is **perfect** if all parent nodes have two children and all leaves are on the same level.

The **level** of a node is the number of edges along the path between it and the root node.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

Example 1:



Input: `root = [2,3,5,8,13,21,34]`

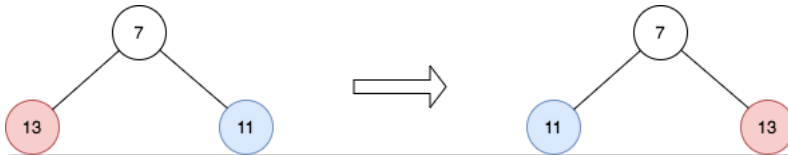
Output: `[2,5,3,8,13,21,34]`

Explanation:

The tree has only one odd level.

The nodes at level 1 are 3, 5 respectively, which are reversed and become 5, 3.

Example 2:



Input: `root = [7,13,11]`

Output: `[7,11,13]`

Explanation:

The nodes at level 1 are 13, 11, which are reversed and become 11, 13.

Example 3:

Input: `root = [0,1,2,0,0,0,0,1,1,1,1,2,2,2,2]`

Output: `[0,2,1,0,0,0,0,2,2,2,2,1,1,1,1]`

Explanation:

The odd levels have non-zero values.

The nodes at level 1 were 1, 2, and are 2, 1 after the reversal.

The nodes at level 3 were 1, 1, 1, 1, 2, 2, 2, 2, and are 2, 2, 2, 2, 1, 1, 1, 1 after the reversal.

Constraints:

- The number of nodes in the tree is in the range `[1, 214]`.
- `0 <= Node.val <= 105`
- `root` is a **perfect** binary tree.

JavaScript



```
1 const reverseOddLevels = (root) => {
2   let d = levelOrderBFS(root), a = [];
3   for (let i = 0; i < d.length; i++) {
```

```

4      if (i & 1) d[i].reverse();
5      }
6      for (const e of d) {
7          for (const x of e) a.push(x);
8      }
9      let res = buildTree(a);
10     // printTree(res)
11     return res;
12 };
13
14 const levelOrder_BFS = (root) => {
15     let data = [];
16     getAllLevels(root, 0, data);
17     return data;
18 };
19
20 const getAllLevels = (root, level, data) => {
21     if (!root) return;
22     if (level >= data.length) data.push([]);
23     data[level].push(root.val);
24     getAllLevels(root.left, level + 1, data);
25     getAllLevels(root.right, level + 1, data);
26 };
27
28 const buildTree = (a) => {
29     let i = 0, root = a[i] != null ? new TreeNode(a[i]) : null, q = [root];
30     i++;
31     while (q.length && i < a.length) {
32         let cur = q.shift();
33         if (cur) {
34             cur.left = a[i] != null ? new TreeNode(a[i]) : null;
35             q.push(cur.left);
36             i++;
37             if (i >= a.length) break;
38             cur.right = a[i] != null ? new TreeNode(a[i]) : null;
39             q.push(cur.right);
40             i++;
41         }
42     }
43     return root;
44 };
45
46 const printTree = (root) => { // level order bfs with null
47     let q = [root], a = [];
48     while (q.length) {
49         let cur = q.shift();
50         a.push(cur != null ? cur.val : null);
51         if (cur != null) {
52             q.push(cur.left);
53             q.push(cur.right);
54         }
55     }
56     while (a[a.length - 1] == null) a.pop();
57     console.log(JSON.stringify(a));
58 };

```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)Submission Result: **Accepted** (/submissions/detail/802518673/) ⓘ[More Details > \(/submissions/detail/802518673/\)](#)

Share your acceptance!

◀ 1

Copyright © 2022 LeetCode

[Help Center \(/support\)](#) | [Jobs \(/jobs\)](#) | [Bug Bounty \(/bugbounty\)](#) | [Online Interview \(/interview/\)](#) | [Students \(/student\)](#) | [Terms \(/terms\)](#) | [Privacy Policy \(/privacy\)](#) [United States \(/region\)](#)