**CSc 2720 - Data Structures: Lab 8**

**Deadline to Submit: [3/5/2021] [11:00pm] ET(US)**

Failure to submit will result in a zero for this lab.

**#Note : Helper Code provided.**

**Problem: [100 points]**

**Find and return the kth largest Node in the Binary Search Tree (BST), here K = 3, so you need to return value 60.**

**BST For us to use in Lab 8. BST Root is 50.**

**50**

**/ \**

**30 70**

**/ \ / \**

**20 40 60 80**

-------------------------------------- Class Template --------------------------------------------

**// A binary tree node**

**class Node {**

**int data;**

**Node left, right;**

**Node(int d)**

**{**

**data = d;**

**left = right = null;**

**}**

**}**

**class BinarySearchTree {**

**// Root of BST**

**Node root;**

**// Constructor**

**BinarySearchTree()**

**{**

**root = null;**

**}**

**// function to insert nodes**

**public void insert(int data)**

**{**

**this.root = this.insertRec(this.root, data);**

**}**

**/\* A utility function to insert a new node**

**with given key in BST \*/**

**Node insertRec(Node node, int data)**

**{**

**//WRITE CODE HERE**

**return node;**

**}**

**// class that stores the value of count**

**public class count {**

**int c = 0;**

**}**

**// utility function to find kth largest node in a given tree**

**void kthLargestUtil(Node node, int k, count C)**

**{**

**// Base cases, the second condition is important to**

**// avoid unnecessary recursive calls**

**if (node == null || C.c >= k)**

**return;**

**//WRITE CODE HERE**

**}**

**// Method to find the kth largest no in given BST**

**void kthLargest(int k)**

**{**

**count c = new count(); // object of class count**

**//WRITE CODE HERE**

**}**

**#Note : Driver Code provided next page.**

**// Driver function**

**public static void main(String[] args)**

**{**

**BinarySearchTree tree = new BinarySearchTree();**

**tree.insert(50);**

**tree.insert(30);**

**//WRITE CODE : BUILD COMPLETE TREE HERE**

**//WRITE CODE HERE TO RETURN Kth largest Node.**

**}**

**}**

**}**