

算法库

算法库提供大量用途的函数（例如查找、排序、计数、操作），它们在元素范围上操作。注意范围定义为 [first, last)，其中 last 指代要查询或修改的最后元素的^后一个元素。

受约束算法

C++20 在命名空间 std::ranges 中提供大多数算法的受约束版本，能以迭代器-哨位对或单个 range 参数指定范围，并且支持投影和指向成员指针可调用对象。另外更改了大多数算法的返回类型，以返回算法执行过程中计算的所有潜在有用信息。(C++20 起)

```
std::vector<int> v = {7, 1, 4, 0, -1};
std::ranges::sort(v); // 受约束算法
```

执行策略

大多数算法拥有接受执行策略的重载。标准算法库提供几种执行策略，并提供对应执行策略的类型和对象。用户可以静态地选择执行策略，通过以对应类型的执行策略对象为参数，调用并行算法。

标准库实现（但不是用户）可以定义附加的执行策略作为扩展。以实现定义类型的执行策略对象调用的并行算法的语义是实现定义的。

允许算法的并行版本（除了 std::for_each 与 std::for_each_n）从范围进行任意的元素复制，只要 std::is_trivially_copy_constructible_v<T> 与 std::is_trivially_destructible_v<T> 均为 true，其中 T 是元素的类型。(C++17 起)

在标头 <execution> 定义 在命名空间 std::execution 定义	
sequenced_policy	(C++17)
parallel_policy	(C++17) 执行策略类型
parallel_unsequenced_policy	(C++17) (类)
unsequenced_policy	(C++20)
seq	(C++17)
par	(C++17)
par_unseq	(C++17)
unseq	(C++20)
在命名空间 std 定义	
is_execution_policy	(C++17)
测试一个类是否表示某种执行策略 (类模板)	

不修改序列的操作

在标头 <algorithm> 定义	
all_of	(C++11)
any_of	(C++11)
none_of	(C++11)
检查谓词是否对范围中所有、任一或无元素为 true (函数模板)	
ranges::all_of	(C++20)
ranges::any_of	(C++20)
ranges::none_of	(C++20)
检查谓词是否对范围中所有、任一或无元素为 true (nieblويد)	
for_each	应用函数到范围中的元素 (函数模板)
ranges::for_each	(C++20)
应用函数到范围中的元素 (nieblويد)	
for_each_n	(C++17)
应用一个函数对象到序列的前 n 个元素 (函数模板)	
ranges::for_each_n	(C++20)
应用函数对象到序列的首 n 个元素 (nieblويد)	
count	返回满足指定判别标准的元素数
count_if	(函数模板)
ranges::count	(C++20)
ranges::count_if	(C++20)
返回满足指定判别标准的元素数 (nieblويد)	

mismatch	寻找两个范围出现不同的首个位置 (函数模板)
ranges::mismatch (C++20)	寻找两个范围出现不同的首个位置 (niebloid)
find find_if find_if_not (C++11)	寻找首个满足特定判别标准的元素 (函数模板)
ranges::find (C++20) ranges::find_if (C++20) ranges::find_if_not (C++20)	寻找首个满足特定判别标准的元素 (niebloid)
find_end	在特定范围中寻找最后出现的元素序列 (函数模板)
ranges::find_end (C++20)	在特定范围中寻找最后出现的元素序列 (niebloid)
find_first_of	搜索元素集中的任意元素 (函数模板)
ranges::find_first_of (C++20)	搜索元素集中的任一元素 (niebloid)
adjacent_find	查找首对相邻的相同（或满足给定谓词的）元素 (函数模板)
ranges::adjacent_find (C++20)	查找首对相邻的相同（或满足给定谓词的）元素 (niebloid)
search	搜索一个元素范围 (函数模板)
ranges::search (C++20)	搜索一个元素范围 (niebloid)
search_n	在范围中搜索一定量的某个元素的连续副本 (函数模板)
ranges::search_n (C++20)	在范围中搜索一定量的某个元素的连续副本 (niebloid)
ranges::starts_with (C++23)	检查一个范围是否始于另一范围 (niebloid)
ranges::ends_with (C++23)	检查一个范围是否终于另一范围 (niebloid)

