



Experiment2.2

Student Name: Virat Samdarshi

Branch: B.E(CSE)

Semester: Fifth

Subject Name: AP LAB 1

UID: 22BCS12648

Section/Group: IOT_627-B

Date of Performance: 12/09/24

Subject Code: 22CSP-314

- 1. Aim:** You are given a pointer to the root of a binary search tree and values to be inserted into the tree. Insert the values into their appropriate position in the binary search tree and return the root of the updated binary tree. You just have to complete the function.
- 2. Objective:** The objective of the given function is to insert new values into a Binary Search Tree (BST) while maintaining its properties. The function takes a pointer to the root of the BST and a value to be inserted.
- 3. Implementation/Code :-**

```
Node * insert(Node * root, int data) {  
    if (root == NULL) {  
        return new Node(data);  
    }  
    if (data < root->data) {  
        root->left = insert(root->left, data);  
    }  
    else {  
        root->right = insert(root->right, data);  
    }  
  
    return root;  
}
```

```
> #include <iostream>...  
  
class Node {  
public:  
    int data;  
    Node *left;  
    Node *right;  
    Node(int d) {  
        data = d;  
        left = NULL;  
        right = NULL;  
    }  
  
    Node * insert(Node * root, int data) {  
        if (root == NULL) {  
            return new Node(data);  
        }  
        if (data < root->data) {  
            root->left = insert(root->left, data);  
        }  
        else {  
            root->right = insert(root->right, data);  
        }  
  
        return root;  
    }  
}
```

4. Output :-

Congratulations!
You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✔ Sample Test case 0

Input (stdin) [Download](#)

1	6
2	4 2 3 1 7 6

Your Output (stdout)

1	4 2 1 3 7 6
---	-------------

Expected Output [Download](#)

1	4 2 1 3 7 6
---	-------------

5. Time Complexity:

Best Case (Balanced BST): $O(\log n)$

Worst Case (Unbalanced BST): $O(n)$

PROBLEM 2

1.Aim: In this challenge, you are required to implement inorder traversal of a tree. Complete the inorder function in your editor below, which has 1 parameter: a pointer to the root of a binary tree. It must print the values in the tree's inorder traversal as a single line of space-separated values.

2.Objective: The objective of this challenge is to implement the inorder traversal of a binary tree. The function takes a pointer to the root of the binary tree and recursively traverses the tree in left-root-right order.

3.Implementation/Code :-

```
void inOrder(Node *root) {  
    if (root == NULL) {  
        return;  
    }  
  
    inOrder(root->left);  
  
    cout << root->data << " ";  
  
    inOrder(root->right);  
}
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
> #include <bits/stdc++.h> ...
class Node {
public:
    int data;
    Node *left;
    Node *right;
    Node(int d) {
        data = d;
        left = NULL;
        right = NULL;
    }
};

void inOrder(Node *root) {
    if (root == NULL) {
        return;
    }

    inOrder(root->left);

    cout << root->data << " ";

    inOrder(root->right);
}
```

4. Output :-

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

✓ Sample Test case 1

Input (stdin)[Download](#)

1	6
2	1 2 5 3 6 4

Your Output (stdout)

1	1 2 3 4 5 6
---	-------------

Expected Output[Download](#)

1	1 2 3 4 5 6
---	-------------

5. Time Complexity: $O(n)$