package com.mycompany.main;

import java.util.Random;

public class BST {

node root;

static int size=0;

String str="";

public BST() {

}

public void add(int x){

int m=(int)x;

if(!search(x))

size++;

root=add(root,m);

}

private node add( node root ,int data) {

if(root==null){

root=new node(data);

return root;

}

else if(data<root.data){

root.left=add(root.left,data);

}

else if(data>root.data){

root.right=add(root.right,data);

}

return root;

}

public String inOrder(){

return inOrder(root);

}

public String inOrder(node root){

if(root==null){

return "";

}

inOrder(root.left);

str=str+root.data + " ";

inOrder(root.right);

return str;

}

public boolean search(int x){

if(search(root,x)==null){

return false;

}

return true;

}

private node search(node root, int data) {

if(root==null){

return null;

}

if(root.data==data){

return root;

}

if(data>root.data){

return search(root.right,data);

}

else if(data<root.data){

return search(root.left,data);

}

return null;

}

public void delete(int x){

size--;

root=delete(root,x);

}

private node delete(node root, int data) {

if(root==null){

return root;

}

else{

if(data<root.data){

root.left=delete(root.left,data);

}

else if(data>root.data){

root.right=delete(root.right,data);

}

else{

if(root.left==null){

return root.right;

}

else if(root.right==null){

return root.left;

}

else{

root.data=minval(root.right);

root.right=delete(root.right,root.data);

}

}

}

return root;

}

private int minval(node root) {

int min=root.right.data;

while(root.left!=null){

min=root.left.data;

root=root.left;

}

return min;

}

public void update(int data,int x){

if(search(x))return;

BST l=new BST();

size--;

root=copy(l,this.root,data).root;

add(x);

}

private node update(node root,int data,int x){

if(search(x)==true){

return null;

}

if(root==null){

return root;

}

else{

if(data<root.data){

update(root.left,data,x);

}

else if(data>root.data){

update(root.right,data,x);

}

else{

delete(this.root,data);

add(this.root,x);

}

}

return root;

}

public BST copy(BST l,node root,int m){

if(root==null)return l;

if(root.data!=m){

l.add(root.data);size--;

}

if(root.left!=null)copy(l,root.left,m);

if(root.right!=null)copy(l,root.right,m);

return l;

}

public void m(node root,Random r){//helper method to updete the elemnts in the main class

if(root==null)return ;

if(root.left!=null)m(root.left,r);

update(root.data,r.nextInt());

if(root.right!=null)m(root.right,r);

}