

CS101

A Mars rover, likely a Curiosity rover, is shown on a rocky, orange-brown landscape. The rover is positioned in the center-left of the frame, facing right. It has six large, treaded wheels and a complex body with various instruments and cameras. The background shows a vast, flat, orange-brown horizon under a bright, hazy sky. The sun is visible in the upper right corner, creating a strong glare. The overall scene is a typical Mars surface environment.

信奥
算法

课件下载地址:

<http://pan.baidu.com/s/1nu6kYkL>

作业网站:

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

容器修改

1

找到旧元素

```
it=s.begin()
```

```
it=s.find(x)
```

2

建立新元素

```
dog y={"big",2017}
```

3

插入新元素

```
s.insert(y)
```

4

删除旧元素

```
if(it!=s.end()) s.erase(it)
```

易错点

容器在删除/插入操作以后
迭代器的位置会失效

迭代器需要重新定位

```
1 #include<iostream>
2 #include<set>
3 using namespace std;
4 int main() {
5     multiset<int> s;
6     multiset<int>::iterator it;
7     s.insert(8); s.insert(9);
8     s.insert(6); s.insert(7);
9     for(it=s.begin(); it!=s.end(); it++){
10         int x=*it;
11         cout<<x<<endl;
12         if(x>=100) break;
13         s.erase(it);
14         s.insert(x+10);
15     }
16     return 0;
17 }
```

数据容器

map

一对一映射

multimap

一对多映射

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



key
键



value
值

map - 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

第一次访问时
自动新建并赋值

map - 迭代器,begin(),end()

自动排序和去重

```
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['b'] = 100;
8      d['a'] = 200;
9      d['c'] = 300;
10     // show content:
11     for(it=d.begin();it!=d.end();++it)
12         cout<<it->first<<" "<<it->second<<endl;
13     for(it=d.begin();it!=d.end();++it)
14         cout<<(*it).first<<" "<<(*it).second<<endl;
15     return 0;
16 }
```


map - size()

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

map - 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 }
```

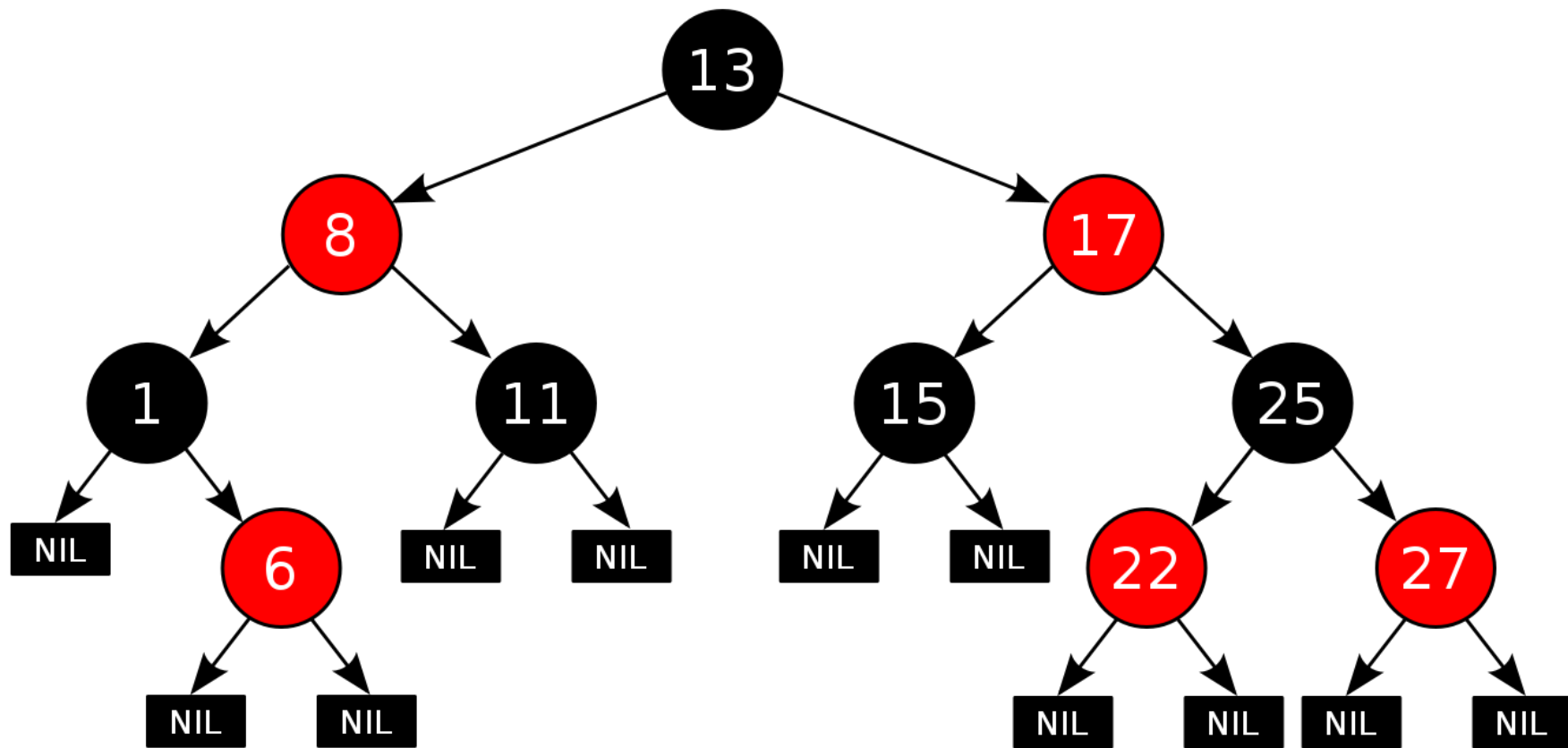
map - erase()

