library
<algorithm>   ⟨algorithm⟩

Standard Template Library: Algorithms

The header ⟨algorithm⟩ defines a collection of functions especially designed to be used on ranges of elements.

A range is any sequence of objects that can be accessed through iterators or pointers, such as an array or an instance of some of the STL containers. Notice though, that algorithms operate through iterators directly on the values, not affecting in any way the structure of any possible container (it never affects the size or storage allocation of the container).

Functions in <algorithm>

Non-modifying sequence operations:

all_of	Test condition on all elements in range (function template)
any_of	Test if any element in range fulfills condition (function template)
none_of	Test if no elements fulfill condition (function template)
for_each	Apply function to range (function template)
find	Find value in range (function template)
find_if	Find element in range (function template)
find_if_not	Find element in range (negative condition) (function template)
find_end	Find last subsequence in range (function template)
find_first_of	Find element from set in range (function template)
adjacent_find	Find equal adjacent elements in range (function template)
count	Count appearances of value in range (function template)
count_if	Return number of elements in range satisfying condition (function template)
mismatch	Return first position where two ranges differ (function template)
equal	Test whether the elements in two ranges are equal (function template)
is_permutation	Test whether range is permutation of another (function template)
search	Search range for subsequence (function template)
search_n	Search range for elements (function template)

Modifying sequence operations:

copy	Copy range of elements (function template)
copy_n	Copy elements (function template)
copy_if	Copy certain elements of range (function template)
copy_backward	Copy range of elements backward (function template)
move	Move range of elements (function template)
move_backward	Move range of elements backward (function template)
swap	Exchange values of two objects (function template)
swap_ranges	Exchange values of two ranges (function template)
iter_swap	Exchange values of objects pointed to by two iterators (function template)
transform	Transform range (function template)
replace	Replace value in range (function template)
replace_if	Replace values in range (function template)
replace_copy	Copy range replacing value (function template)
replace_copy_if	Copy range replacing value (function template)
fill	Fill range with value (function template)
fill_n	Fill sequence with value (function template)
generate	Generate values for range with function (function template)
generate_n	Generate values for sequence with function (function template)
remove	Remove value from range (function template)
remove_if	Remove elements from range (function template)
remove_copy	Copy range removing value (function template)
remove_copy_if	Copy range removing values (function template)
unique	Remove consecutive duplicates in range (function template)
unique_copy	Copy range removing duplicates (function template)
reverse	Reverse range (function template)
reverse_copy	Copy range reversed (function template)
rotate	Rotate left the elements in range (function template)
rotate_copy	Copy range rotated left (function template)

random_shuffle	Randomly rearrange elements in range (function template)
shuffle	Randomly rearrange elements in range using generator (function template)

Partitions:

is_partitioned	Test whether range is partitioned (function template)
partition	Partition range in two (function template)
stable_partition	Partition range in two - stable ordering (function template)
partition_copy	Partition range into two (function template)
partition_point	Get partition point (function template)

Sorting:

sort	Sort elements in range (function template)
stable_sort	Sort elements preserving order of equivalents (function template)
partial_sort	Partially sort elements in range (function template)
partial_sort_copy	Copy and partially sort range (function template)
is_sorted	Check whether range is sorted (function template)
is_sorted_until	Find first unsorted element in range (function template)
nth_element	Sort element in range (function template)

Binary search (operating on partitioned/sorted ranges):

lower_bound	Return iterator to lower bound (function template)
upper_bound	Return iterator to upper bound (function template)
equal_range	Get subrange of equal elements (function template)
binary_search	Test if value exists in sorted sequence (function template)

Merge (operating on sorted ranges):

merge	Merge sorted ranges (function template)
inplace_merge	Merge consecutive sorted ranges (function template)
includes	Test whether sorted range includes another sorted range (function template)
set_union	Union of two sorted ranges (function template)
set_intersection	Intersection of two sorted ranges (function template)
set_difference	Difference of two sorted ranges (function template)
set_symmetric_difference	Symmetric difference of two sorted ranges (function template)

Heap:

push_heap	Push element into heap range (function template)
pop_heap	Pop element from heap range (function template)
make_heap	Make heap from range (function template)
sort_heap	Sort elements of heap (function template)
is_heap	Test if range is heap (function template)
is_heap_until	Find first element not in heap order (function template)

Min/max:

min	Return the smallest (function template)
max	Return the largest (function template)
minmax	Return smallest and largest elements (function template)
min_element	Return smallest element in range (function template)
max_element	Return largest element in range (function template)
minmax_element	Return smallest and largest elements in range (function template)

Other:

lexicographical_compare	Lexicographical less-than comparison (function template)
next_permutation	Transform range to next permutation (function template)
prev_permutation	Transform range to previous permutation (function template)

Spotted an error? contact us