

<https://course.acciojob.com/idle?question=e5bc5b24-fb3f-4112-9bb0-461f44306650>

• **HARD**

• **Max Score: 50 Points**

-
-
-
-
-
-
-
-
-
-
-
-

Left view of binary tree

Given the `root` node of a BST, print its left view i.e print all the nodes from top to bottom that will appear while looking the tree from left.

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 left view of the tree as a single line of space-separated values.

Example 1

Input

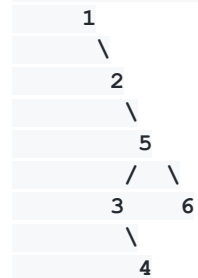
```
6
1 2 5 3 4 6
```

Output

```
1 2 5 3 4
```

Explanation

The BST is like :-



So, the left view of tree results in 1,2,5,3,4 as the required result. only 6 is not visible from left view

Example 2

Input

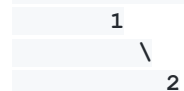
```
3
1 2 3
```

Output

```
1 2 3
```

Explanation

The BST is like :-



```
    \
    3
So, the left view will have all three nodes.
```

Constraints:

```
1 <= n <= 500
-100 <= val[node] <= 100
```

Topic Tags

- Trees

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 int compare_hight=0; //use for level order flag
static void left_view(Node root,int h)
{

    if(root !=null)
    {
        if(h==compare_hight)
        {
            System.out.print(root.data+" ");
            compare_hight++;
        }
        left_view(root.prev,h+1);
        left_view(root.next,h+1);
    }
}

    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);
        }

        left_view(root,0);
    }
}

```