https://course.acciojob.com/idle?question=f27668ae-c6cb-4b8a-9b64-7fef45257748

- HARD
- Max Score: 50 Points

Mirror Image of the Binary Tree

You are given the number of nodes present in the tree. You have to input the nodes and form a Binary Search Tree (BST).

BST should be formed in ways like:

Let us consider an array named Val having the values of the nodes. Here, Val[0] will be the root of BST. Then, you have to insert Val[1] in the BST, then insert Val[2] in the BST, and so on...

After forming the BST, convert it into its mirror.

Print the Inorder traversal of the mirror tree.

Input Format

The first line contains an integer n, the number of nodes.

The next line inputs the value of n nodes.

Output Format

Print the Inorder traversal of the mirror tree as a single line of space-separated values.

Example 1

Input

```
6
1 2 5 3 4 6
```

Output

```
6 5 4 3 2 1
```

Explanation

```
The BST is like this:-
```

```
     1
      \
       2
        \
         5
        /  \
       3    6
        \
         4
```

```
Its mirror view is:
```

```
       1
      /
     2
    /
   5
  / \
 6   3
    /
   4
```

```
So,the inorder order traversal of mirror tree results in 6,5,4,3,2,1 as the required
result.
```

Example 2

Input

```
3
2 1 3
```

Output

```
3 2 1
```

Explanation

```
The BST is like this:-
```

```
    2
   / \
  1   3

Its mirror view is:

    2
   / \
  3   1

So, the in-order traversal of mirror tree results in 3,2,1 as the required result.
```

Constraints

1 <= `n` **<= 500**

-100 <= `val[node]` **<= 100**

Topic Tags

- Trees
- BST

My code

```
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 Node mir_img(Node root)
  {
    if(root!=null)
    {
      Node t=root.prev;
      root.prev=root.next;
      root.next=t;
       root.next = mir_img(root.next);
       root.prev = mir_img(root.prev);
      }
       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);
        }
        root=mir_img(root);
          inorder(root);
                }
}
```