修改序列的操作

在标头 <algorithm> 定义	
copy copy_if (C++11)	将某一范围的元素复制到一个新的位置 (函数模板)
ranges::copy (C++20) ranges::copy_if (C++20)	将某一范围的元素复制到一个新的位置 (niebloid)
copy_n (C++11)	将一定数目的元素复制到一个新的位置 (函数模板)
ranges::copy_n (C++20)	将一定数目的元素复制到一个新的位置 (niebloid)
copy_backward	按从后往前的顺序复制一个范围内的元素 (函数模板)
ranges::copy_backward (C++20)	按从后往前的顺序复制一个范围内的元素 (niebloid)
move (C++11)	将某一范围的元素移动到一个新的位置 (函数模板)
ranges::move (C++20)	将某一范围的元素移动到一个新的位置 (niebloid)
move_backward (C++11)	按从后往前的顺序移动某一范围的元素到新的位置 (函数模板)
ranges::move_backward (C++20)	按从后往前的顺序移动某一范围的元素到新的位置 (niebloid)
fill	将一个给定值复制赋值给一个范围内的每个元素 (函数模板)
ranges::fill (C++20)	将一个给定值复制赋值给一个范围内的每个元素 (niebloid)
fill_n	将一个给定值复制赋值给一个范围内的 N 个元素 (函数模板)
ranges::fill_n (C++20)	将一个值复制赋值给一定量的元素 (niebloid)

transform	将一个函数应用于某一范围的各个元素，并在目标范围存储结果 (函数模板)
ranges::transform (C++20)	将一个函数应用于某一范围的各个元素 (niebloid)
generate	将相继的函数调用结果赋值给一个范围中的每个元素 (函数模板)
ranges::generate (C++20)	保存函数结果到一个范围中 (niebloid)
generate_n	将相继的函数调用结果赋值给一个范围中的 N 个元素 (函数模板)
ranges::generate_n (C++20)	保存 N 次函数应用的结果 (niebloid)
remove remove_if	移除满足特定判别标准的元素 (函数模板)
ranges::remove (C++20) ranges::remove_if (C++20)	移除满足特定判别标准的元素 (niebloid)
remove_copy remove_copy_if	复制一个范围的元素，忽略满足特定判别标准的元素 (函数模板)
ranges::remove_copy (C++20) ranges::remove_copy_if (C++20)	复制一个范围的元素，忽略满足特定判别标准的元素 (niebloid)
replace replace_if	将所有满足特定判别标准的值替换为另一个值 (函数模板)
ranges::replace (C++20) ranges::replace_if (C++20)	将所有满足特定判别标准的值替换为另一个值 (niebloid)
replace_copy replace_copy_if	复制一个范围内的元素，并将满足特定判别标准的元素替换为另一个值 (函数模板)
ranges::replace_copy (C++20) ranges::replace_copy_if (C++20)	复制一个范围内的元素，并将满足特定判别标准的元素替换为另一个值 (niebloid)
swap	交换两个对象的值 (函数模板)
swap_ranges	交换两个范围的元素 (函数模板)
ranges::swap_ranges (C++20)	交换两个范围的元素 (niebloid)
iter_swap	交换两个迭代器所指向的元素 (函数模板)
reverse	逆转范围中的元素顺序 (函数模板)
ranges::reverse (C++20)	逆转范围中的元素顺序 (niebloid)
reverse_copy	创建一个范围的逆向副本 (函数模板)
ranges::reverse_copy (C++20)	创建一个范围的逆向副本 (niebloid)
rotate	旋转范围中的元素顺序 (函数模板)
ranges::rotate (C++20)	旋转范围中的元素顺序 (niebloid)
rotate_copy	复制并旋转元素范围 (函数模板)
ranges::rotate_copy (C++20)	复制并旋转元素范围 (niebloid)
shift_left shift_right (C++20)	迁移范围中的元素 (函数模板)
ranges::shift_left ranges::shift_right (C++23)	迁移范围中的元素 (niebloid)
random_shuffle (C++17 前) shuffle (C++11)	随机重排范围中的元素 (函数模板)
ranges::shuffle (C++20)	随机重排范围中的元素 (niebloid)
sample (C++17)	从一个序列中随机选择 n 个元素 (函数模板)

ranges::sample (C++20)	从序列中随机选择 n 个元素 (niebloid)
unique	移除范围内的连续重复元素 (函数模板)
ranges::unique (C++20)	移除范围中的连续重复元素 (niebloid)
unique_copy	创建某范围的不含连续重复元素的副本 (函数模板)
ranges::unique_copy (C++20)	创建某范围的不含连续重复元素的副本 (niebloid)

