package com.mycompany.main;

import java.time.Clock;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.LinkedList;

import java.util.Random;

public class Main {

public static void main(String[] args) {

long m1=System.currentTimeMillis();

BST b=new BST();

Random r=new Random();

while(b.size<5000){

b.add(r.nextInt());

}

long m2=System.currentTimeMillis();

long m=m2-m1;

long up1=System.currentTimeMillis();

b.m(b.root, r);

long up2=System.currentTimeMillis();

long l=up2-up1;

long in1=System.currentTimeMillis();

b.inOrder();

long in2=System.currentTimeMillis();

long i=in2-in1;

long se1=System.currentTimeMillis();

for(int j=0;j<b.size;j++){

b.search(r.nextInt());

}

long se2=System.currentTimeMillis();

long s=se2-se1;

long de1=System.currentTimeMillis();

for(int h=0;h<b.size;h++){

b.delete(r.nextInt());

}

long de2=System.currentTimeMillis();

long d=de2-de1;

//////////////////////////////////////////////////////////////////////////////////

DoublyLinkerList dd=new DoublyLinkerList();

long a1=System.currentTimeMillis();

for(int g=0;g<5000;g++){

dd.add(r.nextInt());

}

long a2=System.currentTimeMillis();

long t=a2-a1;

long up22=System.currentTimeMillis();

node1 curr=dd.head;

for(int u=0;u<dd.size;u++){

dd.update(curr.data,r.nextInt());

curr=curr.next;

}

long up11=System.currentTimeMillis();

long c=up11-up22;

long p1=System.currentTimeMillis();

dd.print();

long p2=System.currentTimeMillis();

long y=p2-p1;

long s1=System.currentTimeMillis();

for(int q=0;q<dd.size;q++){

dd.search(r.nextInt());

}

long s2=System.currentTimeMillis();

long ss=s2-s1;

long d1=System.currentTimeMillis();

for(int rr=0;rr<dd.size;rr++){

dd.delete(r.nextInt());

}

long d2=System.currentTimeMillis();

long de1e=d2-d1;

// System.out.println("delete do "+de1e );

//////////////////////////////////////////////////////////////////////////

HashMap<Integer,Integer> map=new HashMap<>();

long h1=System.currentTimeMillis();

int k=0;

while(map.size()<5000){

int m11=r.nextInt();

if(!map.containsValue(m11)){

map.put(k, m11);

k++;

}

}

long h2=System.currentTimeMillis();

long h=h2-h1;

long hs1=System.currentTimeMillis();

for(int kk=0;kk<map.size();kk++){

map.containsValue(r.nextInt());

}

long hs2=System.currentTimeMillis();

long hh=hs2-hs1;

long u1=System.currentTimeMillis();

int yy=0;

for(int y1=0;y1<map.size();y1++){

int g=r.nextInt();

if(!(map.get(y1).equals(g))){

map.put(yy,g);

yy++;

}

}

long u2=System.currentTimeMillis();

long u=u2-u1;

long p11=System.currentTimeMillis();

DoublyLinkerList dl=new DoublyLinkerList();//i put the elemnts in the double... to print them in ascending

for(int i11=0;i11<map.size();i11++){//i used the method add but i didnt use the search cause when i insertsd the elemnts in the hash map they werent duplicated

if(map.get(i11)!=null)

dl.add1(map.get(i11));

}

dl.print();

long p22=System.currentTimeMillis();

long p=p22-p11;

long hr1=System.currentTimeMillis();

for(int o=0;o<map.size();o++){

map.remove(o);

}

long hr2=System.currentTimeMillis();

long bb=hr2-hr1;

System.out.println();

System.out.println();

System.out.println();

System.out.println();

System.out.print("DATA STRUCTURE"+"\t");

System.out.print("\t"+"Insert"+" \t");

System.out.print("Remove"+"\t");

System.out.print("Search"+"\t");

System.out.print("Update"+"\t");

System.out.print("Print"+"\t");

System.out.println();

System.out.print("BinarySearchTree"+"\t");

System.out.print(m+"\t");

System.out.print(d+"\t");

System.out.print(s+"\t");

System.out.print(l+"\t");

System.out.print(i+"\t");

System.out.println();

System.out.print("DoublylinkedList"+"\t");

System.out.print(t+"\t");

System.out.print(de1e+"\t");

System.out.print(ss+"\t");

System.out.print(c+"\t");

System.out.print(y+"\t");

System.out.println();

System.out.print("HashMap"+"\t"+"\t"+"\t");

System.out.print(h+"\t");

System.out.print(bb+"\t");

System.out.print(hh+"\t");

System.out.print(u+"\t");

System.out.print(p+"\t");

// System.out.print();

}

}