

第五章 数的存储与组织

第 6 1 课 set 与 map

《信息学奥赛一本通·编程启蒙 C++版》

set 的常用操作

函数	说明
q.insert(x)	将 x 插入 q 中
q.erase(x)	删除 q 中的 x 元素, 返回 0 或 1, 0 表示 set 中不存在 x
q.clear()	清空 q
q.empty()	判断 q 是否为空, 若是返回 1, 否则返回 0
q.size()	返回 q 中元素的个数
q.find(x)	在 q 中查找 x, 返回 x 的迭代器, 若 x 不存在, 则返回指向 q 尾部的迭代器即 q.end()
q.lower_bound(x)	返回一个迭代器, 指向第一个键值不小于 x 的元素
q.upper_bound(x)	返回一个迭代器, 指向第一个键值大于 x 的元素
q.rend()	返回第一个元素的的前一个元素迭代器
q.begin()	返回指向 q 中第一个元素的迭代器
q.end()	返回指向 q 最后一个元素下一个位置的迭代器
q.rbegin()	返回最后一个元素

map 的常用操作

函数	说明
<code>q.insert()</code> 或 <code>q[i]=x</code>	将 <code>x</code> 插入 <code>q</code> 中
<code>q.erase(pos)</code>	删除 <code>pos</code> 迭代器所指的元素
<code>q.clear()</code>	清空 <code>q</code>
<code>q.empty()</code>	判断 <code>q</code> 是否为空，若是返回 1，否则返回 0
<code>q.size()</code>	返回 <code>q</code> 中元素的个数
<code>q.find(key)</code>	在 <code>q</code> 中查找键 <code>key</code> 是否存在，若存在，返回该键对应元素的迭代器，若不存在，则返回 <code>map</code>
<code>q.lower_bound(keyELem)</code>	返回第一个 <code>key >= keyELem</code> 元素的迭代器
<code>q.upper_bound(keyELem)</code>	返回第一个 <code>key > keyELem</code> 元素的迭代器

【例 61.1】 机器翻译

【题目描述】

小晨的电脑上安装了一个机器翻译软件，他经常用这个软件来翻译英语文章。这个翻译软件的原理很简单，它只是从头到尾，依次将每个英文单词用对应的中文含义来替换。对于每个英文单词，软件会先在内存中查找这个单词的中文含义，如果内存中有，软件就会用它进行翻译；如果内存中没有，软件就会在外存中的词典内查找，查出单词的中文含义然后翻译，并将这个单词和译义放入内存，以备后续的查找和翻译。

假设内存中有 M 个单元，每单元能存放一个单词和译义。每当软件将一个新单词存入内存前，如果当前内存中已存入的单词数不超过 $M-1$ ，软件会将新单词存入一个未使用的内存单元；若内存中已存入 M 个单词，软件会清空最早进入内存的那个单词，腾出单元来，存放新单词。

