Problem 314: Binary Tree Vertical Order Traversal

Problem Information

Difficulty: Medium
Acceptance Rate: 57.62%
Paid Only: Yes
Tags: Hash Table, Tree, Depth-First Search, Breadth-First Search, Sorting, Binary Tree

Problem Description

Given the `root` of a binary tree, return _**the vertical order traversal** of its nodes' values_. (i.e., from top to bottom, column by column).

If two nodes are in the same row and column, the order should be from **left to right**.

Example 1:

![](https://assets.leetcode.com/uploads/2024/09/23/image1.png)

Input: root = [3,9,20,null,null,15,7] **Output:** [[9],[3,15],[20],[7]]

Example 2:

![](https://assets.leetcode.com/uploads/2024/09/23/image3.png)

Input: root = [3,9,8,4,0,1,7] **Output:** [[4],[9],[3,0,1],[8],[7]]

Example 3:

![](https://assets.leetcode.com/uploads/2024/09/23/image2.png)

Input: root = [1,2,3,4,10,9,11,null,5,null,null,null,null,null,null,null,6] **Output:** [[4],[2,5],[1,10,9,6],[3],[11]]

Constraints:

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

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:
vector<vector<int>> verticalOrder(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 List<List<Integer>> verticalOrder(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 verticalOrder(self, root: Optional[TreeNode]) -> List[List[int]]:
```