1、顺序式的容器，Array是C++11新增的东西，也是一个类（把语言的数组包装成一个类）

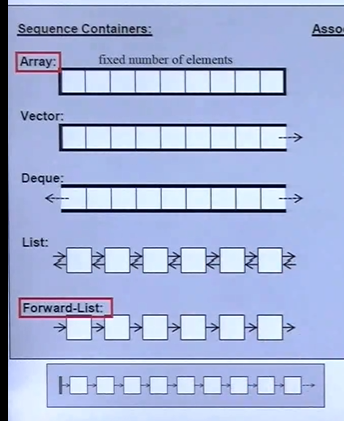
Array的空间是固定的，不能改变。

Vector的空间不够是可以在末尾自动增长，是分配器allocator帮它做内存的事情。

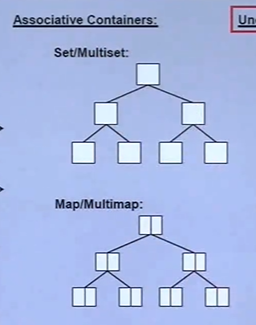
Deque双向队列，两端可进可出

List链表，用双向指针穿起来

Forward-List单向链表，是C++11新增的，消耗的内存没有List多，



2、关联式的容器，用key找value，适合快速的/大量的查找



用红黑树实现的，红黑树是一种特殊的二叉树，是一种高度平衡的二叉树，避免了查找时最坏的情况（比如再最右边一直找找找），左右平衡是自己自动调整的。

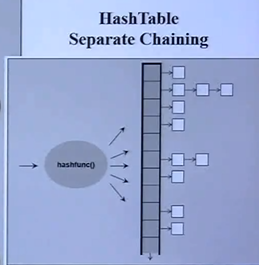
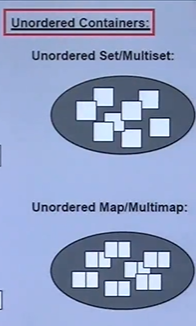
Map的每一个结点有两个部分，一个是key一个是value。

而Set中比较特殊，可以理解为，key就是value，value就是key，两种不分。

没有Multi时，set里面的元素不能重复，map里面的key不能重复。

有Multi时，key可以重复。

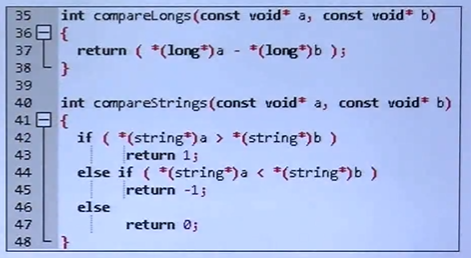
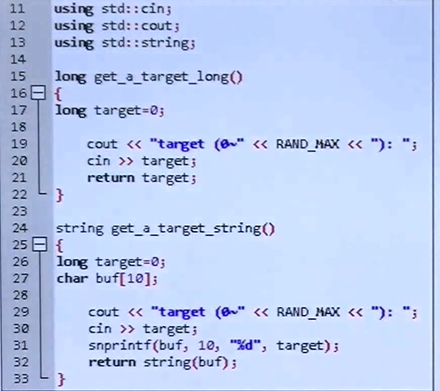
3、不定序的容器，底部是用哈希表做的，其实也可以放在关联式容器的一类，C++11新增的东西。



可以想象有一系列的“篮子”，“篮子”里面有一个链表，公认Hash Table最佳实现方法是Separate Chaining

Hash Table：假如你有20个空间放元素，这些元素需要通过一个“计算”来才知道放在这20个空间中的哪一个。如果有重复的元素计算得到要放在相同的“篮子”，那么就会发生碰撞。因为这些解决“碰撞”的方法没有那么理想，就把一个个“篮子”变成链表头，让“碰撞的”元素一个个在“篮子”里接下去就好了。不过“篮子的”链表不能太长。

4、



交互1：选择一个容器、选择容器数量，再随机在0~32767中选择这么数量的元素放进容器中，最后返回一个时间（微秒级）

交互2：输入一个想要查找的元素。

24行：将int字符转换成字符串（snprintf函数）

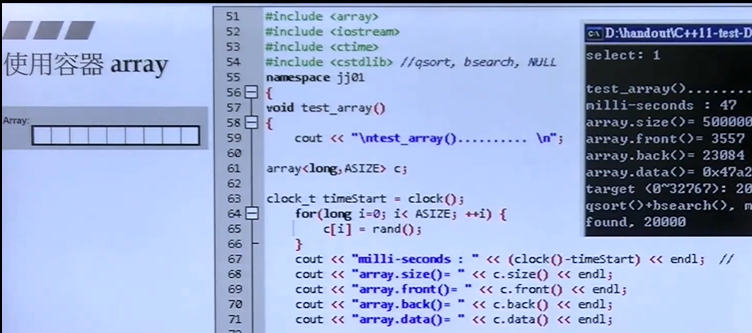
5、

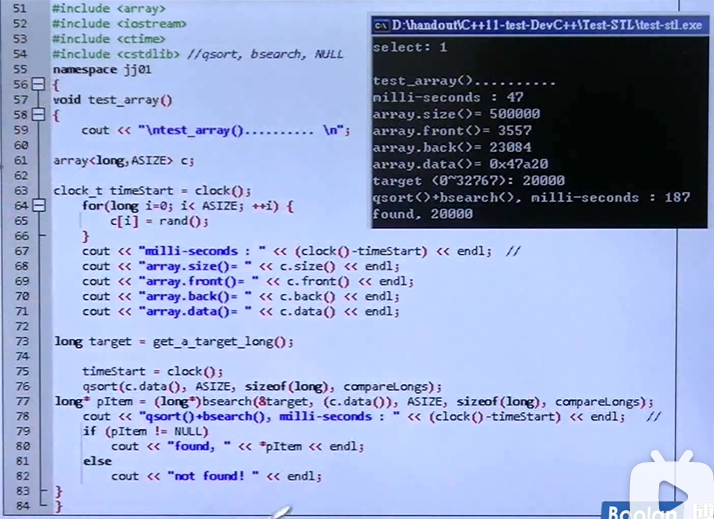
61行：array<long, ASIZE> c; //第一个参数说明类型，第二个参数说明大小

71行：c.data()会返回第一个元素（起点数值）的地址

76行：qsort(1，2，3，4)（在cstdlib），1是起始元素地址，2是元素个数，3是单个元素大小，4是怎么比较大小

Clock()（在ctime）返回程序运行至当前时刻所有的毫秒值。





二分查找前必须要先排好序