Flatten Binary Tree to Linked List

Try to solve the Flatten Binary Tree to Linked List problem.

We'll cover the following

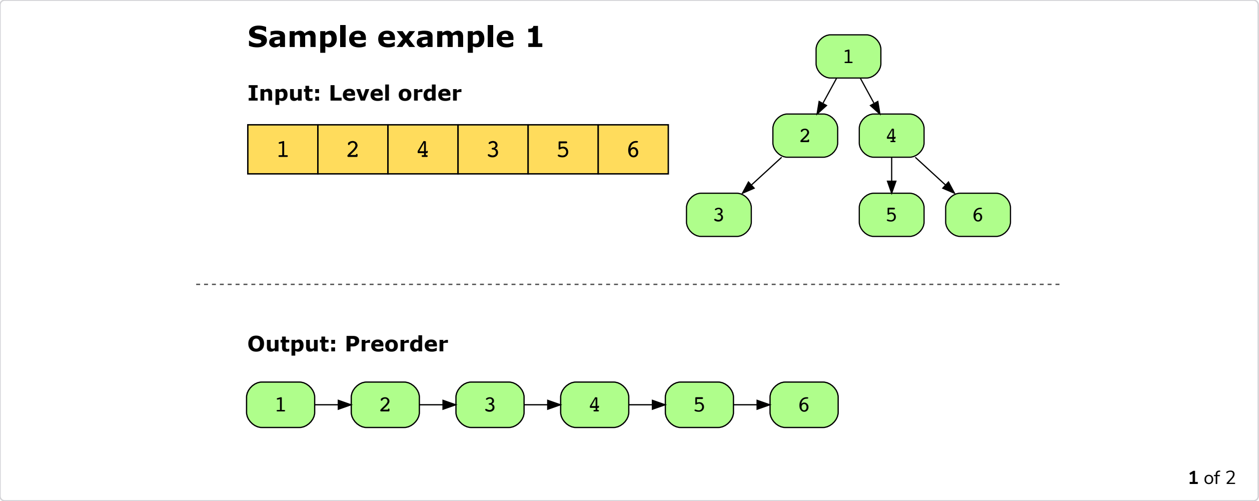
Statement

Given the `root` of a binary tree, the task is to flatten the tree into a linked list using the same `TreeNode` class. The left child pointer of each node in the linked list should always be NULL, and the right child pointer should point to the next node in the linked list. The nodes in the linked list should be in the same order as that of the preorder traversal of the given binary tree.

Constraints:

- $-100 \leq$ `Node.data` ≤ 100.
- The tree contains nodes in the range $[1, 500]$.

Examples



Sample example 1

Input: Level order

1	2	4	3	5	6

Output: Preorder

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:

Flatten Binary Tree to Linked List

1 Choose the correct flattened linked list for the given binary tree:

```
      3
     / \
    2   17
   / \  / \
  1   4 19  5
```

A) 3 → 2 → 17 → 1 → 4 → 19 → 5

B) 3 → 2 → 1 → 4 → 17 → 19 → 5

C) 3 → 2 → 1 → 4 → 19 → 17 → 5

D) 17 → 19 → 5 → 3 → 2 → 1 → 4

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.

For every node, check whether or not it has a left child. If it does not have a left child, we move to the right child.

Otherwise, find the node on the rightmost branch of the left subtree that does not have a right child.

Once we find this rightmost node, connect it with the right child of the current node. After connecting, set the right child of the current node to the left child of the current node.

Finally, set the left child of the current node to NULL.

Repeat the process until the given binary tree becomes flattened.

Reset Show Solution Submit

Try it yourself

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

Java

```java
import ds_v1.BinaryTree.TreeNode;

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

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

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

        // your code will replace this placeholder return statement
        return root;
    }
}
```

Powered by AI ▷ Submit

Test Cases Results

Case 1 Case 2 Case 3

Input #1

```
[3,2,17,1,4,19,5]
```

Flatten Binary Tree to Linked List

← Back

Tree Depth-first Searc...

Next →

Solution: Flatten Binar...