https://course.acciojob.com/idle?question=99d7a5cb-22ad-4a8a-9eec-e1d65b3258cb

- EASY
- Max Score: 30 Points

Kth largest element in Binary Search Tree

Given `root` of binary search tree, and an integer 'K' . Find the Kth largest element in the BST.

Input Format

Here you are given an array as an input and using that array we make binary search tree.

First line of input contains the number of Nodes and an integer K.

Second line contains the value of the nodes.

Output Format

Return the integer which is the Kth largest element in the given tree.

Example 1

Input

```
5 4
2 4 15 6 3
```

Output

```
3
```

Explanation

```
The tree looks like:

    2
     \
      4
     /  \
    3   15
        /
        6

The 4th Largest element in the BST is 3.
```

Example 2

Input

```
3 1
2 1 3
```

Output

```
3
```

Explanation

```
The tree looks like:

    2
   / \
  1   3

The Largest element is 3.
```

Constraints

1 <= Number of nodes <= 1000

1<= K <= Number of nodes

1<= value of each node <= 10000

Topic Tags

- BST

My code

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

class BinarySearchTree
{
    Node constructBST(int[]arr,int start,int end,Node root)
    {
        if(start>end)
            return null;
        int mid=(start+end)/2;

        if(root==null)
            root=new Node(arr[mid]);

        root.left=constructBST(arr,start,mid-1, root.left);
        root.right=constructBST(arr,mid+1,end, root.right);

        return root;

    }
}

public class Main {
    public static void main(String[] args) throws Throwable {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int k = sc.nextInt();
        int arr[]=new int[n];
        for (int i = 0; i < n; i++)
        {
            arr[i] = sc.nextInt();
        }

        Arrays.sort(arr);
        Node root=null;
        BinarySearchTree bst=new BinarySearchTree();
        root=bst.constructBST(arr,0,n-1,root);

        Accio A = new Accio();
        int ans=A.kthLargest(root,k);
        System.out.println(ans);
        sc.close();

    }
}

class Accio
```

```
{
    static int i=0,ans=-1;
    static void fun(Node root, int k)
    {
        if(root==null)
            return;
          fun( root.right,  k);
         i++;
        if(i==k)
            ans=root.data;
        fun( root.left,  k);

    }
  int kthLargest(Node root, int k){
    //Write your code here
      fun(root,k);
      return ans;
  }
}
```