

# CS101

A Mars rover, likely a Curiosity rover, is shown on a rocky, reddish-brown landscape. The rover is white with various instruments and cameras. It has six large, treaded wheels. The background shows a hazy, orange sky and distant hills. The overall scene is a typical Mars surface environment.

信奥  
算法

课件下载地址:

<http://pan.baidu.com/s/1o885tz0>

作业网站:

<http://120.132.18.213:8080/thrall-web/main#home>

# 作业1 作业提交

```
1  #include<iostream>
2  #include<string>
3  #include<map>
4  using namespace std;
5  int main() {
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 || score>d[name])
15             d[name]=score;
16     }
17     cout<<d.size()<<endl;
18     for(mit=d.begin();mit!=d.end();mit++)
19         cout<<mit->first<<" "<<mit->second<<endl;
20     return 0;
21 }
```

# 作业2 购物节

```
1  #include<iostream>
2  #include<map>
3  #include<string>
4  #include<iomanip>
5  using namespace std;
6  int main(){
7      map<string,double> d;
8      map<string,double>::iterator mit;
9      int n;
10     cin>>n;
11     for(int i=0;i<n;i++) {
12         string item;
13         double value;
14         cin>>item>>value;
15         if(d.count(item)==0)
16             d[item]=value*0.75;
17         else
18             d[item]+=value;
19     }
20     for(mit=d.begin();mit!=d.end();mit++)
21         cout<<mit->first<<" "<<fixed<<setprecision(2)<<mit->second<<endl;
22     return 0;
23 }
```

# 作业3 发红包

```
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     int p,m;
12     char ch;
13     string giver,taker;
14     cin>>giver>>p>>ch>>m;
15     d[giver]-=p*m;
16     for(int j=0;j<m;j++){
17         cin>>taker;
18         d[taker]+=p;
19     }
20 }
21 cout<<d.size()<<endl;
22 for(mit=d.begin();mit!=d.end();mit++)
23     cout<<mit->first<<" "<<mit->second<<endl;
24 return 0;
```

# 作业4 加密规则

```
1  #include<iostream>
2  #include<string>
3  #include<map>
4  using namespace std;
5  int main() {
6      map<char,char> dict;
7      string a,b;
8      getline(cin,a);
9      getline(cin,b);
10     b=b+b[0];
11     for(int i=0;i<b.size()-1;i++)
12         dict[b[i]]=b[i+1];
13     for(int i=0;i<a.size();i++)
14         if(dict.count(a[i])) cout<<dict[a[i]];
15         else cout<<a[i];
16     return 0;
17 }
```

# 数据容器

map

一对一映射

+

struct

结构体

# map - 键/值 类型自定义

```
#include<map>
```

引入map库

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

每个string对应一个int

```
map < hero, score> d;
```

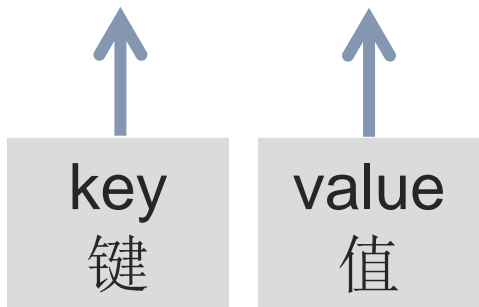
每个hero对应一个score

```
map <zombie,double> d;
```

每个zombie对应一个double

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

每个int对应一个zombie





# map键值自定义类型-基本操作

```
5 struct info{int IQ;string color;};
6 struct dog{string name;int year;};
7 struct cmp{
8     bool operator()(const dog&a,const dog&b)const{
9         return a.name<b.name||a.name==b.name&&a.year<b.year;
10    }
11 };

13 map<dog,info,cmp> d;
14 map<dog,info,cmp>::iterator mit;
```

# map键值自定义类型-基本操作

```
15 //insert elements
16 dog x={"mike",2017};
17 info y={250,"yellow"};
18 d[x]=y;
19 d[(dog){ "batman",2018}]=(info){59,"black"};
20 d[(dog){ "mike",2017}]=(info){25,"white"};
21 //begin(),end()
22 for(mit=d.begin();mit!=d.end();mit++){
23     cout<<mit->first.name<<" "<<mit->first.year<<" ";
24     cout<<mit->second.IQ<<" "<<mit->second.color<<endl;
25 }
26 //count()
27 cout<<d.count((dog){ "mike",2017})<<endl;
28 //find()
29 mit=d.find((dog){ "mike",2017});
30 //erase()
31 if(mit!=d.end()) d.erase(mit);
32 //size()
33 cout<<d.size()<<endl;
```

# 易错点

以下操作只针对key(键)，不能用于value(值)

