Inorder Traversal 5

Problem Submissions Leaderboard Discussions

Write a java program to perform Inorder tree traversal.

Input Format

1 10 1 20 1 30 1 40 1 50 1 60 2 3

Constraints

No Constraints

Output Format

InorderTraversal is: 10 20 30 40 50 60

Sample Input 0

```
1
10
1
20
1
30
1
40
1
50
1
60
2
3
```

Sample Output 0

```
InorderTraversal is:
10 20 30 40 50 60
```

Contest ends in 9 days

Submissions: 109
Max Score: 10
Difficulty: Medium

Rate This Challenge:
☆☆☆☆☆

More

Java 7

```
import java.util.*;
class Node {
    int data;
    Node left;
    Node right;
    public Node( int item) {
        this.data = item;
        this.left = null;
        this.right = null;
    }
}

class StackNode {
    Node node;
    StackNode next;
    public void StackNode(Node b) {
```

```java
        this.node = b;
        this.next = null;
    }
}

public class NonRecursiveInorder {
    StackNode top;
    Node root;
    public void NonRecursiveInorder() {
        top = null;
        root = null;
    }
    boolean isEmpty() {
        if(top == null) {
            return true;
        }
        return false;
    }
    void push(Node b) {
        StackNode temp;
        temp = new StackNode();
        if(temp == null) {
            System.out.printf("Stack is overflow.\n");
        } else {
            temp.node = b;
            temp.next = top;
            top = temp;
        }
    }
    Node peek() {
        if (top == null) {
            return null;
        }
        return top.node;
    }
    Node pop() {
        StackNode temp;
        Node b;
        if(top == null) {
            System.out.printf("Stack is underflow.\n");
            return null;
        } else {
            temp = top;
            top = top.next;
            b = temp.node;
            return b;
        }
    }
    void inorderInBST(Node root) {
        Node curr = root;
        while(true) {
            if(curr!= null) {
                push(curr);
                curr = curr.left;
            } else {
                curr = pop();
                System.out.printf("%d ",curr.data);
                curr = curr.right;
            }
            if(isEmpty() && curr == null)
                break;
        }
    }
/* Insertion into binary search tree */
    Node insertBinarySearchTree(Node root, int item) {

        /* If the tree is empty new node became root */
        if (root == null) {
            root = new Node(item);
            return root;
        }

        /* Otherwise, if item is less then root then recur left side */
        if (item < root.data)
            root.left = insertBinarySearchTree(root.left, item);
        else if (item > root.data)
            root.right = insertBinarySearchTree(root.right, item);

        /* return the root node pointer */
        return root;
    }

    // Driver main method Code
    public static void main(String[] args) {
        NonRecursiveInorder tree = new NonRecursiveInorder();
        Scanner sc = new Scanner(System.in);
        int option;
        int item;
        //System.out.println("Enter 1 to insert\nEnter 2 to display BST in inorder\nEnter 3 to Exit");
        while(true) {
            //System.out.print("Enter your option: ");
            option = sc.nextInt();
            switch(option) {
                default:
                    System.out.println("Enter the right option");
                    break;
                case 1:
                    //System.out.print("Enter the element to insert: ");
                    item = sc.nextInt();
                    tree.root = tree.insertBinarySearchTree(tree.root, item);
                    break;
                case 2:
                    if(tree.root == null) {
                        System.out.println("Tree is empty, root is null");
                    }else {
                        System.out.println("InorderTraversal is:"); tree.inorderInBST(tree.root);
                        System.out.println();
                    }
                    break;
                case 3:
                    return;
            }
        }

    }
}
```

Line: 1 Col: 1

Upload Code as File   Test against custom input   Run Code   Submit Code

Testcase 0 ✓

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
1
10
1
20
1
30
1
40
1
50
1
60
```

```
2
3
```

Your Output (stdout)

```
InorderTraversal is:
10 20 30 40 50 60
```

Expected Output

```
InorderTraversal is:
10 20 30 40 50 60
```