

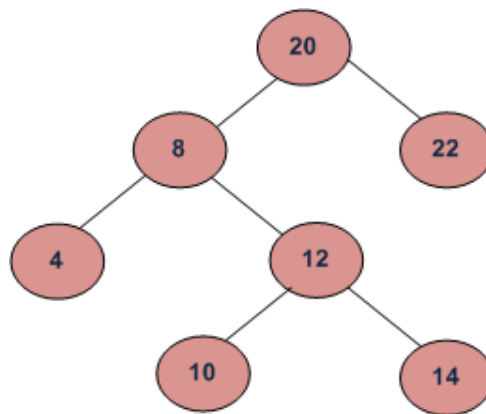


Find k-th smallest element in BST (Order Statistics in BST)

Last Updated : 12 Aug, 2024

Given the root of a binary search tree and K as input, find Kth smallest element in BST.

For example, in the following BST, if $k = 3$, then the output should be 10, and if $k = 5$, then the output should be 14.



Using Inorder ($O(n)$ time and $O(h)$ space)

The [Inorder Traversal](#) of a [BST](#) traverses the nodes in increasing order. So the idea is to traverse the tree in Inorder. While traversing, keep track of the count of the nodes visited. If the count becomes k, print the node.

C++

C

Java

Python

C#

JavaScript

```
#include <iostream>
using namespace std;

// A BST node
struct Node {
    int data;
    Node *left, *right;
    Node(int x) {
        data = x;
        left = right = nullptr;
    }
};
```

```

// Function to find k'th smallest element in BST
Node* kthSmallest(Node* root, int& k, int& count) {
    if (root == nullptr)
        return nullptr;

    Node* left = kthSmallest(root->left, k, count);
    if (left != nullptr)
        return left;

    count++;
    if (count == k)
        return root;

    return kthSmallest(root->right, k, count);
}

// Method to initialize count and call kthSmallest
Node* findKthSmallest(Node* root, int k) {
    int count = 0;
    return kthSmallest(root, k, count);
}

// Recursive function to insert a key into BST
Node* insert(Node* root, int x) {
    if (root == nullptr)
        return new Node(x);
    if (x < root->data)
        root->left = insert(root->left, x);
    else if (x > root->data)
        root->right = insert(root->right, x);
    return root;
}

int main() {
    Node* root = nullptr;
    int keys[] = { 20, 8, 22, 4, 12, 10, 14 };
    for (int x : keys)
        root = insert(root, x);

    int k = 3;
    Node* res = findKthSmallest(root, k);

    if (res == nullptr)
        cout << "There are less than k nodes in the BST";
    else
        cout << "K-th Smallest Element is " << res->data;

    return 0;
}

```

Output

K-th Smallest Element is 10

Time complexity: $O(n)$ where n is the number of nodes in a binary search tree.