Binary Tree Zigzag Level Order Traversal

Try to solve the Binary Tree Zigzag Level Order Traversal problem.

We'll cover the following

- Statement
- Examples
- Understand the problem
- Figure it out!
- Try it yourself

Statement

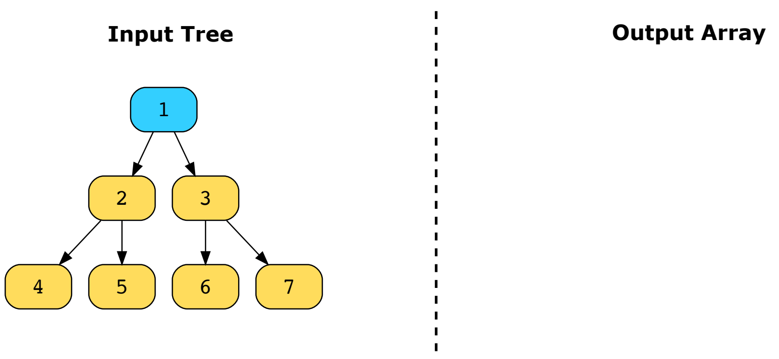
Given a binary tree, return its zigzag level order traversal. The zigzag level order traversal corresponds to traversing nodes from left to right for one level, and then right to left for the next level, and so on, reversing direction after every level.

Constraints:

- The number of nodes in the tree is in the range 0 to 2000.
- $-100 \leq$ `node.data` ≤ 100

Examples

Sample example # 1



We provide the root of the tree as the input. The output is an array of arrays, with each array representing a certain level.

1 of 5

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:

Binary Tree Zigzag Level Order Traversal

1 What will be the zigzag level order traversal of the following tree?

[0, 2, 4, 1, NULL, 3, -1, 5, 1, NULL, 6, NULL, 8]

A) [[0], [2, 4] , [1, NULL, 3, -1], [5, 1, NULL, 6, NULL, 8]]

B) [[0], [4, 2], [1, 3, -1], [8, 6, 1, 5]]

C) [0, 4, 2, 1, 3, -1, 8, 6, 1, 5]

D) [NULL]

Submit Answer

Question 1 of 2
0 attempted

Reset Quiz

Figure it out!

Here's 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.

Add the root of the tree to the deque and initialize the order flag `reverse` to FALSE.

Iterate over the deque as long as it's not empty.

For each dequed element, add its children to the deque from the front if the value of `reverse` is TRUE. Otherwise, append them to the back of the deque.

To deque elements, enque from the back if the value of `reverse` is TRUE. Otherwise, enque them from the front and maintain the order of the nodes.

For each level, append the array of its nodes to the `results` array.

Reset | Show Solution | Submit

Try it yourself

Implement your solution in `main.java` in the following coding playground. You will need the provided supporting code to implement your solution.

Java

usercode > main.java

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

// Definiton of a binary tree node class
// class TreeNode<T> {
```

```java

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

public class Main{
    public static List<List<Integer>> zigzagLevelOrder(TreeNode<Integer> root) {

      // Write your code here
      return new ArrayList<List<Integer>>();
    }
}
```

Powered by AI | Submit

Test Cases | Results

Case 1 | Case 2 | Case 3

Input #1

[2,1]

Binary Tree Zigzag Level Order Traversal

✓ Mark as Completed