

Insert a node in a BST



Easy

Accuracy: 47.15%

Submissions: 98K+

Points: 2

Explore Job Fair for students & freshers for daily new opportunities.

Find A Job Today!



Given a BST and a key K. If K is not present in the BST, Insert a new Node with a value equal to K into the BST.

Note: If K is already present in the BST, don't modify the BST.

Example 1:

Input:

2

/ \

1 3

K = 4

Output: 1 2 3 4

Explanation: After inserting the node 4

Inorder traversal will be 1 2 3 4.

```
1 // } Driver Code Ends
115
116
117 // Function to insert a node in a BST.
118 Node* insert(Node* root, int key) {
119     // Your code here
120     if(root == NULL){
121         Node *temp = new Node(key);
122         return temp;
123     }
124     if(root->data == key){
125         return root;
126     }else if(root->data < key){
127         root->right = insert(root->right, key);
128     }else{
129         root->left = insert(root->left, key);
130     }
131     return root;
132 }
133
```

[Custom Input](#)

Compile & Run

Submit



Delete a node from BST

Medium Accuracy: 40.7% Submissions: 89K+ Points: 4

Explore Job Fair for students & freshers for daily new opportunities.

Find A Job Today!

Given a Binary Search Tree and a node value X. Delete the node with the given value X from the BST. If no node with value x exists, then do not make any change.

Example 1:

Input:

```

    2
   / \
  1   3

```

X = 12

Output: 1 2 3

Explanation: In the given input there is no node with value 12 , so the tree will remain same.

C++ (g++ 5.4)

Average Time: 30m

Start Timer

```

110
111 // Function to delete a node from BST.
112 int minval(Node*root){
113     Node*curr = root;
114     while(curr->left) curr = curr->left;
115     return curr->data;
116 }
117 Node *deleteNode(Node *root, int X) {
118     if(root==NULL) return NULL;
119     if(root->data==X){
120         if(!root->left && !root->right){
121             delete(root);
122             return NULL;
123         }if(root->left && !root->right){
124             Node * temp = root->left;
125             delete(root);
126             return temp;
127         }if(!root->left && root->right){
128             Node * temp = root->right;
129             delete(root);
130             return temp;
131         }if(root->left && root->right){
132             int val = minval(root->right);
133             root->data = val;
134             root->right = deleteNode(root->right,val);
135             return root;
136         }
137     }else if(root->data < X){
138         root->right = deleteNode(root->right,X);
139         return root;
140     }else{
141         root->left = deleteNode(root->left,X);
142         return root;
143     }
144     return root;
145 }
146

```



Custom Input

Compile & Run

Submit

