Populating Next Right Pointers in Each Node

Try to solve the Populating Next Right Pointers in Each Node problem.

We'll cover the following

Statement

Given a binary tree, connect all nodes of the same hierarchical level. We need to 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 will be NULL.

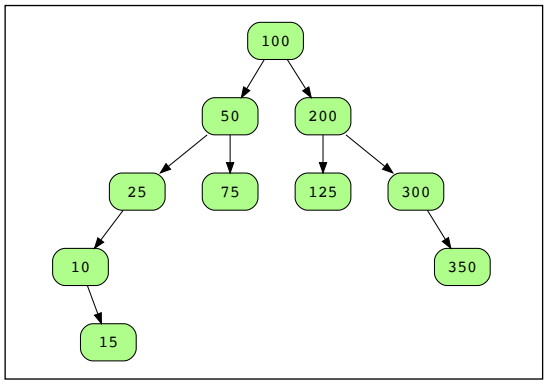
For this problem, each node in the binary tree has one additional pointer (the `next` pointer) along with the `left` and `right` pointers.

Constraints:

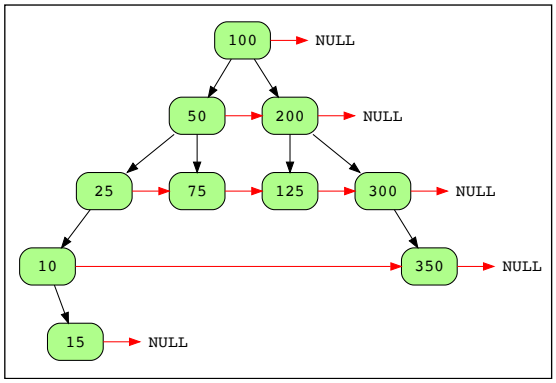
- The number of nodes in the tree is in the range $[0, 2^{12} - 1]$.
- $-1000 \leq$ `Node.data` ≤ 1000

Examples

Consider the following binary tree as an example:



After connecting the nodes at each level, the following is what the tree will look like:



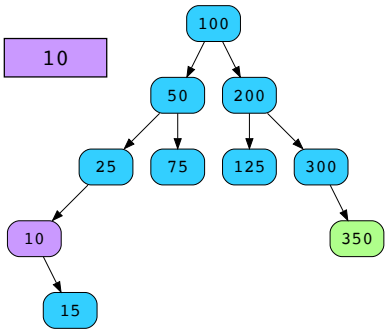
In the slides below, the first input parameter is a list that represents the level-order traversal of the binary tree. The second input parameter represents the node whose next node we need to find.

Sample example 1

Input

100	50	200	25	75	125	300	10	350	15

10



Output

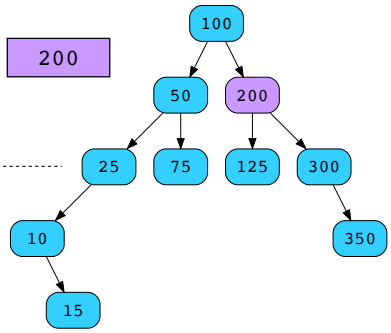
next node: 350

Sample example 2

Input

100	50	200	25	75	125	300	10	350	15

200



Output

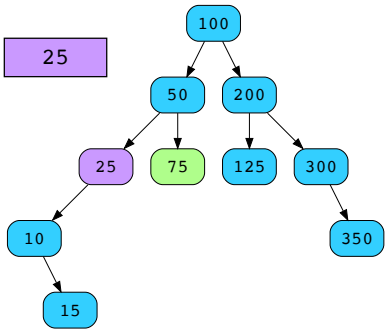
next node: NULL

Sample example 3

Input

100	50	200	25	75	125	300	10	350	15

25



Output

next node: 75

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:

Populating Next Right Pointers in Each Node

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

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

node value = 14

```
          __ 25 _
         |       |
      __ 14 _    35 _
     |       |       |
    -1 _    15    __ 200 _
         |       |        |
        10      90       350
```

A) 35

B) -1

C) 15

D) 25

Submit Answer

Question 1 of 3
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.

If both children of the current node exist, first connect them with each other and then with the previous nodes at the same level.

Else, if only the left child of the current node exists, connect it with the previous nodes at the same level.

Else, if only the right child of the current node exists, connect it with the previous nodes at the same level.

Set the next pointer of the right-most node to NULL and move to the next level.

Stop traversing the tree when all nodes have been visited.

Reset | Show Solution | 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` that hold the node's value.

NextRightPointers.java

EduTreeNode.java

EduBinaryTree.java

```
public class NextRightPointers{
    // Function to populate same level pointers
    public static void populateNextPointers(EduTreeNode<Integer> node) {
        // Write your code here

    }

    // Do not modify the code below
    // Function to find the given node and return its next node
    public static EduTreeNode<Integer> getNextNode(EduTreeNode<Integer> node,
        // Performing Binary Search
        while (node != null && nodeData != node.data) {
            if (nodeData < node.data) {
```

```
                node = node.left;
            } else {
                node = node.right;
            }
        }

        if (node != null) {
            return node.next;
        } else {
            return null;
        }
    }
}
```

Powered by AI

Submit

Test Cases Results

Case 1 Case 2 Case 3

Input #1

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

Input #2

50

Populating Next Right Pointers in Each Node

Back

Next