

unordered_map是无序的，时间会快一点；

map是会根据key自动排序，时间会慢一点。

1.map基本概念

简介：

- map中所有元素都是pair
- pair中的第一个元素为key（键值），起到索引的作用，第二个为value（实值）
- 所有元素都会根据元素的key键值自动排序

本质：

- map / multimap属于**关联式容器**，底层结构是用二叉树实现。

优点：

- 可以根据key值快速找到value值

map和multimap的区别：

- map不允许有重复key值元素
- multimap允许容器中有重复key值元素

2.构造和赋值

map<int,int> m;

构造：

- m.insert(pair< int ,int >(2,21)); //使用对组pair插入
- m.insert(make_pair (3,33));
- m[key] = value; //直接插入

拷贝构造：

- map<int, int>m2(m); //拷贝构造

赋值：

- map<int, int>m3;

m3 = m; //赋值

遍历：

```
for (map<int, int>::iterator it = m.begin(); it != m.end(); it++)
{
    cout << (*it).first << " " << it->second << endl;
}
```

3.常用函数接口：

- m.size(); //返回map的大小
- m.empty(); //容器判空
- m.swap(m1); //交换

删除：

- m.erase(m.begin()); //按照迭代器来删除
- m.erase(3); //按照key值来删除
- m.erase(m.begin(), m.end()); //删除begin和end范围之间的元素
- m.clear(); //清空map

查找和统计：

- m.find(key); //查找key是否存在，若存在返回该元素的迭代器，不存在返回set.end()

```
auto pos = m.find(1);
if (pos != m.end())
    cout << pos->first << " " << pos->second;
else
    cout << "未找到元素" << endl;
```

- m.count(key); //统计key元素的个数

```
int num = m.count(3);
cout << "num = " << num << endl;
```

注意：map中find只有0和1，multimap可以有其他的值

4.map容器排序

利用仿函数实现key值逆序排序

1. 在main函数之前写上仿函数：

```
class Compare
{
public:
    bool operator()(int v1, int v2) const
    {
        //降序
        return v1 > v2;
    }
};
```

2.注意在构造map时要加上自己之前定义的仿函数名：

```
map<int, int, Compare> m;
m.insert(pair<int, int>(1, 22));    //使用对组pair插入
m[3] = 23;
m[4] = 52;
m[2] = 21;
m[5] = 21;
```

3.之后遍历就是降序输出了，结果如下：

```
5 21
4 52
3 23
2 21
1 22
```

5.map实现对value的排序

可以再定义一个map将原来的map的key存入新建map的value，原来map的value存入新建map的key，这样就实现对value的排序了。

6.unordered_map, unordered_set

普通的map和set内部实现了一个红黑树，红黑树具有自动排序的功能，因此map和set内部的所有元素都是有序的，红黑树的每一个节点都代表着map或set的一个元素。因此，对于map进行的查找，删除，添加等一系列的操作都相当于是对红黑树进行的操作。

unordered_map，unordered_set内部实现了一个哈希表（也叫散列表，通过把关键码值映射到Hash表中一个位置来访问记录，查找的时间复杂度可达到 $O(1)$ ，其在海量数据处理中有着广泛应用）。因此，其元素的排列顺序是无序的。

使用时，需要添加相应的头文件：`#include <unordered_map>`,`#include <unordered_set>`

map的基本操作：

1.计算出现次数

```
cin >> s;  
map<char, int> m;  
for(int j = 0; j < s.size(); j++) {  
    m[s[j]]++;  
    if (s[j] == 'P') p = j;  
    if (s[j] == 'T') t = j;  
}
```