

C++标准程序库读书笔记

STL算法

包含头文件 <algorithm>或<numeric>

STL算法采用覆盖模式而非安插模式，所以调用者必须保证目标区间拥有足够的元素空间。也可以运用特殊的安插型迭代器将覆盖模式改变为安插模式。

可以运用条件判断式完成以下工作：

- 对于搜寻算法，可以传递一个函数或者仿函数，指定一个一元判断式作为搜寻准则。
- 对于排序算法，可以传递一个函数或者仿函数，指定一个二元判断式作为排序准则。
- 可以传递一个一元判断式作为准则，判断是否应该对某些元素施以某项运算。
- 可以为某个数值算法指定一个数值运算。

判断式不应该在函数调用过程中改变其自身状态

STL算法命名时，引入两个特别的尾词：

1. 尾词 `_if`
2. 尾词 `_copy`

STL算法分类：

- 非变动性算法 **nonmodifying algorithm**
- 变动性算法 **modifying algorithm**
- 移除性算法 **removing algorithm**
- 变序性算法 **mutating algorithm**
- 排序算法 **sorting algorithm**
- 已序区间算法 **sorted range algorithm**
- 数值算法 **numeric algorithm**

非变动性算法

通过input和forward迭代器完成工作，因此可以作用于所有标准容器身上。

名称	作用
for_each()	对每个元素执行某操作
count()	返回元素个数
count_if()	返回满足某一个准则的元素个数
min_element()	返回最小值元素（以一个迭代器表示）
max_element()	返回最大值元素
find()	搜寻等于某值的第一个元素
find_if()	搜寻满足某个准则的第一个元素
search_n()	搜寻具有某些特性的第一段n个连续元素
search()	搜寻某个子区间第一次出现位置
find_end()	搜寻某个子区间最后一次出现位置
find_first_of()	搜寻等于某数个值之一的第一个元素
adjacent_find()	搜寻连续两个相等的（或者符合特等准则）的元素
equal()	判断两区间是否相等
mismatch()	返回两个序列的各组对应元素中，第一对不相等元素
lexicographical_compare()	判断某一序列在字典顺序下是否小于另一序列

与string 搜寻函数的比较

|搜寻|string函数|STL算法| |--|-----|----: ||某元素第一次出现位置|find())|find())| |某元素最后一次出现位置|rfind())|find()), 采用逆向迭代器| |某子区间第一次出现位置|find())|search())| |某子区间最后一次出现位置|rfind())|find_end())| |某数个元素第一次出现位置|find first of())|find first of())| |某数个元素最后一次出现位置|

置|find_last_of()|find_first_of()采用逆向迭代器|

|n个连续元素第一次出现位置||search_n()|

变动性算法

名称	效果
foreach()	针对某个元素执行某项操作
copy()	从第一个元素开始，复制某段区间
copy_backward()	从最后一个元素开始，复制某段区间
transform()	变动元素，将两个区间的元素合并
merge()	合并两个区间
swap_range()	交换两区间内的元素
fill()	以给定值替换每一个元素
fill_n()	以给定值替换n个元素
generate()	以某项操作的结果替换每一个结果
generate_n()	以某项操作的结果替换n个元素
replace()	将具有某项特定值的元素替换为另一个值
replace_if()	将符合某准则的元素替换为另一个值
replace_copy()	复制整个区间，同时并将具有某特定值的元素替换为另一个值
replace_copy_if()	复制整个区间，同时并将符合某准则的元素替换为另一个值

移除性算法

可以移除某区间内的元素，也可以在复制过程中执行移除动作。

名称	效果
remove()	将等于某特定值的元素全部移除
remove_if()	将满足某准则的元素全部移除
remove_copy()	将不等于某特定值的元素全部复制到它处
remove_copy_if()	将不满足某准则的元素全部复制到它处
unique()	移除毗邻的重复元素
unique_copy()	移除毗邻的重复元素，并复制到它处

移除算法只是在逻辑上移除元素，手段是:将不需被移除的元素往前覆盖应被移除的元素，因此它并不改变操作区间内的元素个数，而是返回逻辑上的新终点位置。

变序性算法

通过元素值得赋值和交换，改变元素顺序但不改变元素值。

名称	效果
reverse()	将元素的次序逆转
reverse_copy()	复制的同时，逆转元素顺序
rotate()	旋转元素次序
rotate_copy()	复制的同时，旋转元素次序
next_permutation()	得到元素的下一个排列次序
prev_permutation()	得到元素的上一个排列次序
random_shuffle()	将元素的次序随机打乱
partition()	改变元素次序，将符合某准则者移到前面
stable_partiton()	与partition() 相似，但保持符准则与不符合准则之各个元素之间的相对位置

排序算法

排序算法的复杂度通常低于线性算法，而且需要动用随机存取迭代器。

名称	效果

sort()	对所有元素排序
stable_sort()	对所有元素排序，并保持相等元素间的相等次序
partial_sort()	排序，直到前n个元素就位
partial_sort_copy()	排序，直到前n个元素就位，结果复制于它处
nth_element()	根据第n个位置进行排序
partition()	改变元素次序，使符合某准则的元素放在前面
stable_partiton()	与partiton()相同，但保持符合准则和不符合准则的各个元素之间的相对位置
make_heap()	将一个区间转换为一个heap
push_heap()	将元素加入一个heap
pop_heap()	从heap移除一个元素
sort_heap()	对heap进行排序（执行后就不再是个heap了）

- sort()内部采用quicksort算法
- partial_sort()内部采用heapsort算法
- stable_sort()内部采用mergesort()

已序区间算法

名字	效果
binary_search()	判断某区间内是否包含某个元素
includes()	判断某区间内的每一个元素是否都涵盖于另一个区间中
lower_bound()	搜寻第一个大于等于给定值的元素
upper_bound()	搜寻第一个大于给定值的元素
equal_range()	返回等于给定值的所有元素构成的区间
merge()	将两个区间的元素合并
set_union()	求两个区间的并集
set_intersection()	求两个区间的交集
set_difference()	求位于第一区间但不位于第二区间的所有元素，形成一个已序区间
set_symmetric_difference()	找出只出现于两区间之一的所有元素，形成一个已序区间
inplace_merge()	将两个连续的已序区间合并

数值算法

名字	效果
accumulate()	组合所有元素（求总和，求乘积）
inner_product()	组合两区间内的所有元素
adjacent_difference()	将每个元素和其前一元素组合
	partial_sum()