```
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     // insert some values:
8     d['a']=10; d['b']=20;
9     d['c']=30; d['d']=40;
10    d['e']=50; d['f']=60;
11    it=d.find('b');
12    if(it!=d.end()) d.erase(it); // erasing by iterator
13    d.erase('c'); // erasing by key
14    d.erase('x'); // erasing by key
15    // show content:
16    for(it=d.begin(); it!=d.end(); ++it)
17        cout<<it->first<<" "<<it->second<<endl;
18    return 0;
19 }
```

map - 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;  d['b']=100;
8      d['c']=150; d['d']=200;
9      it=d.find('b');
10     if(it!=d.end())
11         d.erase(it);
12     cout<<"a => "<<d.find('a')->second<<endl;
13     cout<<"c => "<<d.find('c')->second<<endl;
14     cout<<"d => "<<d.find('d')->second<<endl;
15     return 0;
16 }
```

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

输入一行字符串，输出其中每个字符出现的次数。按照字典序排列

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  
( 3  
) 3  
+ 2  
. 2  
; 2  
= 2  
b 1  
d 3  
e 2  
f 1  
g 1  
i 4  
m 3  
n 2  
o 1  
r 1  
t 3
```

扑克高手

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。如果发现了问题，就输出**I am angry**,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

```
I am angry
```

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

哎妈呀

英国女孩Emma Ya来到中国，发现她的名字和东北话“哎妈呀”太接近，走在大街上总以为别人在叫自己。她想为自己改个新名字，希望确定新名字是否在身边朋友里是唯一的。注意：她的姓是不改变的。

输入第一行为正整数 n ，代表她朋友一共多少人，以下 n 行为每个朋友的全名，名和姓之间有一个空格。最后一行为她想改的全名。输出Yes或者No，代表新名字是否在她朋友里是唯一的。 $n \leq 100$

输入样例

4

Emma Stone
Mike Chen
Vivian Zhang
Harry Potter
Vivian Ya

输入样例

2

Max Shen
Lucia Wen
John Ya

数据容器映射的
是哪两种类型

输出样例

No

输出样例

Yes

参考资料

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