450. Delete Node in a BST

Add to List
QuestionEditorial Solution
My Submissions

- Total Accepted: **5543**
- Total Submissions: **15933**
- Difficulty: **Medium**
- Contributors: **tsipporah5945**

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.

Note: Time complexity should be O(height of tree).

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

    5
   / \
  3   6
 / \   \
2   4   7

Given key to delete is 3. So we find the node with value 3 and delete it.

One valid answer is [5,4,6,2,null,null,7], shown in the following BST.

    5
   / \
  4   6
 /     \
2       7

Another valid answer is [5,2,6,null,4,null,7].

    5
   / \
  2   6
   \   \
    4   7
```

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    TreeNode* deleteNode(TreeNode* root, int key) {
        if(root==NULL) return NULL;
        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)
            {
                return root->right;
            }else if (root->right == NULL)
            {
                return root->left;
            }
            TreeNode *minNode = findMin(root->right);
```

```
            root->val = minNode->val;
            root->right = deleteNode(root->right,root->val);
        }
        return root;
    }

    TreeNode *findMin(TreeNode *node)
    {
        while (node->left!=NULL)
            node = node->left;
        return node;
    }
};
```