#LeetCode450

https://leetcode.com/problems/delete-node-in-a-bst/description/

Yifeng Zeng

#题目描述

Delete Node in a BST

#思路报告

The problem can be separate into 4 sub problems. Step 1, find the node that needs to be removed (node.val == key); Step 2, from the left children of this node, find the one with the largest value, which is the one one the right most of the left children (TreeNode remove); Step 3, left node.val = remove.val; Step 4, recursively remove that remove node from node.left (deleteNode(node.left, remove.val);)

代码如下：

```
public TreeNode deleteNode(TreeNode root, int key) {
    if (root == null) {
        return root;
    }
    // if (root.left == null && root.right == null && root.val == key) {
    //     return null;
    // }
    TreeNode node = root;
    if (node.val < key) {
        // deleteNode(node.right, key); // 坑
        node.right = deleteNode(node.right, key);
    } else if (node.val > key) {
        // deleteNode(node.left, key); // 坑
        node.left = deleteNode(node.left, key);
    } else if (node.val == key) {
        if (node.left == null) {
            node = node.right;
        } else {
            TreeNode remove = node.left;
            while (remove.right != null) {
                remove = remove.right;
```

```
            }
            node.val = remove.val;
            // deleteNode(node.left, remove.val); // 坑
            node.left = deleteNode(node.left, remove.val);
        }
    }
    // return root;
    return node;
}
```

#套路总结

- Sometime change the node.val is better than change the node.left, node.right pointer.
- Dividing a big problem into a sequence of sub problems is a very impoartant idea.