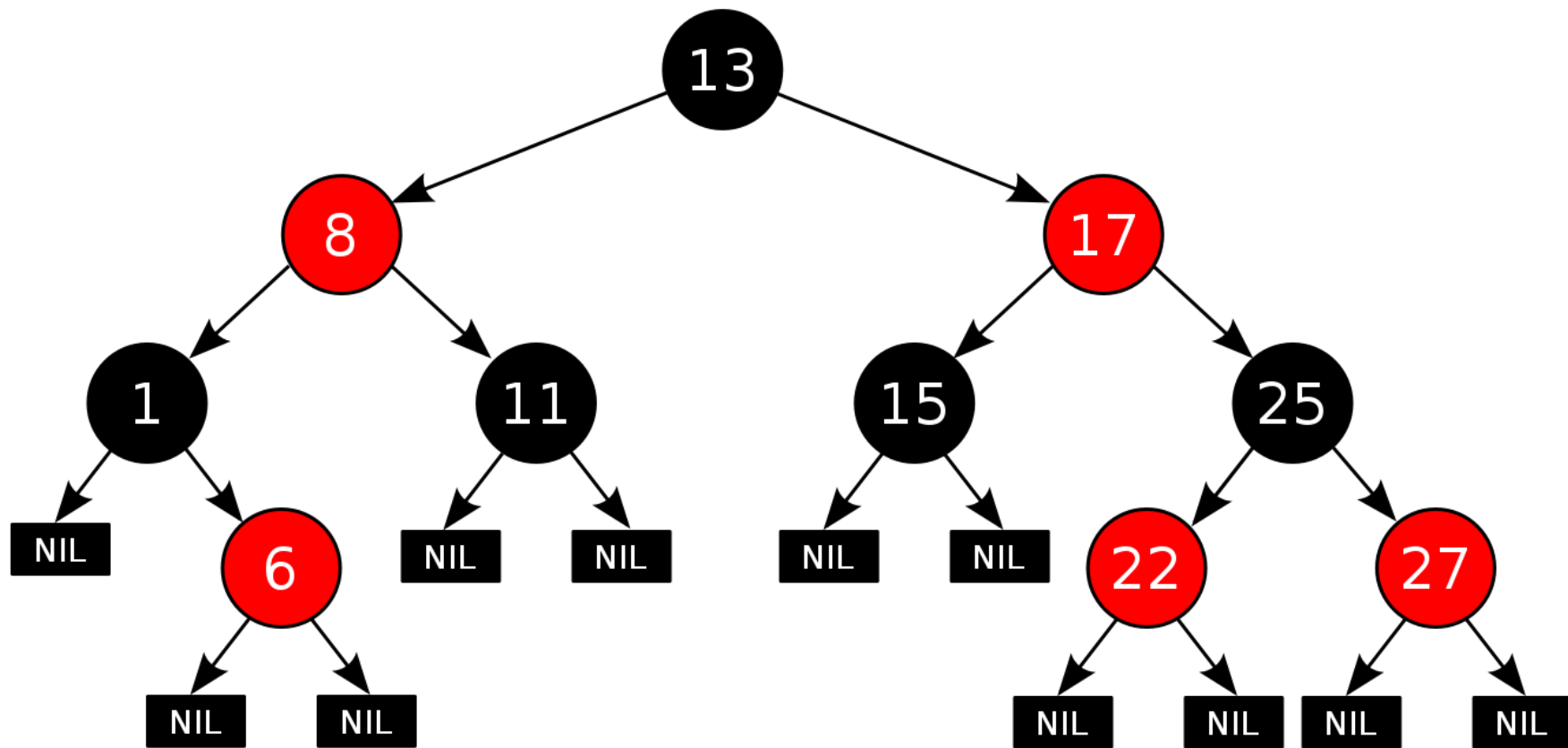
自动排序

计数count()

去重

查找find()

# 数据结构：红黑树RB-tree



map的底层实现是红黑树

# map 综合练习

# 字符统计II

输入一行字符串，输出其中每个字符出现的次数和首次出现的位置。  
按照**ASCII**码从小到大排列。

输入样例

```
for(mit=d.begin();mit!=d.end();mit++)
```

数据容器映射的  
是哪两种类型

输出样例

```
! 1 21  
( 3 3  
) 3 16  
+ 2 34  
. 2 9  
; 2 17  
= 2 7  
b 1 10  
d 3 8  
e 2 11  
f 1 0  
g 1 12  
i 4 5  
m 3 4  
n 2 14  
o 1 1  
r 1 2  
t 3 6
```

# 扑克高手II

扑克高手**MIKE**，他玩的游戏使用1幅扑克牌（不使用大鬼和小鬼）共**52**张牌。**MIKE**今天运气不好，一直在输，他怀疑扑克牌被动了手脚：如果他能发现**2**张牌花色和大小都一样，那么就能确定牌有问题。输入第一行是正整数**n**代表已经打出几张牌，以下**n**行依次为这些牌的花色和大小。一旦发现了问题，就输出最早发现的两张相同牌分别是第几张，否则输出**Nice hand**。

输入样例

```
5
diamond 8
spade A
diamond 10
heart Q
diamond 8
```

输出样例

```
1 5
```

输入样例

```
4
spade A
heart A
club A
diamond A
```

输出样例

```
Nice hand
```

数据容器映射的  
是哪两种类型

# 参考资料

---

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