

unordered_map

增:

```
hash[key] = val  
hash.insert(make_pair(key, val));
```

删:

```
hash.erase(key);
```

改:

查:

```
hash.count(val) > 0 or if( hash.count(val) )  
hash.find(key) != hash.end()
```

排序:

Not

Unordered_map 从vector 的快速创建 (和 unorded_set 一样, 没有重复元素???)

```
unordered_map<int, bool> hash;  
for(auto val: nums) hash[val] = true;  
  
hash.erase(word);          // unorded_set 可以删除要删除的值, 而在vector中就必须删除指定位置的iterator了。
```

插入元素

```
hash.insert(make_pair(val,boo);  
// or  
hash[val] = val;
```

判断某元素存不存在 (和unordered_map一样)

```
while(hash.count(val) ) // 或者这样, sets.count() > 0  
/// or  
while( hash.find(val) != sets.end() )
```

实例二: 统计单词出现的频率

```

/*****
    File Name: main.cpp
    Author: bin.wang
    Mail:    sa615168@mail.ustc.edu.cn
    Created Time: Fri 16 Jun 2017 09:03:37 PM CST
*****/

#include <iostream>
#include <map>
#include <unordered_map>
#include <fstream>
using namespace std;
int main()
{
    unordered_map<int ,int>  result;
    fstream infile("./number.txt");
    int temp;
    while(infile >> temp){
        result[temp]++;    //用法来源于c++primer c++11, stl
    }

    for(auto num:result){
        if(num.second == 1) cout<< num.first  << endl;    // 每一个值有这个
    }
    return 0;
}

```