

450. Delete Node in a BST

Medium

Topics

Companies

Given a root node reference of a BST and a key, delete the node with the given key in the BST. Return *the **root node reference** (possibly updated) of the BST*.

Basically, the deletion can be divided into two stages:

1. Search for a node to remove.
2. If the node is found, delete the node.

```
struct TreeNode* deleteNode(struct TreeNode* root, int key)
{
    if (root == NULL) return root;

    if (key < root->val)
    {
        root->left = deleteNode(root->left, key);
    }
    else if (key > root->val)
    {
        root->right = deleteNode(root->right, key);
    }
    else
    {
        if (root->left == NULL) {
            struct TreeNode* temp = root->right;
            free(root);
            return temp;
        } else if (root->right == NULL) {
            struct TreeNode* temp = root->left;
            free(root);
            return temp;
        }
        struct TreeNode* temp = root->right;
        while (temp && temp->left != NULL)
            temp = temp->left;
        root->val = temp->val;
        root->right = deleteNode(root->right, temp->val);
    }
    return root;
}
```

Accepted Runtime: 4 ms

Case 1

Case 2

Case 3

Input

root =
[5,3,6,2,4,null,7]

key =
3

Output

[5,4,6,2,null,null,7]

Expected

[5,4,6,2,null,null,7]

Accepted Runtime: 4 ms

Case 1

Case 2

Case 3

Input

root =
[5,3,6,2,4,null,7]

key =
0

Output

[5,3,6,2,4,null,7]

Expected

[5,3,6,2,4,null,7]

Accepted Runtime: 4 ms

Case 1

Case 2

Case 3

Input

root =
[]

key =
0

Output

[]

Expected

[]