



# 信奥算法

# 数据容器

map

一对一映射

# map映射

每个人名都对应一个身高

每个string对应一个double

每个阿拉伯数字都对应一个英文拼写

每个int对应一个string

每个学生姓名都对应一个考试分数

每个string对应一个int

# map 定义

```
#include<map>
```

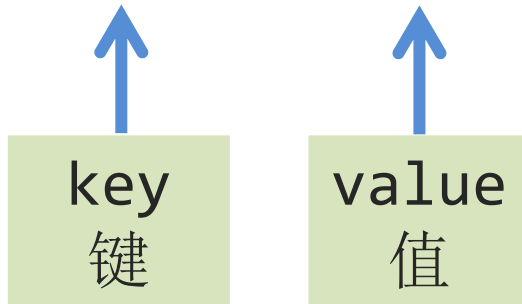
引入map库

```
map < string, int > d;
```

每个string对应一个int

```
map < char, string > d;
```

每个char对应一个string



# 中括号操作符[]

```
1  #include<iostream>
2  #include<map>
3  #include<string>
4  using namespace std;
5  int main(){
6      map<char,string> d;
7      d['a']="wawawa";
8      d['b']="hahaha";
9      d['c']=d['a'];
10     d['b']="hohoho";
11     cout<<d['a'] <<endl;
12     cout<<d['b'] <<endl;
13     cout<<d['c'] <<endl;
14     return 0;
15 }
```

类似数组的  
[]操作符  
用于访问元素

请预测输出结果

# .size()

```
1 #include<iostream>
2 #include<map>
3 using namespace std;
4 int main(){
5     map<char,int> d;
6     d['a']=101;
7     d['b']=202;
8     d['c']=302;
9     cout<<d.size()<<endl;
10    return 0;
11 }
```

请预测输出结果

# .count()

```
1 #include<iostream>
2 #include<map>
3 using namespace std;
4 int main (){
5     map<char,int> d;
6     d['a']=50;
7     d['b']=100;
8     cout<<d.count('a')<<endl;
9     cout<<d.count('b')<<endl;
10    cout<<d.count('x')<<endl;
11    return 0;
12 }
```

请预测输出结果



# 计数器map

```
1 #include<iostream>
2 #include<map>
3 #include<string>
4 using namespace std;
5 int main(){
6     map<string,int> cnt;
7     cnt["Mike"]++;
8     cnt["John"]++;
9     cnt["Mike"]++;
10    cnt["Peter"]++;
11    cout<<cnt["Mike"]<<endl;
12    cout<<cnt["John"]<<endl;
13    cout<<cnt["Peter"]<<endl;
14    return 0;
15 }
```

类似数组的  
[]操作符  
用于访问元素

第一次访问时  
自动新建并赋值  
(为默认值0)

请预测输出结果



# 迭代器 .begin()

对于key  
自动排序

```
6 map<string,int> cnt;  
7 cnt["Mike"]=1;  
8 cnt["John"]=2;  
9 cnt["Peter"]=3;  
10 map<string,int>::iterator mit;  
11 mit=cnt.begin();  
12 cout<<(*mit).first<<" ";  
13 cout<<(*mit).second<<endl;
```

请预测输出结果

# 迭代器 .begin()

```
6 map<string,int> cnt;  
7 cnt["Mike"]=1;  
8 cnt["John"]=2;  
9 cnt["Peter"]=3;  
10 map<string,int>::iterator mit;  
11 mit=cnt.begin();  
12 cout<<(*mit).first<<" ";  
13 cout<<(*mit).second<<endl;  
14 cout<<mit->first<<" ";  
15 cout<<mit->second<<endl;
```

请预测输出结果

# 头尾位置

迭代器  
iterator

d.begin()

d.end()

<0,"c">

<1,"a">

<2,"b">

<3,"c">

d.end()  
指向容器d外

# map 遍历

对于key  
自动排序  
自动去重

```
5 map<char,int> d;  
6 d['b'] = 100;  
7 d['a'] = 200;  
8 d['c'] = 300;  
9 map<char,int>::iterator it;  
10 for(it=d.begin();it!=d.end();++it)  
11     cout<<it->first<<" "<<it->second<<endl;  
12 for(it=d.begin();it!=d.end();++it)  
13     cout<<(*it).first<<" "<<(*it).second<<endl;
```

