Vertical Order Traversal of a Binary Tree

Try to solve the Vertical Order Traversal of a Binary Tree problem.

We'll cover the following

Statement

Find the vertical order traversal of a binary tree when the root of the binary tree is given. In other words, return the values of the nodes from top to bottom in each column, column by column from left to right. If there is more than one node in the same column and row, return the values from left to right.

Constraints:

- The number of nodes in the tree is in the range `[1, 1000]`.
- $-1000 \leq$ `Node.val` ≤ 1000

Examples

In the slides below, the input parameter is a list that represents the level order traversal of the binary tree.

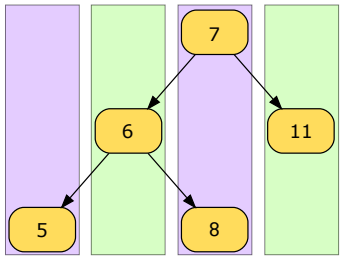
Sample example 1

Input

7	6	11	5	8

Output

[5]	[6]	[7, 8]	[11]



1 of 2

Understand the problem

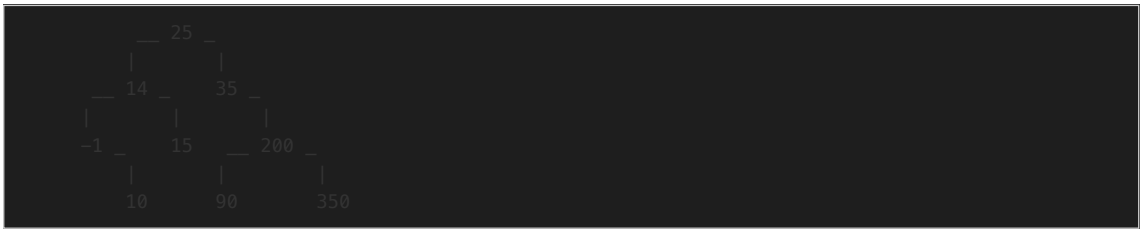
Let's take a moment to make sure you've correctly understood the problem. The quiz below helps you check if you're solving the correct problem:

Vertical Order Traversal of a Binary Tree

1

What should be the output if the following tree is given as input?

tree = [25, 14, 35, -1, 15, 200, 10, 90, 350]



A) [[25], [14, 35], [-1, 15, 200], [10, 90, 350]]

B) [[-1], [14, 10], [25, 15], [35, 90], [200], [350]]

C) [[-1], [14], [10], [25], [15], [35], [90], [200], [350]]

D) [[25, 14, -1], [10], [14, 15], [200, 90], [125, 35, 200, 350]]

Submit Answer

Question 1 of 2
0 attempted

Reset Quiz

Figure it out!

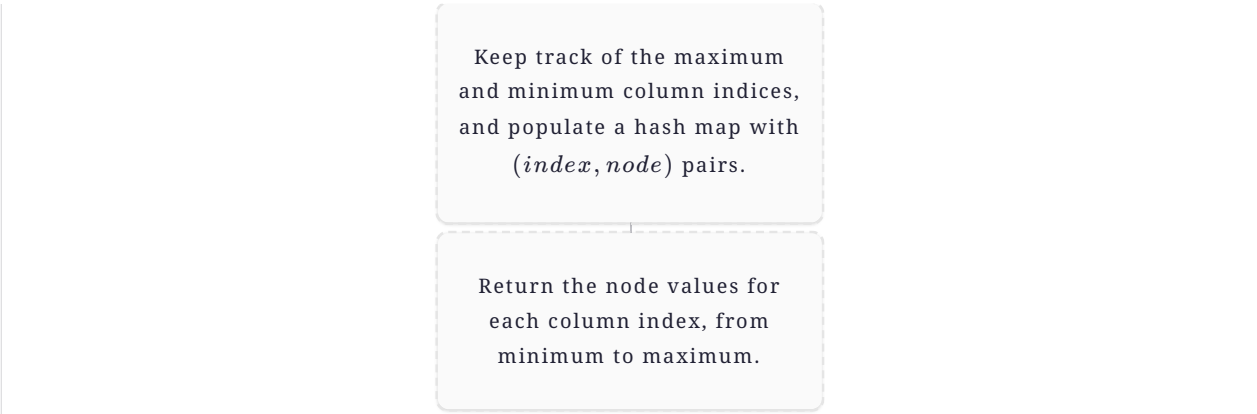
We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

Drag and drop the cards to rearrange them in the correct sequence.

Traverse the tree, level by level, starting from the root node.

Push the nodes to a queue along with their column index.

If a node has children, assign column index $current - 1$ to the left child and $current + 1$ to the right child.



Reset | Show Solution | Submit

Try it yourself

Implement your solution in `main.java` in the following coding playground.

Java

usercode > main.java

```
// Definiton of a binary tree node class
// class TreeNode<T> {
//     T data;
//     TreeNode<T> left;
//     TreeNode<T> right;

//     TreeNode(T data) {
//         this.data = data;
// }

import java.util.*;
import ds_v1.BinaryTree.TreeNode;

public class Main{

  public static List<List<Integer>> verticalOrder(TreeNode<Integer> root) {

        // Your code will replace this placeholder return statement
        return new ArrayList<List<Integer>>();
  }
}
```

Powered by AI | Submit

Test Cases | Results

Case 1 | Case 2 | Case 3

Input #1

[100,50,200,25,75,300,10,350,15]