Functions in <algorithm>

Non-modifying sequence operations:

for_each	Apply function to range
find	Find value in range
find_if	Find element in range
find_end	Find last subsequence in range
find_first_of	Find element from set in range
adjacent_find	Find equal adjacent elements in range
count	Count appearances of value in range
count_if	Number of elements in range satisfying condition
mismatch	Return first position where two ranges differ
equal	Test whether the elements in two ranges are equal
search	Find subsequence in range
search_n	Find succession of equal values in range

Modifying sequence operations:

copy	Copy range of elements
copy_backward	Copy range of elements backwards
swap	Exchange values of two objects
swap_ranges	Exchange values of two ranges
iter_swap	Exchange values of objects pointed by two iterators
transform	Apply function to range
replace	Replace value in range
replace_if	Replace values in range
replace_copy	Copy range replacing value
replace_copy_if	Copy range replacing value
fill	Fill range with value
fill_n	Fill sequence with value
generate	Generate values for range with function
generate_n	Generate values for sequence with function
remove	Remove value from range
remove_if	Remove elements from range
remove_copy	Copy range removing value
remove_copy_if	Copy range removing values
unique	Remove consecutive duplicates in range
unique_copy	Copy range removing duplicates
reverse	Reverse range
reverse_copy	Copy range reversed
rotate	Rotate elements in range
rotate_copy	Copy rotated range
random_shuffle	Rearrange elements in range randomly
partition	Partition range in two
stable_partition	Partition range in two - stable ordering

Sorting:

sort	Sort elements in range
stable_sort	Sort elements preserving order of equivalents
partial_sort	Partially Sort elements in range
partial_sort_copy	Copy and partially sort range
nth_element	Sort element in range

Binary search (operating on sorted ranges):

lower_bound	Return iterator to lower bound
upper_bound	Return iterator to upper bound
equal_range	Get subrange of equal elements
binary_search	Test if value exists in sorted array

Merge (operating on sorted ranges):

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

Heap:

push_heap	Push element into heap range
pop_heap	Pop element from heap range
make_heap	Make heap from range
sort_heap	Sort elements of heap

Min/max:

min	Return the lesser of two arguments
max	Return the greater of two arguments
min_element	Return smallest element in range
max_element	Return largest element in range
lexicographical_compare	Lexicographical less-than comparison
next_permutation	Transform range to next permutation
prev_permutation	Transform range to previous permutation