请预测输出结果

# map 易错点

对于key  
自动排序自  
动去重

value附属于key  
不会自动排序  
不会自动去重

```
map < char, string > d;
```

key  
键

value  
值

# .find()

```
1 #include<iostream>
2 #include<map>
3 using namespace std;
4 int main(){
5     map<char,int> d;
6     map<char,int>::iterator it;
7     d['a']=50;
8     d['b']=100;
9     d['c']=150;
10    it=d.find('c');
11    cout<<it->second<<endl;
12    return 0;
13 }
```

请预测输出结果

# .erase()

```
5 map<char,int> d;  
6 map<char,int>::iterator it;  
7 d['a']=50;  
8 d['b']=100;  
9 d['c']=150;  
10 it=d.find('b');  
11 if(it!=d.end()) ← 如果找到了  
12     d.erase(it);  
13 for(it=d.begin();it!=d.end();++it)  
14     cout<<it->second<<endl;
```

请预测输出结果



# 存在性判断

判断特定key键x是否出现过?

1

当s.count(x)返回0时



说明s里没有x键

当s.count(x)返回非0时



说明s里存在x键

2

当s.find(x)==s.end()时



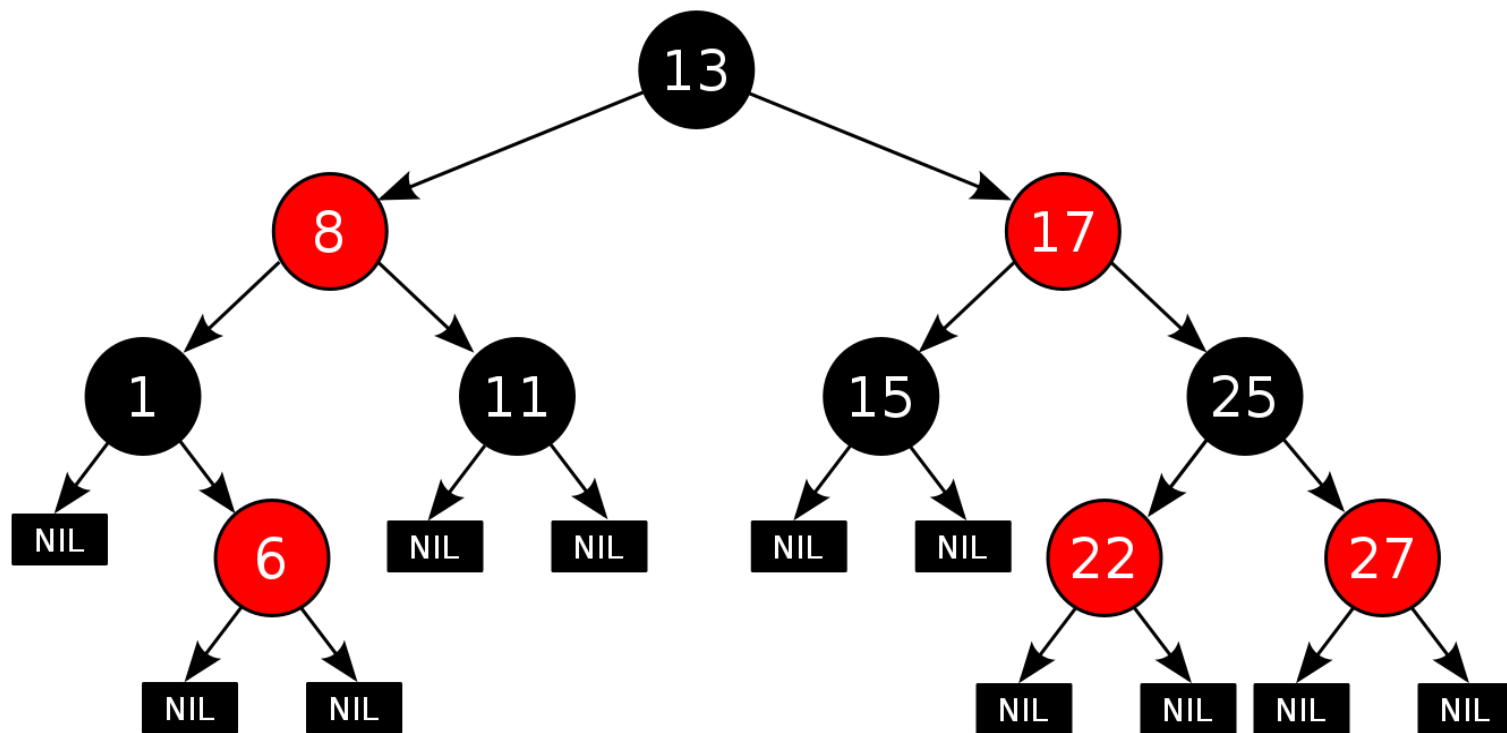
说明s里没有x键

当s.find(x)!=s.end()时



说明s里存在x键

# 红黑树RB-tree



map的底层实现都是红黑树

二叉树

对key排序

子树高度平衡

# map综合问题

# 字符统计

输入一行字符串，输出其中每个字符出现的总次数。按照ASCII码从小到大排列。

输入样例  
abcaaacc

输出样例

a 4  
b 1  
c 3

输入样例  
abcaaacc

输出样例

a 4  
b 1  
c 3

输入样例  
x=y=z;

输出样例

; 1  
= 2  
x 1  
y 1  
z 1

核心问题

映射：什么信息对应什么信息

记笔记

```
map<char, int> d;
```

定义映射d:每个字母符号  
对应1个当前出现次数

快快编程  
kkcoding.net

# 字符统计

```
1  #include<iostream>
2  #include<string>
3  #include<map>
4  using namespace std;
5  int main() {
6      map<char,int> d;
7      map<char,int>::iterator mit;
8      string s;
9      getline(cin,s);
10     for(int i=0;i<s.size();i++)
11         d[s[i]]++;
12     for(mit=d.begin();mit!=d.end();mit++)
13         cout<<mit->first<<" "<<mit->second<<endl;
14     return 0;
15 }
```

# 现场挑战

## 快快编程323

快快编程  
kkcoding.net



核心问题

映射：什么信息对应什么信息

记笔记

```
map<string,int> d;
```

定义映射d:每个姓名字符串  
对应1个当前最高分整数

快快编程  
kkcoding.net

```
6 map<string,int> d;  
7 map<string,int>::iterator mit;  
8 int n;  
9 cin>>n;  
10 for(int i=0;i<n;i++){  
11     string name;  
12     int score;  
13     cin>>name>>score;  
14     if(d.count(name)==0 || )  
15         d[name]=score;  
16 }
```

# 现场挑战

## 快快编程324

快快编程  
kkcoding.net

核心问题

映射：什么信息对应什么信息

记笔记

```
map<string,int> d;
```

定义映射d:每个姓名字符串  
对应1个当前资金变化量整数

快快编程  
kkcoding.net

```
6 map<string,int> d;
7 int n;
8 cin>>n;
9 for(int i=0;i<n;i++){
10     int p,m;
11     char ch;
12     string giver,taker;
13     cin>>giver>>p>>ch>>m;
14     d[giver]-=p*m;
15     for(int j=0;j<m;j++){
16         cin>>taker;
17         
18     }
19 }
```

请写出变量含义

快快编程  
kkcoding.net

输出第一行为所有参与者的个数，  
以下若干行每行为按照姓名字典序  
输出所有参与者的姓名和最终的资金变化。

```
20 cout<<d.size()<<endl;
21 map<string,int>::iterator mit;
22 for(mit=d.begin();mit!=d.end();mit++){
23     cout<<mit->first<<" ";
24     cout<<[REDACTED]<<endl;
25 }
```

# 现场挑战

## 快快编程325

快快编程  
kkcoding.net



核心问题

映射：什么信息对应什么信息

```
map<char, char> code;
```

记笔记

定义映射code: 每个被加密字母符号  
对应1个加密后的字母符号

快快编程  
kkcoding.net

```
6      string a,b;
7      getline(cin,a);
8      getline(cin,b);
9      map<char,char> code;
10     int len=b.size();
11     code[b[len-1]]=b[0];
12     for(int i=0;i<=  ;i++)
13         code[b[i]]=b[i+1];
```

```
14 for(int i=0;i<a.size();i++)
15     if(code.count(a[i]))
16         cout<< 
17     else
18         cout<< 
```

# 参考资料

<http://www.cplusplus.com/reference/map/map/>

# 快快编程作业

323

324

325

拓展题

348, 403, 349, 332