## STL全部的算法

**注意：这个文档的主要功能还是给STL的算法分个类，要查看算法严格的描述，还是去微软官网查看吧，官方文档又准又全。**

#### <一>查找算法(13个)：判断容器中是否包含某个值

##### adjacent\_find:

在iterator对标识元素范围内，查找一对相邻重复元素，找到则返回指向这对元素的第一个元素forwardIterator。否则返回最后一个元素的forwardIterator。

##### binary\_search:

在有序序列中查找value，找到返回true。重载的版本实用指定的比较函数对象或函数指针来判断相等。

##### count:

利用等于操作符，把标志范围内的元素与输入值比较，返回相等元素个数。

##### count\_if:

利用输入的操作符，对标志范围内的元素进行操作，返回结果为true的个数。

##### equal\_range:

**注意，必须对有序容器进查找，下面的lower\_bound和upper\_bound也是同理。**

功能类似equal，返回一对iterator，第一个表示lower\_bound，第二个表示upper\_bound。

find利用底层元素的等于操作符，对指定范围内的元素与输入值进行比较。当匹配时，结束搜索，返回该元素的一个InputIterator。

##### find\_end:

在指定范围内查找"由输入的另外一对iterator标志的第二个序列"的最后一次出现。找到则返回最后一对的第一个迭代器，否则返回输入的"另外一对"的第一个迭代器。重载版本使用用户输入的操作符代替等于操作。

##### find\_first\_of:

在指定范围内查找"由输入的另外一对iterator标志的第二个序列"中任意一个元素的第一次出现。重载版本中使用了用户自定义操作符。

##### find\_if:

使用输入的函数代替等于操作符执行find。

##### lower\_bound:

返回一个iterator，指向在有序序列范围内的可以插入指定值而不破坏容器顺序的第一个位置。重载函数使用自定义比较操作。

##### upper\_bound:

返回一个iterator，指向在有序序列范围内插入value而不破坏容器顺序的最后一个位置，该位置标志一个大于value的值。重载函数使用自定义比较操作。

##### search:

**这两个是真的不好描述，去微软官网查看一下吧，简单，比我在这里总结的强多了。**

search\_n:

#### <二>排序和通用算法(14个)：提供元素排序策略

##### inplace\_merge:

##### merge:

##### nth\_element:

##### partial\_sort:

##### partial\_sort\_copy:

##### partition:

##### random\_shuffle:

##### reverse:

##### reverse\_copy:

##### rotate:

##### rotate\_copy:

##### sort:

##### stable\_sort:

##### stable\_partition:

#### <三>删除算法(15个)

##### copy:

##### copy\_backward:

##### iter\_swap:

##### remove:

##### remove\_copy:

##### remove\_if:

##### remove\_copy\_if:

##### replace:

##### replace\_copy:

##### replace\_if:

##### replace\_copy\_if:

##### swap:

##### swap\_range:

##### unique:

##### unique\_copy:

#### <四>排列组合算法(2个)：提供计算给定集合按一定顺序的所有可能排列组合

##### next\_permutation:

##### prev\_permutation:

#### <五>算术算法(4个)

##### accumulate:

##### partial\_sum:

##### inner\_product:

##### adjacent\_difference:

#### <六>生成和异变算法(6个)

##### fill:

##### fill\_n:

##### for\_each:

##### generate:

##### generate\_n:

##### transform:

#### <七>关系算法(8个)

##### equal:

##### includes:

##### lexicographical\_compare:

##### max:

##### max\_element:

##### min:

##### min\_element:

##### mismatch:

#### <八>集合算法(4个)

##### set\_union:

##### set\_intersection:

##### set\_difference:

##### set\_symmetric\_difference:

#### <九>堆算法(4个)

##### make\_heap:

##### pop\_heap:

##### push\_heap:

##### sort\_heap: