

太戈编程
etiger.vip

信奥算法

太戈编程308

流浪狗

流浪动物管理所登记册上共有 n 只流浪狗的信息，这些是有名牌的狗，名牌上写有名字和出生年份。如果两只狗的姓名和出生年份一模一样，被认为是对同一只狗重复登记了。现在有 m 个主人来查询是否有自己家的狗被找到。

输入第一行为正整数 n 和 m ，代表有 n 条流浪狗信息，和 m 个查询。以下 n 行每行为狗的名字和出生年份。最后 m 行每行为一个查询，为狗的出生年份和姓名。 $1 \leq n, m \leq 10000$ 。狗的名字都是小写字母。
输出一行 m 个数字， 1 表示找到， 0 表示没找到。

输入样例

4 2

wangcai 2016

emma 2010

mike 2015

harrypotter 2011

2010 harrypotter

2010 emma

输出样例

0 1

暴力算法

登记册上共有 n 只流浪狗的信息

$\text{year}[i]$ 代表 i 号狗的年份

$\text{name}[i]$ 代表 i 号狗的名字

有 m 个主人来查询

对于每个查询
都枚举 n 只登记册上信息
进行核对

时间复杂度 $O(nm)$

暴力代码

```
8 cin>>n>>m;
9 for(int i=0;i<n;i++)cin>>name[i]>>year[i];
10 for(int i=0;i<m;i++){
11     int x;
12     string y;
13     cin>>x>>y;
14     int OK=0;
15     for(int j=0;j<n;j++)
16         if(name[j]==y&&[ ]){
17             OK=1;
18             break;
19         }
20     cout<<OK<<" ";
21 }
```

OK代表是否找到

若找到就跳出循环

时间复杂度 $O(nm)$

能否再加速?

自定义类型



数据容器

自定义类型



数据容器

struct

set
multiset

自定义类型

识别对象	狗
------	---

识别对象 的属性	名字 年份
-------------	----------

```
struct dog{  
    string name;  
    int year;  
};
```

如何定义 包含dog类型的set

错误程序

错在哪里

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 struct dog{
4     string name;
5     int year;
6 };
7 int main() {
8     set<dog> s;
9     dog a,b;
10    a.name="mike"; a.year=2020;
11    b.name="wangcai"; b.year=2019;
12    s.insert(a);
13    s.insert(b);
14    return 0;
15 }
```

定义包含dog类型的set容器

把dog信息插入set

报错信息

308流浪狗重载错误程序.cpp

stl_function.h

```
361 operator()(const _Tp& __x, const _Tp& __y) const
362 { return __x > __y; }
363 };
```

```
365 /// One of the @Link comparison_functors comparison functors@endLink.
```

```
366 template<typename _Tp>
367 struct less : public binary_function<_Tp, _Tp, bool>
368 {
369     bool
370     operator()(const _Tp& __x, const _Tp& __y) const
371     { return __x < __y; }
372 };
```

C++软件提醒：
set里需要比较
两个对象的大小

但你的程序
没有提供
比较规则

正确写法

```
3 struct dog{
4     string name;
5     int year;
6     bool operator<(const dog&d)const{
7         if(year<d.year)return 1;
8         if(year>d.year)return 0;
9         if(name<d.name)return 1;
10        return 0;
11    }
12};
```

重载小于号<

set内部需要排序
必须知道任意
2只狗的前后排序规则

重新自定义
1只狗是否"小于"
另1只狗

正确写法

```
3 struct dog{  
4     string name;  
5     int year;  
6     bool operator<(const dog&d)const{  
7         if(year<d.year)return 1;  
8         if(year>d.year)return 0;  
9         if(name<d.name)return 1;  
10        return 0;  
11    }  
12 };
```

比较规则
隶属于dog类型内部

第7,8,9,10行
对应比较规则

我这只狗
和另一只狗d
的比较规则

正确写法

```
3 struct dog{  
4     string name;  
5     int year;  
6     bool operator<(const dog&d) const{  
7         if(year<d.year) return 1;  
8         if(year>d.year) return 0;  
9         if(name<d.name) return 1;  
10        return 0;  
11    }  
12 };
```

第7,8,9,10行
对应比较规则

若我的年份小于d的年份
那么我排在d前面

若我的年份大于d的年份
那么我不排在d前面

正确写法

```
3 struct dog{  
4     string name;  
5     int year;  
6     bool operator<(const dog&d) const{  
7         if(year<d.year) return 1;  
8         if(year>d.year) return 0;  
9         if(name<d.name) return 1;  
10        return 0;  
11    }  
12 };
```

第7,8,9,10行
对应比较规则

若年份相同,再比名字

若我的名字字典序小于d的名字
那么我排在d前面

否则,我不排在d前面

正确写法

```
3 struct dog{  
4     string name;  
5     int year;  
6     bool operator<(const dog&d) const{  
7         if(year<d.year) return 1;  
8         if(year>d.year) return 0;  
9         if(name<d.name) return 1;  
10        return 0;  
11    }  
12};
```

用**bool**类型

返回结果是或非,1或0

判断

我这只狗是否排在另一只狗d前面

正确写法

```
3 struct dog{  
4     string name;  
5     int year;  
6     bool operator<(const dog&d)const{  
7         if(year<d.year)return 1;  
8         if(year>d.year)return 0;  
9         if(name<d.name)return 1;  
10        return 0;  
11    }  
12};
```

用**operator<**

小于号操作符的函数名

operator中文含义
是操作符

正确写法

```
3 struct dog{  
4     string name;  
5     int year;  
6     bool operator<(const dog&d)const{  
7         if(year<d.year)return 1;  
8         if(year>d.year)return 0;  
9         if(name<d.name)return 1;  
10        return 0;  
11    }  
12};
```

用**const**

英文constant
不变量的缩写

保证比较时不修改内容

正确写法

```
3 struct dog{  
4     string name;  
5     int year;  
6     bool operator<(const dog&d)const{  
7         if(year<d.year)return 1;  
8         if(year>d.year)return 0;  
9         if(name<d.name)return 1;  
10        return 0;  
11    }  
12};
```

用&

参数传递时只传递地址

易错点汇总

有哪些错?

```
3 struct dog{  
4     string name;  
5     int year;  
6 };  
7 bool operator<(dog d){  
8     if(year<d.year)return 1;  
9     if(year>d.year)return 0;  
10 }
```

预测输出结果

```
3 struct dog{  
4     string name;  
5     int year;  
6     bool operator<(const dog&d)const{  
7         if(year<d.year)return 1;  
8         if(year>d.year)return 0;  
9         if(name<d.name)return 1;  
10        return 0;  
11    }  
12 };
```

```
14 set<dog> s;  
15 dog a,b;  
16 a.name="mike";    a.year=2020;  
17 b.name="wangcai"; b.year=2019;  
18 s.insert(a);    s.insert(b);  
19 dog x=*(s.begin());  
20 cout<<x.name<<endl;
```

流浪狗

流浪动物管理所登记册上共有 n 只流浪狗的信息，这些是有名牌的狗，名牌上写有名字和出生年份。如果两只狗的姓名和出生年份一模一样，被认为是对同一只狗重复登记了。现在有 m 个主人来查询是否有自己家的狗被找到。

```
3 struct dog{
4     string name;
5     int year;
6     bool operator<(const dog&d) const{
7         if(year<d.year) return 1;
8         if(year>d.year) return 0;
9         if(name<d.name) return 1;
10        return 0;
11    }
12};
```

利用set自带功能
判断存在性

流浪狗

```
14 set<dog> s;  
15 dog d;  
16 int n,m;  
17 cin>>n>>m;  
18 for(int i=0;i<n;i++){  
19     cin>>d.name>>d.year;  
20     s.insert(d);  
21 }
```

定义包含狗的set容器

把狗d的信息插入set

```
22 for(int i=0;i<m;i++){  
23     cin>>d.year>>d.name;  
24     if(s.count(d)) cout<<"1 ";  
25     else cout<<"0 ";  
26 }
```

在set里查询
狗d是否存在

高考排名

你是校长，学校共 n 名学生参加高考，学号从1到 n 编号。每个学生有一个总分，你需要将学生成绩从高到低排序。如果分数相同，学号小的排的前面。

输入第一行为正整数 n ， $n \leq 100000$ 。第二行共 n 个正整数，依次代表 n 个人的总分：第1个分数对应学号为1的学生，第2个分数对应学号为2的学生，以此类推，最后一个分数对应学号为 n 的学生。分数均不超过1000。

输出共 n 行，每行两个数字，由空格隔开，第一个数为学号，第二个数字为该学生的总分。注意行末不能有空格。

输入样例

```
4
59 60 60 100
```

输出样例

```
4 100
2 60
3 60
1 59
```

高考排名

你是校长，学校共 n 名学生参加高考，学号从1到 n 编号。每个学生有一个总分，你需要将学生成绩从高到低排序。如果分数相同，学号小的排的前面。

识别对象	学生
------	----

识别对象的属性	总分
	学号

讨论：需要 multiset 吗
不需要 因为学号不同

高考排名

你是校长，学校共 n 名学生参加高考，学号从1到 n 编号。每个学生有一个总分，你需要将学生成绩从高到低排序。如果分数相同，学号小的排的前面。

识别对象	学生
------	----

识别对象的属性	总分
	学号

此题需要
排序操作

利用set
自动排序

自定义类型

重载小于号< 自定义比较规则

```
struct student{  
    int id,score;  
    bool operator<(const student&b)const{  
        if(score>b.score)return 1;  
        if(score<b.score)return 0;  
        if(id<b.id)return 1;  
        return 0;  
    }  
};
```

score代表总分

id代表学号

利用set 自动排序功能

```
14  int n;  
15  cin>>n;  
16  set<student> s;  
17  student x;  
18  for(int i=0;i<n;i++){  
19      cin>>x.score;  
20      x.id=i+1;  
21      s.insert(x);  
22  }  
  
23  set<student>::iterator it;  
24  for(it=s.begin();it!=s.end();it++)  
25      cout<<(*it).id<<" "<<(*it).score<<endl;
```

其他方法

此类排序问题解法很多

都需要自定义类型student

1

set<>数据容器
struct内重载小于号

自动排序

2

数组sort()
用比较函数cmp()

sort(f, f+n, cmp);

3

数组sort()
struct内重载小于号

sort(f, f+n);

太戈编程309

请同学写下题目大意
已知什么求什么

限时3分钟

请同学核对题目大意
不能遗漏核心要点

309

n 人依次到救济站等待发大米，每人大米初始数量给定。共发 n 次大米：每次大米最少的人领取3袋大米，如有两人大米相同则先来救济站的人先领取。领完大米的人可继续领取，请问最终有多少人没有领过大米？

识别对象	人
------	---

识别对象的属性	当前大米数量
	第几个到现场

比较规则	先比米谁更少
	再比谁来的更早

如何识别大米发给谁	利用set找最值功能	动态最值	修改操作
-----------	------------	------	------

```
3 struct ren{
4     int x,id;
5     bool operator<(const ren&b)const{
6         if(x<b.x)return 1;
7         if(x>b.x)return 0;
8         if(id<b.id)return 1;
9         return 0;
10    }
11 };
```

x代表此人当前大米数量

id代表此人第几个到现场

请理解每一行含义

```
13  int n;  
14  cin>>n;  
15  set<ren> s;  
16  set<ren>::iterator it;  
17  ren d;  
18  for(int i=1;i<=n;i++){  
19      cin>>d.x;  
20      d.id=i;  
21      s.insert(d);  
22  }
```

容器s里存放
所有人的信息

请理解每一行含义

```
23 set<int> ok;  
24 for(int i=1;i<=n;i++){  
25     it=s.begin();  
26     d.x=(*it).x+3;  
27     d.id=(*it).id;  
28     s.erase(it);  
29     s.insert(d);  
30     ok.insert(d.id);  
31 }  
32 cout<<n-ok.size()<<endl;
```

提取信息

删除信息

插入信息

容器ok里存放
领过米的人的编号

修改信息

给某个人发米

本质是修改此人
当前大米数量

修改操作
分3步

- 1 提取信息
- 2 删除信息
- 3 插入信息

太戈编程

308

1497

184

拓展题

309

本次作业需要使用
数据容器实现程序