Hacker Rank Challenges

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Maximum Subarray Sum:**

import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.math.\*;

import java.util.regex.\*;

public class Solution {

static long maximumSum(long[] a, long m) {

int n = a.length;

long max = Integer.MIN\_VALUE;

for (int i = 0; i < n; i++){

long sum = 0;

for (int j = i; j < n; j++) {

sum = (sum+(a[j]%m))%m;

if (sum > max)

max = sum;

}

}

return max;

}

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

int q = in.nextInt();

for(int a0 = 0; a0 < q; a0++){

int n = in.nextInt();

long m = in.nextLong();

long[] a = new long[n];

for(int a\_i = 0; a\_i < n; a\_i++){

a[a\_i] = in.nextLong();

}

long result = maximumSum(a, m);

System.out.println(result);

}

in.close();

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Pairs:**

import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.math.\*;

import java.util.regex.\*;

public class Solution {

static int pairs(int[] a,int k) {

int matches = 0;

Arrays.sort(a);

int i = 0;

int j = 1;

while (j < a.length) {

int diff = a[j] - a[i];

if (diff == k) {

matches++;

j++;

} else if (diff > k) {

i++;

if (i == j) {

j++;

}

} else if (diff < k) {

j++;

}

}

return matches;

}

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

int res;

String n = in.nextLine();

String[] n\_split = n.split(" ");

int \_a\_size = Integer.parseInt(n\_split[0]);

int \_k = Integer.parseInt(n\_split[1]);

int[] \_a = new int[\_a\_size];

int \_a\_item;

String next = in.nextLine();

String[] next\_split = next.split(" ");

for(int \_a\_i = 0; \_a\_i < \_a\_size; \_a\_i++) {

\_a\_item = Integer.parseInt(next\_split[\_a\_i]);

\_a[\_a\_i] = \_a\_item;

}

res = pairs(\_a,\_k);

System.out.println(res);

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Missing Numbers:**

import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.math.\*;

import java.util.regex.\*;

public class Solution {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int aSize = scan.nextInt();

int[] a = new int[aSize];

for(int i = 0; i < aSize; i++) {

a[i] = scan.nextInt();

}

int bSize = scan.nextInt();

int[] b = new int[bSize];

for(int i = 0; i < bSize; i++) {

b[i] = scan.nextInt();

}

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

for(Integer aValue : a){

if(aMap.containsKey(aValue)){

aMap.put(aValue,aMap.get(aValue) + 1);

} else {

aMap.put(aValue, 1);

}

}

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

for(Integer bValue : b){

if(bMap.containsKey(bValue)){

bMap.put(bValue,bMap.get(bValue) + 1);

} else {

bMap.put(bValue, 1);

}

}

List<Integer> finalList = new ArrayList<Integer>();

if(aMap != null && bMap != null){

for(Map.Entry<Integer, Integer> entry : bMap.entrySet()){

if(!bMap.get(entry.getKey()).equals(aMap.get(entry.getKey()))){

finalList.add(entry.getKey());

}

}

}

int[] finalArray = null;

if(finalList != null){

finalArray = new int[finalList.size()];

for (int i=0; i < finalList.size(); i++){

finalArray[i] = finalList.get(i).intValue();

}

}

Arrays.sort(finalArray);

StringBuilder builder = new StringBuilder();

for (Integer value : finalArray) {

builder.append(value);

builder.append(" ");

}

System.out.println(builder.toString());

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**BigSorting:**

**import java.io.\*;**

**import java.util.\*;**

**import java.text.\*;**

**import java.math.\*;**

**import java.util.regex.\*;**

**public class Solution {**

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

**Scanner in = new Scanner(System.in);**

**int n = in.nextInt();**

**String[] unsorted = new String[n];**

**for(int unsorted\_i=0; unsorted\_i < n; unsorted\_i++){**

**unsorted[unsorted\_i] = in.next();**

**}**

**BigInteger[] a=new BigInteger[unsorted.length];**

**for(int i=0;i<n;i++){**

**a[i] = new BigInteger(unsorted[i]);**

**}**

**Arrays.sort(a);**

**for(int i=0; i < n; i++){**

**System.out.println(a[i]);**

**}**

**}**

**}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Into to Tutorials Challenge :**

**import java.io.\*;**

**import java.util.\*;**

**import java.text.\*;**

**import java.math.\*;**

**import java.util.regex.\*;**

**public class Solution {**

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

**Scanner scan = new Scanner(System.in);**

**int search = scan.nextInt();**

**scan.nextInt();**

**int i = 0;**

**while(scan.nextInt() != search){**

**i++;**

**}**

**System.out.println(i);**

**}**

**}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**The Full Counting Sort:**

**import java.io.\*;**

**import java.util.\*;**

**import java.text.\*;**

**import java.math.\*;**

**import java.util.regex.\*;**

**public class Solution {**

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

**Scanner scan=new Scanner(System.in);**

**int size=Integer.parseInt(scan.nextLine());**

**StringBuffer[] st=new StringBuffer[100];**

**for(int i=0;i<100;i++){**

**st[i]=new StringBuffer();**

**}**

**for(int i=0;i<size;i++) {**

**String sts=scan.nextLine();**

**String[] str=sts.split("[\\s]+");**

**int k=Integer.parseInt(str[0]);**

**String s;**

**if(i<size/2)**

**s="- ";**

**else**

**s=str[1]+" ";**

**st[k]=st[k].append(s);**

**}**

**for(int i=0;i<100;i++){**

**System.out.print(st[i]);**

**}**

**}**

**}**