假设一篇英语文章的长度为 N 个单词。给定这篇待译文章，翻译软件需要去外存查找多少次词典？假设在翻译开始前，内存中没有任何单词。

【输入格式】

共 2 行。每行中两个数之间用一个空格隔开。

第一行为两个正整数 M 和 N ，代表内存容量和文章的长度。

第二行为 N 个非负整数，按照文章的顺序，每个数（大小不超过 1000）代表一个英文单词。文章中两个单词是同一个单词，当且仅当它们对应的非负整数相同。

【输出格式】

共 1 行，包含一个整数，为软件需要查词典的次数。

【样例输入】

```
3 7
1 2 1 5 4 4 1
```

【样例输出】

```
5
```

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int m,n,x,cnt,p=-1;
4. set<int> st;
5. int a[100000];
6. int main(){
7.     cin>>m>>n;
8.     for(int i=1;i<=n;i++){
9.         cin>>x;
10.        if(st.find(x)==st.end()){
11.            cnt++;
12.            st.insert(x);
13.            if(st.size()<=m){
14.                p++;
15.                p=p%m;
16.                a[p]=x;
17.            }else{
18.                p++;
19.                p=p%m;
20.                st.erase(a[p]);
21.                a[p]=x;
22.            }
23.        }
24.    }
25.    cout<<cnt;
26.    return 0;
27.}
```

【例 61.2】 最近的一对

【题目描述】

给出包含 n 个元素的数组 a ，求 a 中距离最近的一对 i, j ，满足 $i < j$ 且 $a[i] = a[j]$ 。如果同时存在多对，输出最小的 i 对应的 $a[i]$ 。例如：10 个数 19, 13, 11, 19, 11, 5, 6, 3, 4, 3 满足存在 $a[i] = a[j]$ 的数字包括：19, 11, 3。其中 11, 3 这两对的距离更近，在距离相同的情况下，11 的下标更靠前。如果不存在相同的数字，输出 "No"。

【输入格式】

第一行：1 个数 n 表示数组的长度 ($2 \leq n \leq 100000$)。第 2 至 $n+1$ 行：每行 1 个数，对应数组的元素 ($1 \leq a[i] \leq 10^9$)。

【输出格式】

输出符合条件的 i 最小的 $a[i]$ 。

【样例输入】

10

19

13

11

19

11

5

6

3

4

3

【样例输出】

11

【代码实现】

```
1. #include<bits/stdc++.h>
2. #define INF 0x7fffffff
3. using namespace std;
4. map<int,int> mp;
5. int n,x,ans,minx=INF;
6. int main(){
7.     cin>>n;
8.     for(int i=1;i<=n;i++){
9.         cin>>x;
10.        if(mp[x]==0) mp[x]=i;
11.        else{
12.            if(i-mp[x]<minx){
13.                minx=i-mp[x];
14.                ans=x;
15.            }
16.            mp[x]=i;
17.        }
18.    }
19.    if(ans==0) cout<<"No"<<endl;
20.    else cout<<ans<<endl;
21.    return 0;
22.}
```

【例 61.3】 图书管理员

【题目描述】

图书馆中每本书都有一个图书编码，可以用于快速检索图书，这个图书编码是一个正整数。

每位借书的读者手中有一个需求码，这个需求码也是一个正整数。如果一本书的图书编码恰好以读者的需求码结尾，那么这本书就是这位读者所需要的。

小 D 刚刚当上图书馆的管理员，她知道图书馆里所有书的图书编码，她请你帮她写一个程序，对于每一位读者，求出他所需要的书中图书编码最小的那本书，如果没有他需要的书，请输出 -1。

【输入格式】

输入的第一行，包含两个正整数 n 和 q ，以一个空格分开，分别代表图书馆里书的数量和读者的数量。

接下来的 n 行，每行包含一个正整数，代表图书馆里某本书的图书编码。

接下来的 q 行，每行包含两个正整数，以一个空格分开，第一个正整数代表图书馆里读者的需求码的长度，第二个正整数代表读者的需求码。

$1 < n < 1000, 1 < q < 1000$ ，所有的图书编码和需求码均不超过 10000000。

【输出格式】

输出有 q 行，每行包含一个整数，如果存在第 i 个读者所需要的书，则在第 i 行输出第 i 个读者所需要的书中图书编码最小的那本书的图书编码，否则输出 -1。

【样例输入】