划分操作

在标头 <algorithm> 定义	
is_partitioned (C++11)	判断范围是否已按给定的谓词划分 (函数模板)
ranges::is_partitioned (C++20)	判断范围是否已按给定的谓词划分 (niebloid)
partition	将范围中的元素分为两组 (函数模板)
ranges::partition (C++20)	将范围中的元素分为二组 (niebloid)
partition_copy (C++11)	复制一个范围，将各元素分为两组 (函数模板)
ranges::partition_copy (C++20)	复制一个范围，将各元素分为二组 (niebloid)
stable_partition	将元素分为两组，同时保留其相对顺序 (函数模板)
ranges::stable_partition (C++20)	将元素分成二组，同时保持其相对顺序 (niebloid)
partition_point (C++11)	定位已划分范围的划分点 (函数模板)
ranges::partition_point (C++20)	定位已划分范围的划分点 (niebloid)

排序操作

在标头 <algorithm> 定义	
is_sorted (C++11)	检查范围是否已按升序排列 (函数模板)
ranges::is_sorted (C++20)	检查范围是否以升序排序 (niebloid)
is_sorted_until (C++11)	找出最大的已排序子范围 (函数模板)
ranges::is_sorted_until (C++20)	寻找最大的已排序子范围 (niebloid)
sort	将范围按升序排序 (函数模板)
ranges::sort (C++20)	将范围按升序排序 (niebloid)
partial_sort	排序一个范围的前 N 个元素 (函数模板)
ranges::partial_sort (C++20)	排序一个范围的前 N 个元素 (niebloid)
partial_sort_copy	对范围内的元素进行复制并部分排序 (函数模板)
ranges::partial_sort_copy (C++20)	对范围内的元素进行复制并部分排序 (niebloid)
stable_sort	将范围内的元素排序，同时保持相等的元素之间的顺序 (函数模板)
ranges::stable_sort (C++20)	将范围内的元素排序，同时保持相等的元素之间的顺序 (niebloid)
nth_element	将给定的范围部分排序，确保其按给定元素划分 (函数模板)
ranges::nth_element (C++20)	将给定的范围部分排序，确保其按给定元素划分 (niebloid)

二分搜索操作（在已排序范围上）

在标头 <code><algorithm></code> 定义	
<code>lower_bound</code>	返回指向第一个 不小于 给定值的元素的迭代器 (函数模板)
<code>ranges::lower_bound</code> (C++20)	返回指向首个 不小于 给定值的元素的迭代器 (niebloid)
<code>upper_bound</code>	返回指向第一个 大于 给定值的元素的迭代器 (函数模板)
<code>ranges::upper_bound</code> (C++20)	返回指向首个 大于 某值的元素的迭代器 (niebloid)
<code>binary_search</code>	确定元素是否存在于某范围中 (函数模板)
<code>ranges::binary_search</code> (C++20)	确定元素是否存在于某范围中 (niebloid)
<code>equal_range</code>	返回匹配特定键值的元素范围 (函数模板)
<code>ranges::equal_ranges</code> (C++20)	返回匹配特定值的元素范围 (niebloid)

其他已排序范围上的操作

在标头 <code><algorithm></code> 定义	
<code>merge</code>	归并两个有序范围 (函数模板)
<code>ranges::merge</code> (C++20)	归并两个有序范围 (niebloid)
<code>inplace_merge</code>	就地归并两个有序范围 (函数模板)
<code>ranges::inplace_merge</code> (C++20)	在原位归并两个有序范围 (niebloid)

集合操作（在已排序范围上）

在标头 <code><algorithm></code> 定义	
<code>includes</code>	若一个序列是另一个的子列则返回 <code>true</code> (函数模板)
<code>ranges::includes</code> (C++20)	若一个序列是另一个的子列则返回 <code>true</code> (niebloid)
<code>set_difference</code>	计算两个集合的差集 (函数模板)
<code>ranges::set_difference</code> (C++20)	计算两个集合的差集 (niebloid)
<code>set_intersection</code>	计算两个集合的交集 (函数模板)
<code>ranges::set_intersection</code> (C++20)	计算两个集合的交集 (niebloid)
<code>set_symmetric_difference</code>	计算两个集合的对称差 (函数模板)
<code>ranges::set_symmetric_difference</code> (C++20)	计算两个集合的对称差 (niebloid)
<code>set_union</code>	计算两个集合的并集 (函数模板)
<code>ranges::set_union</code> (C++20)	计算两个集合的并集 (niebloid)

