1. 标准模板库STL
2. 算法algorithm

含有算法的头文件： <algorithm> <numeric> 等

//插入排序：

如果无序序列第一个值比有序序列中最大值更大，则无需继续

如果比有序序列中最大值更小，则继续

j从i-1开始寻找，如果小于时将j+1赋值为j

Remove与remove\_if算法：

这样的方法并不删除数据，而是将不符合删除条件的数据搬到前面

这样能避免迭代器变成野指针，适合中间单个数据的删除。

1. 迭代器iterator

1、分类：

输入迭代器

输出迭代器

前向迭代器

双向迭代器

随机访问迭代器

1. 实质：是一个指针，故而支持->与\*的操作符（默认已经重载）
2. 使用：iterator是专属于某个容器的，不能单独存在

它十分类似普通指针，但指向的是容器中的相对地址而非绝对地址

如 list<int>::iterator p = link.begin( );

//获取一个指向容器首部的迭代器

定义时需要注明作用域，即类名及其实例化

1. 函数对象
2. 定义：一个行为类似函数的对象

可以没有参数，也可以有若干个参数

其功能是获取一个值，确定相关法则并改变相关值

普通函数是函数对象，重载了括号（）运算符的函数也是函数对象

1. 分类
   * 1. 产生器
     2. 一元函数对象（一元谓词）
     3. 二元函数对象（二元谓词）

//一元谓词与二元谓词返回的都是一个bool

1. 使用举例：find与find\_if

//attention：这两个函数都只能找到第一个满足条件的对象而非所有

find(起始位置，结束位置，一元谓词查找对象)

①（普通数据类型）“在一组整数中寻找一个大于7的数并输出”

list<int> :: iterator it = find(l.begin() , l.last() , 2);

如果找到了会返回last

②（类类型的查找，需要重载==操作符）

“查找person类中名字与所给字相同的对象”

find\_if( 与find相同 )

适用于内容为指针等类型的查找

1. 常用的STL容器
2. Vector向量（可变大小数组）
   1. 构造方式

vector<int> v1;

Vector<int> v2(v1.begin,v1.end);等

* 1. 赋值方式
  2. 插入和删除

push\_back() / insert( 迭代器位置，元素值 )

Pop\_back() / erase()

* 1. 容量查询 capacity()

1. Deque双端队列
2. List双向循环链表
3. Forward\_list
4. Array
5. Map/Multi map 映射/多重映射（需要以pair成对插入key与value）
6. 容器适配器
7. stack堆栈
8. Queue队列

//堆栈与队列的默认封装都是双端队列deque

1. Priority\_queue优先队列

//默认封装是vector

参考书：  
 the C++ standard library

Effective STL

STL Programmer’s Guide