```
5 5
2123
1123
23
24
24
2 23
3 123
3 124
2 12
2 12
```

【样例输出】

```
23
1123
-1
-1
-1
```

【代码实现】

```
1. #include <bits/stdc++.h>
2. using namespace std;
3. map<int,int> mp;
4. int main() {
5.     int n,q=0,x,len,
6.     p[9]={1,10,100,1000,10000,100000,1000000,10000000,100000000};
7.     cin>>n>>q;
8.     while(n--){
9.         cin>>x;
10.        for(int i=1;i<=8;i++){
11.            if(mp[x%p[i]]==0 || x<mp[x%p[i]]){
12.                mp[x%p[i]]=x;
13.            }
14.        }
15.    }
16.    while(q--){
17.        cin>>len>>x;
18.        if(mp[x]==0) cout<<-1<<endl;
19.        else cout<<mp[x]<<endl;
20.    }
21.    return 0;
22.}
```

练 61.1 统计字符数

【题目描述】

给定一个由 **a-z** 这 **26** 个字符组成的字符串，统计其中哪个字符出现的次数最多？

【输入格式】

输入包含一行，一个字符串，长度不超过 **1000**。

【输出格式】

输出一行，包括出现次数最多的字符和该字符出现的次数，中间以一个空格分开。如果有多个字符出现的次数相同且最多，那么输出 **ascii** 码最小的那一个字符。

【样例输入】

abbccc

【样例输出】

c 3

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. char c;
4. int a[30],maxx,x;
5. int main(){
6.     while(cin>>c) a[c-'a']++;
7.
8.     for(int i=0;i<=25;i++){
9.         if(a[i]>maxx){
10.             maxx=a[i];
11.             x=i;
12.         }
13.     }
14.     cout<<char('a'+x)<<' '<<maxx;
15.     return 0;
16.}
```

练 61.2 掷骰子

【题目描述】

贝西喜欢玩桌上扮演游戏 (**BRPG**)，所以她说服农场主约翰开车送她到商店，在那里她买了三个骰子。

这三个骰子分别有 S_1, S_2, S_3 个面。每个有 S 面的骰子每一面的点数分别是 $1, 2, 3, \dots, S-1, S$ 。

贝西每次同时扔出三个骰子，她一直扔呀扔，试图找出三个骰子点数之和哪个出现的次数最多。

现在给定三个骰子中每个骰子的面数，求三个骰子的点数和哪个最频繁出现如果有多解，输出最小的一个。

数据范围：($2 \leq S_1 \leq 20$; $2 \leq S_2 \leq 20$; $2 \leq S_3 \leq 40$)。

【输入格式】

第一行：三个正整数 S_1, S_2, S_3 。

【输出格式】

一个整数表示最频繁出现的点数和。

【样例输入】

3 2 3

【样例输出】

5

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int x,y,z,t[105],maxx=3;
4. int main(){
5.     cin>>x>>y>>z;
6.     for(int i=1;i<=x;i++){
7.         for(int j=1;j<=y;j++){
8.             for(int k=1;k<=z;k++){
9.                 t[i+j+k]++;
10.            }
11.        }
12.    }
13.    maxx=*max_element(t,t+85);
14.    for(int i=1;i<=85;i++){
15.        if(t[i]==maxx){
16.            cout<<i<<endl;
17.            return 0;
18.        }
19.    }
20.}
```

练 61.3 开花

【题目描述】

学校又迎来了一年一度的开花活动，有 n 名学生被评为文学优秀奖， m 名学生被评为体育优秀奖。现已知两个奖项获奖同学的编号，每个同学都有唯一的编号。只有同时被评为文学优秀奖和体育优秀奖的学生才能开花，请你帮忙统计一下开花的名单。

【输入格式】

第一行两个整数 n, m ($1 \leq n, m \leq 10^5$)，分别表示文学优秀奖和体育优秀奖的获奖人数。

第二行 n 个不同的整数，表示获得文学优秀奖的同学编号。

第二行 m 个不同的整数，表示获得体育优秀奖的同学编号。

所有编号为正整数且不超过 10^9 。

【输出格式】

一行若干个空格分隔的整数，表示开花的同学编号，按文学优秀奖的先后次序输出。

【样例输入】

```
4 4
5 1 7 3
2 3 4 1
```

【样例输出】

```
1 3
```

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. map<int,bool> vis;
4. int n,m,x,a[100005];
5. int main(){
6.     cin>>n>>m;
7.     for(int i=1;i<=n;i++) cin>>a[i];
8.     for(int i=1;i<=m;i++){
9.         cin>>x;
10.        vis[x]=1;
11.    }
12.    for(int i=1;i<=n;i++){
13.        if(vis[a[i]]) cout<<a[i]<<' ';
14.    }
15.    return 0;
16.}
```

谢谢！