堆操作

在标头 <code><algorithm></code> 定义	
<code>is_heap</code>	检查给定范围是否为一个最大堆 (函数模板)
<code>ranges::is_heap</code> (C++20)	检查给定范围是否为最大堆 (niebloid)
<code>is_heap_until</code> (C++11)	查找能成为最大堆的最大子范围 (函数模板)

ranges::is_heap_until (C++20)	寻找能成为最大堆的最大子范围 (niebloid)
make_heap	从一个元素范围创建出一个最大堆 (函数模板)
ranges::make_heap (C++20)	从一个元素范围创建出一个最大堆 (niebloid)
push_heap	将一个元素加入到一个最大堆 (函数模板)
ranges::push_heap (C++20)	将一个元素加入到一个最大堆 (niebloid)
pop_heap	从最大堆中移除最大元素 (函数模板)
ranges::pop_heap (C++20)	从最大堆中移除最大元素 (niebloid)
sort_heap	将一个最大堆变成一个按升序排序的元素范围 (函数模板)
ranges::sort_heap (C++20)	将一个最大堆变成一个按升序排序的元素范围 (niebloid)

最小/最大操作

在标头 <algorithm> 定义	
max	返回各给定值中的较大者 (函数模板)
ranges::max (C++20)	返回给定值的较大者 (niebloid)
max_element	返回范围内的最大元素 (函数模板)
ranges::max_element (C++20)	返回范围中的最大元素 (niebloid)
min	返回各给定值中的较小者 (函数模板)
ranges::min (C++20)	返回给定值的较小者 (niebloid)
min_element	返回范围内的最小元素 (函数模板)
ranges::min_element (C++20)	返回范围中的最小元素 (niebloid)
minmax (C++11)	返回两个元素的较小和较大者 (函数模板)
ranges::minmax (C++20)	返回两个元素的较小和较大者 (niebloid)
minmax_element (C++11)	返回范围内的最小元素和最大元素 (函数模板)
ranges::minmax_element (C++20)	返回范围中的最小和最大元素 (niebloid)
clamp (C++17)	在一对边界值间夹逼一个值 (函数模板)
ranges::clamp (C++20)	在一对边界值间夹一个值 (niebloid)

比较操作

在标头 <algorithm> 定义	
equal	确定两个元素集合是否是相同的 (函数模板)
ranges::equal (C++20)	确定两个元素集合是否是相同的 (niebloid)
lexicographical_compare	当一个范围按字典顺序小于另一个范围时，返回 true (函数模板)
ranges::lexicographical_compare (C++20)	当一个范围按字典顺序小于另一个范围时，返回 true (niebloid)
lexicographical_compare_three_way (C++20)	用三路比较比较两个范围 (函数模板)

排列操作

在标头 <code><algorithm></code> 定义	
<code>is_permutation</code> (C++11)	判断一个序列是否为另一个序列的排列 (函数模板)
<code>ranges::is_permutation</code> (C++20)	确定一个序列是否为另一序列的重排 (niebloid)
<code>next_permutation</code>	产生某个元素范围的按字典顺序的下一个较大的排列 (函数模板)
<code>ranges::next_permutation</code> (C++20)	产生某个元素范围的按字典序下一个较大的排列 (niebloid)
<code>prev_permutation</code>	产生某个元素范围的按字典顺序的下一个较小的排列 (函数模板)
<code>ranges::prev_permutation</code> (C++20)	产生某个元素范围的按字典序下一个较小的排列 (niebloid)

数值运算

在标头 <code><numeric></code> 定义	
<code>iota</code> (C++11)	用从起始值开始连续递增的值填充一个范围 (函数模板)
<code>ranges::iota</code> (C++23)	用从起始值开始连续递增的值填充一个范围 (niebloid)
<code>accumulate</code>	对一个范围内的元素求和 (函数模板)
<code>inner_product</code>	计算两个范围的元素的内积 (函数模板)
<code>adjacent_difference</code>	计算范围内各相邻元素之间的差 (函数模板)
<code>partial_sum</code>	计算范围内元素的部分和 (函数模板)
<code>reduce</code> (C++17)	类似 <code>std::accumulate</code> ，但不依序执行 (函数模板)
<code>exclusive_scan</code> (C++17)	类似 <code>std::partial_sum</code> ，第 <i>i</i> 个和中排除第 <i>i</i> 个输入 (函数模板)
<code>inclusive_scan</code> (C++17)	类似 <code>std::partial_sum</code> ，第 <i>i</i> 个和中包含第 <i>i</i> 个输入 (函数模板)
<code>transform_reduce</code> (C++17)	应用一个函数对象，然后以乱序规约 (函数模板)
<code>transform_exclusive_scan</code> (C++17)	应用一个函数对象，然后进行排除扫描 (函数模板)
<code>transform_inclusive_scan</code> (C++17)	应用一个函数对象，然后进行包含扫描 (函数模板)

