**算法训练 数的统计**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　在一个有限的正整数序列中，有些数会多次重复出现在这个序列中。  
　　如序列：3，1，2，1，5，1，2。其中1就出现3次，2出现2次，3出现1 次，5出现1次。  
　　你的任务是对于给定的正整数序列，从小到大依次输出序列中出现的数及出现的次数。

输入格式

　　第一行正整数n，表示给定序列中正整数的个数。  
　　第二行是n 个用空格隔开的正整数x，代表给定的序列。

输出格式

　　若干行，每行两个用一个空格隔开的数，第一个是数列中出现的数，第二个是该数在序列中出现的次数。

样例输入

12  
8 2 8 2 2 11 1 1 8 1 13 13

样例输出

1 3  
2 3  
8 3  
11 1  
13 2

数据规模和约定

　　数据：n<=1000；0<x<=1000,000。

本题的C++参考代码如下：

#include <stdio.h>

#include <iostream>

#include <algorithm>

using namespace std;

int main()

{

int a[1000],i,j=0,b[1000]={0},n,sum=1,x=0;

scanf("%d",&n);

for(i=0;i<n;i++)scanf("%d",&a[i]);

sort(a,a+n);

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

{

if(a[i]==a[i+1])sum++;

else b[j++]=sum,sum=1;

}

for(i=0;i<j;i++)

{

x+=b[i];

printf("%d %d\n",a[x-1],b[i]);

}

return 0;

}

本题的C参考代码如下：

#include<stdio.h>

int main()

{

int i,m,n,a[1000],count=0,totalcount=0;

scanf("%d",&n);

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

scanf("%d",&a[i]);

for(m=1;totalcount<n;m++)

{

count=0;

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

if(a[i]==m)

{

count++;

totalcount++;

}

if(count!=0)

{

printf("%d %d",m,count);

printf("\n");

}

}

return 0;

}

本题的Java参考代码如下：

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.StreamTokenizer;

import java.util.ArrayList;

import java.util.Collections;

import java.util.LinkedHashMap;

import java.util.Map;

public class Main {

public static void main(String[] args) throws IOException {

StreamTokenizer st = new StreamTokenizer(new BufferedReader(

new InputStreamReader(System.in)));

st.nextToken();

int n = (int) st.nval;

Map<Integer, Integer> map = new LinkedHashMap<Integer, Integer>();

int m;

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

st.nextToken();

m = (int) st.nval;

if (map.containsKey(m))

map.put(m, map.get(m) + 1);

else

map.put(m, 1);

}

ArrayList<Integer> list = new ArrayList<Integer>();

list.addAll(map.keySet());

Collections.sort(list);

int key;

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

key = list.get(i);

System.out.println(key + " " + map.get(key));

}

}

}