

1239：统计数字

题目描述

某次科研调查时得到了 n 个自然数，每个数均不超过 1500000000 (1.5×10^9)。已知不相同的数不超过 10000 个，现在需要统计这些自然数各自出现的次数，并按照自然数从小到大的顺序输出统计结果。

输入

第一行是整数 n ，表示自然数的个数；

第 $2 \sim n+1$ 每行一个自然数。

输出

包含 m 行 (m 为 n 个自然数中不相同数的个数)，按照自然数从小到大的顺序输出。每行输出两个整数，分别是自然数和该数出现的次数，其间用一个空格隔开。

输入样例

```
8
2
4
2
4
5
100
2
100
```

输出样例

```
2 3
4 2
5 1
100 2
```

提示

数据范围：

40%的数据满足： $1 \leq n \leq 1000$ ；

80%的数据满足： $1 \leq n \leq 50000$ ；

100%的数据满足： $1 \leq n \leq 200000$ ，每个数均不超过 1500000000 (1.5×10^9)。

本题目禁止使用STL及包含可以使用的相关调用。

解析

思路很简单，将所有数据排序后依次比较前后相等即可。题目中又限定了不可以使用STL库，实际上就是要求大家来编写一个二分的排序算法。很明显，我们可以使用归并排序。

编码

```
#include <iostream>

using namespace std;

int n;
int nums[200001];

void merge(int arr[], int low, int mid, int high) {
    //low为第1有序区的第1个元素，i指向第1个元素，
    //mid为第1有序区的最后1个元素
    int i = low, j = mid + 1, k = 0;
    //mid+1为第2有序区第1个元素，j指向第1个元素
    //temp数组暂存合并的有序序列
    int *temp = new int[high - low + 1];
    while (i <= mid && j <= high) {
        if (arr[i] <= arr[j]) //较小的先存入temp中
            temp[k++] = arr[i++];
        else {
            temp[k++] = arr[j++];
        }
    }
    //若比较完之后，第一个有序区仍有剩余，则直接复制到t数组中
    while (i <= mid)
        temp[k++] = arr[i++];
    while (j <= high) //同上
        temp[k++] = arr[j++];
    //将排好序的存回arr中low到high这区间
    for (i = low, k = 0; i <= high; i++, k++)
        arr[i] = temp[k];
    //释放内存，由于指向的是数组，必须用delete []
    delete[] temp;
}

void MergeSort(int arr[], int low, int high) {
    // 终止递归的条件，子序列长度为1
    if (low >= high) {
        return;
    }
    int mid = (low + high) / 2;
    // 对左半边递归
    MergeSort(arr, low, mid);
    // 对右半边递归
    MergeSort(arr, mid + 1, high);
    merge(arr, low, mid, high); // 合并
}
```

```

void Print() {
    int num = 1; //当前数字出现的数目
    for (int i = 0; i < n; i++) {
        //如果后一个数字等于前一个数字，则进行记录。
        if (nums[i] == nums[i + 1]) {
            num++;
        } else {
            //出现了新的数字就打印，并将数据归并为1
            printf("%d %d\n", nums[i], num);
            num = 1;
        }
    }
}

int main(int argc, char **argv) {
    cin >> n;
    for (int i = 0; i < n; i++) {
        scanf("%d", &nums[i]);
    }
    MergeSort(nums, 0, n - 1);
    Print();
    return 0;
}

```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