未初始化内存上的操作

在标头 <code><memory></code> 定义	
<code>uninitialized_copy</code>	将范围内的对象复制到未初始化的内存区域 (函数模板)
<code>ranges::uninitialized_copy</code> (C++20)	复制元素范围到未初始化的内存区域 (niebloid)
<code>uninitialized_copy_n</code> (C++11)	将指定数量的对象复制到未初始化的内存区域 (函数模板)
<code>ranges::uninitialized_copy_n</code> (C++20)	复制定量元素到未初始化的内存区域 (niebloid)
<code>uninitialized_fill</code>	复制一个对象到以范围定义的未初始化内存区域 (函数模板)
<code>ranges::uninitialized_fill</code> (C++20)	复制一个对象到范围所定义的未初始化的内存区域 (niebloid)
<code>uninitialized_fill_n</code>	复制一个对象到以起点和计数定义的未初始化内存区域 (函数模板)
<code>ranges::uninitialized_fill_n</code> (C++20)	复制一个对象到起始与计数所定义的未初始化的内存区域 (niebloid)
<code>uninitialized_move</code> (C++17)	移动一个范围的对象到未初始化的内存区域 (函数模板)

<code>ranges::uninitialized_move</code> (C++20)	移动对象范围到未初始化的内存区域 (niebloid)
<code>uninitialized_move_n</code> (C++17)	移动一定数量对象到未初始化内存区域 (函数模板)
<code>ranges::uninitialized_move_n</code> (C++20)	移动一定量对象到未初始化的内存区域 (niebloid)
<code>uninitialized_default_construct</code> (C++17)	在范围所定义的未初始化的内存区域以默认初始化构造对象 (函数模板)
<code>ranges::uninitialized_default_construct</code> (C++20)	在范围所定义的未初始化的内存区域以默认初始化构造对象 (niebloid)
<code>uninitialized_default_construct_n</code> (C++17)	在起始和计数所定义的未初始化内存区域用默认初始化构造对象 (函数模板)
<code>ranges::uninitialized_default_construct_n</code> (C++20)	在起始与计数所定义的未初始化的内存区域以默认初始化构造对象 (niebloid)
<code>uninitialized_value_construct</code> (C++17)	在范围所定义的未初始化内存中用值初始化构造对象 (函数模板)
<code>ranges::uninitialized_value_construct</code> (C++20)	在范围所定义的未初始化的内存区域以值初始化构造对象 (niebloid)
<code>uninitialized_value_construct_n</code> (C++17)	在起始和计数所定义的未初始化内存区域以值初始化构造对象 (函数模板)
<code>ranges::uninitialized_value_construct_n</code> (C++20)	在起始与计数所定义的未初始化的内存区域以值初始化构造对象 (niebloid)
<code>destroy</code> (C++17)	销毁一个范围中的对象 (函数模板)
<code>ranges::destroy</code> (C++20)	销毁范围中的元素 (niebloid)
<code>destroy_n</code> (C++17)	销毁范围中一定数量的对象 (函数模板)
<code>ranges::destroy_n</code> (C++20)	销毁范围中一定量的元素 (niebloid)
<code>destroy_at</code> (C++17)	销毁在给定地址的对象 (函数模板)
<code>ranges::destroy_at</code> (C++20)	销毁位于给定地址的元素 (niebloid)
<code>construct_at</code> (C++20)	在给定地址创建对象 (函数模板)
<code>ranges::construct_at</code> (C++20)	在给定地址创建对象 (niebloid)

C 库

在标头 <code><cstdlib></code> 定义	
<code>qsort</code>	对未指定类型的元素的一个范围进行排序 (函数)
<code>bsearch</code>	在未指定类型的数组中搜索元素 (函数)

参阅

算法 的 C 文档
来自“<https://zh.cppreference.com/mwiki/index.php?title=cpp/algorithm&oldid=75445>”