Connect All Siblings of a Binary Tree

Try to solve the Connect All Siblings of a Binary Tree problem.

Statement

The task is to connect all nodes in a binary tree. Connect them from left to right so that the next pointer of each node points to the node on its immediate right. The next pointer of the right-most node at each level should point to the first node of the next level in the tree.
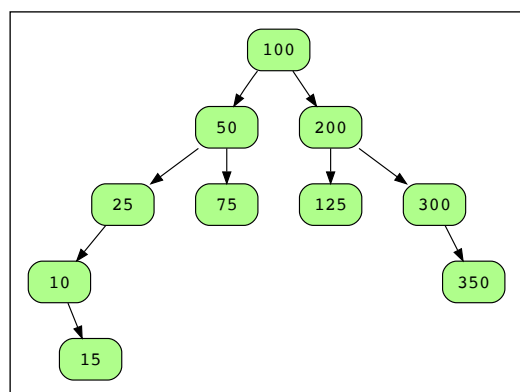
Each node in the given binary tree for this problem includes a `next` pointer, along with the left and right pointers. Your solution must set the `next` pointer to connect the same level nodes to each other and across levels.
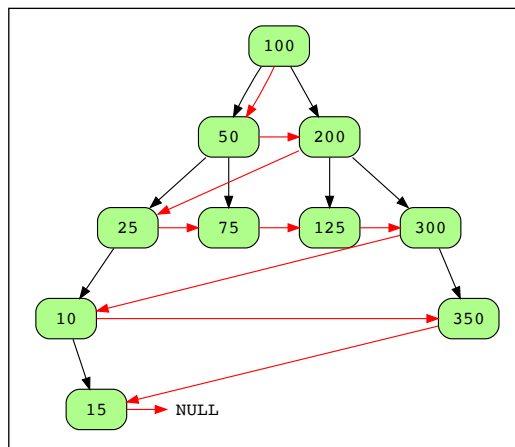
Constraints:

- The number of nodes in the tree is in the range $[0, 2^{12} - 1]$.
- $-1000 \leq$ `Node.val` ≤ 1000

Examples

Consider the following binary tree:

Here's how the final tree looks like when all the next pointers are connected:
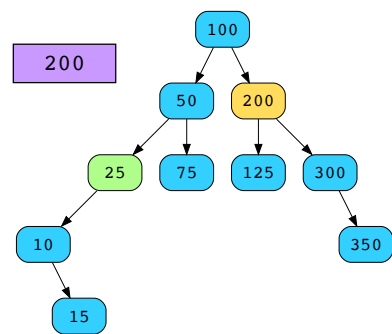
The input will be a list representing the level-order traversal of the binary tree followed by a value that represents the node whose next node we need to find.

Sample example 1

Input

| 100 | 50 | 200 | 25 | 75 | 125 | 300 | 10 | 350 | 15 |

| 200 |

Output

next node: 25

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:

Connect All Siblings of a Binary Tree

1 What should be the next node if the following tree and node value are given as input?

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

node value = 14

A) -1

B) 15

C) 35

D) 25

Submit Answer

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.

> **Note:** As an additional challenge, we have intentionally hidden the solution to this puzzle.

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

Update the current node to the current node's next node.

Initialize two pointers, *current* and *last,* to root.

Else, if the current node has a right child,
set the *next* pointer of the last node to this right child,
and set the *last* pointer to this right child.

If the current node has a left child,
set the *next* pointer of the last node to this left child and set the *last* pointer to this left child.

?

Tᴛ

☾

Return the *next* node of the
desired node.

Reset   Submit

Try it yourself

Implement your solution in `NextRightPointers.java` in the following coding playground. You'll need the provided supporting code to implement your solution.

> **Note:** The binary tree node's class has members `left` and `right` to store references to other nodes, along with the member `data` to hold the node's value.

> **Note:** We have left the solution to this challenge as an exercise for you. You may try to translate the logic of the solved puzzle into a coded solution.

Java

EduBinaryTree.java

```
1    public class NextRightPointers{
4        public static void populateNextNodePointers(EduTreeNode<Integer> node) {
5            // Write your code here
6
7        }
8
9        // Do not modify the code below
10       // Function to find the given node and return its next node
11       public static EduTreeNode<Integer> getNextNode(EduTreeNode<Integer> node,
12
13           // Performing Binary Search
14           while (node != null && nodeData != node.data) {
15               if (nodeData < node.data) {
16                   node = node.left;
17               } else {
18                   node = node.right;
19               }
20           }
21
22           if (node != null) {
23               return node.next;
24           } else {
25               return null;
26           }
27       }
28   }
```

Powered by AI   ▷   Submit

Test Cases   Results

Case 1   Case 2   Case 3

Input #1

[100,50,200,25,75,125,250,10,30]

Input #2

50

Connect All Siblings of a Binary Tree