https://course.acciojob.com/idle?question=a6aeb3f2-62db-4840-89f7-7affb74d8643

- EASY
- Max Score: 30 Points

Binary Tree Inorder Traversal

Given the `root` of a binary tree, return *the inorder traversal of its nodes"values*.

Note: You just need to implement the inorderTraversal() function. Input and output have been handled in the driver code.

Input Format

First line contains a string representing the tree as described below.

The values in the string are in the order of level order traversal of the tree where, numbers denote node values, and a character “N” denotes NULL child.

Output Format

Print the inorder traversal of the tree.

Example 1

Input

```
6 5 6 1 1 N 6
```

Output

```
1 5 1 6 6 6
```

Explanation

```
The given tree can be represented as:-
```

```
        6
       / \
      5   6
     / \   \
    1   1   6
The inorder traversal of the tree is :- 1 5 1 6 6 6
```

Example 2

Input

```
1 2 3 2 2 N 6 2 N N N N N N N
```

Output

```
2 2 2 2 1 3 6
```

Explanation

```
The given tree can be represented as: -

            1
           / \
          2   3
         / \   \
        2   2   6
       /
      2
The inorder traversal of the tree is :- 2 2 2 2 1 3 6
```

Constraints

- **The number of nodes in the tree is in the range** `[0, 10^4]`.
- `-1000 <= Node.val <= 1000`
- **The depth of the tree will not exceed** `1000`.

Topic Tags

- Recursion
- Trees

My code

```
// in java
import java.util.*;
import java.lang.*;
import java.io.*;

class Node
{
    int data;
    Node next ,prev;

    Node(int data, Node next,Node prev)
    {
        this.data = data;
        this.next = next;
        this.prev = prev;
    }

    Node() {}
}
public class Main
{
  static Node insert(Node root,int n)
  {
    if(root==null)
    {
      root=new Node(n,null,null);
    return root;
    }
   else if(n< root.data)
```

```
      root.prev= insert( root.prev, n);
    else   if(n>root.data)
      root.next= insert( root.next, n);
    return root;
  }

static void inorder(Node root)
  {
    if(root !=null)
    {
      inorder(root.prev);
      System.out.print(root.data+" ");
      inorder(root.next);
    }
  }
        public static void main (String[] args) throws java.lang.Exception
        {
                //your code here
      Scanner s=new Scanner(System.in);
      int n=s.nextInt();
      //int arr[]=new int[n];
      Node root=null;
      for(int i=0;i<n;i++)
      {
      int m=s.nextInt();
        root=insert( root, m);
      }
        inorder(root);
          }
}
```