

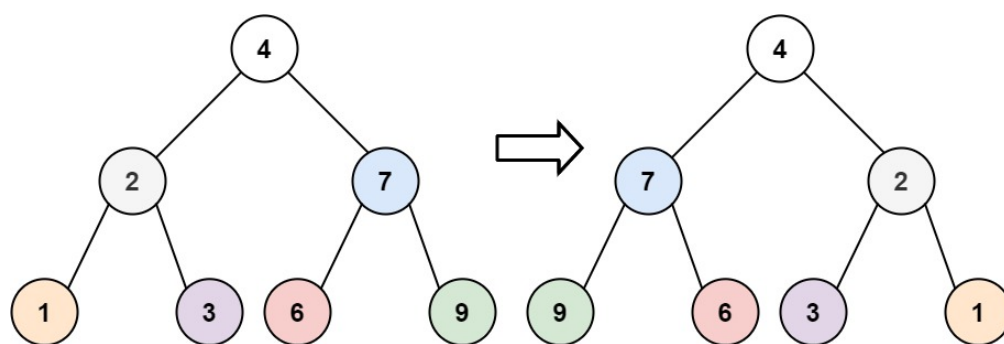
[Description](#)
[Solution](#)
[Discuss \(999+\)](#)
[Submissions](#)

## 226. Invert Binary Tree

Easy
 [9067](#)
[126](#)
[Add to List](#)
[Share](#)

Given the `root` of a binary tree, invert the tree, and return *its root*.

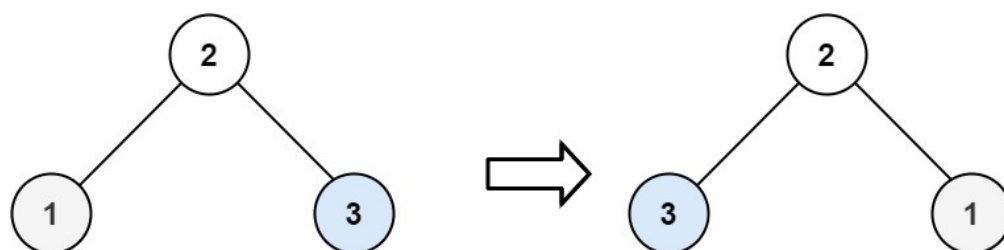
### Example 1:



Input: `root = [4,2,7,1,3,6,9]`

Output: `[4,7,2,9,6,3,1]`

### Example 2:



Input: `root = [2,1,3]`

Output: `[2,3,1]`

### Example 3:

Input: `root = []`

Output: `[]`

### Constraints:

- The number of nodes in the tree is in the range `[0, 100]`.
- `-100 <= Node.val <= 100`

C++

```

1  /**
2   * Definition for a binary tree node.
3   * struct TreeNode {
4   *     int val;
5   *     TreeNode *left;
6   *     TreeNode *right;
7   * };
8   * Definition for a binary tree node.
9   * struct TreeNode {
10    *     int val;
11    *     TreeNode *left;
12    *     TreeNode *right;
13    * };
14    class Solution {
15    public:
16        TreeNode* invertTree(TreeNode* root) {
17            if (root == null) return null;
18            root->left = invertTree(root->right);
19            root->right = invertTree(root->left);
20            return root;
21        }
22    };
23    
```

Testcase Run Code Re

Accepted Runtime

Your input [4,

Output [4,

Expected [4,

Console Use

Run Code ^

Problems

Pick One

< Prev

226/2344

Next >