Lesson 9

1.在容器中删除元素时，会导致该元素的迭代器失效，所以当使用 erase 时，应该用迭代器接收返回值。

2.map<K, V>是一种 pair 的容器，pair 的种类是 pair<K, V>

3.map 中的元素按照 key 进行从小到大排列。

4.map 的底层实现是采用二叉树，一般是使用红黑树。

5.map 对容器的要求：key 类型必须支持<比较，而 value 不做要求。

6.使用下标访问 map 中不存在的元素时，会增加一个新的键值对。

7.words[word]++;如果 word 是一个不存在的值，那么首先在 map 中添加一个键值对，为(word, 0)，然后对 value 执行+1 操作。

8.map 采用下标访问已经存在的 key，会更新 value。而使用 insert 则导致插入失败，并且不会更新 value。

9.map 的 key 值是不可更改的。

10.set 底层采用红黑树实现，按照值进行排序。

11.map 中 key 的值是唯一的，set 中的元素都是唯一的。

12.练习：

a) 统计单词频率

b) 本地保存一个停用词文件，表示不需要统计的词汇

13.编译器自动为我们合成一个拷贝构造函数。A(const A &).

14.对象复制的时机：

a) 显式复制。

b) 使用对象做形参

c) 使用对象做返回值

d) 往容器中放入对象

15.浅拷贝和深拷贝是对类中持有指针而言，如果对象复制的时候，仅仅去拷贝指针的值，这称为浅拷贝（shallow copy），如果不是去拷贝指针的值，而是去拷贝指针指向的内存空间，称为深拷贝（deep copy）。

16.执行完深拷贝之后，对象之间没有任何关联。

17.对象的赋值，调用的是类的赋值运算符。

18.编译器自动为我们合成一个赋值运算符。

19.编写赋值运算符和拷贝构造函数的区别：

a) 赋值运算符可能需要先释放资源

b) 赋值预算符需要考虑自身赋值问题。

20.编写赋值预算符的准则：

a) 必须处理自身赋值问题

b) 必须返回自身引用

21.三法则：拷贝构造函数、赋值运算符、析构函数，三者都与类内部的指针密切相关。当需要编写其中一个时，一般需要编写其他两个。

22.当需要禁止一个类进行复制或者赋值时，只需将类的拷贝构造函数和赋值运算符设为私有，而且只提供声明。

23.可以采用这样一个宏去禁用 copy 或者 assign 的能力：

```
#define DISALLOW_COPY_AND_ASSIGN(TypeName) \
        TypeName(const TypeName&); \
        void operator=(const TypeName&)
```

注意后面没有分号。

24.作业：给之前的 Queue 加上三法则。