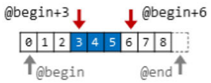
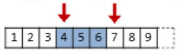
C++ Standard Library Algorithms

Iterator Ranges



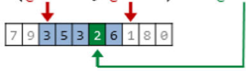
distance(@first, @last) → number_of_elements



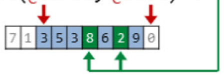
<iterator>

Non-Modifying Sequence Operations <algorithm>

min_element(@first, @last) → @minimum

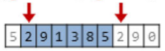


minmax_element(@first, @last) → {@minimum, @maximum}



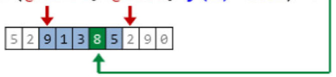
any_of (@first, @last, f(○)↦bool) →
all_of (@first, @last, f(○)↦bool) →
none_of(@first, @last, f(○)↦bool) →

- true, if f yields true for *any*, *all* or *none* elements in the input range
- false otherwise



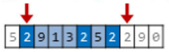
find (@first, @last, value) →
find_if(@first, @last, f(○)↦bool) →

- @1st match
- @last if no match

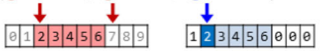


count (@first, @last, value) → number of
count_if(@first, @last, f(○)↦bool) → occurrences

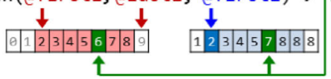
(here: 3 if value = 2)



equal(@first1, @last1, @first2) → true if all elements in both ranges are equal

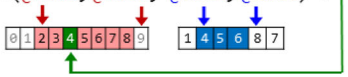


mismatch(@first1,@last1, @first2) → {@mismatch_in_1, @mismatch_in_2}



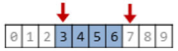
search(@fst1,@lst1, @fst2,@lst2) →

- @1st occurrence of sequence 2 inside sequence 1
- @last1 otherwise



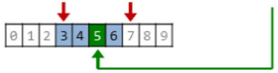
on *sorted* sequences ⇒ $\mathcal{O}(\log n)$

binary_search(@first, @last, value) → true if value found



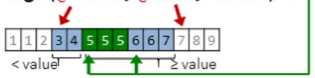
lower_bound(@first, @last, value) →

- position of first element not smaller than value
- @last if not found



equal_range(@first, @last, value) →

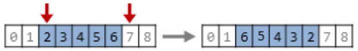
- iterator range of equivalent items
- empty range if not found



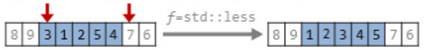
Modifying Sequence Operations <algorithm>

that reorder elements

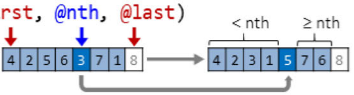
reverse(@first, @last)



sort(@first, @last)
sort(@first, @last, f(○,○)↦bool)

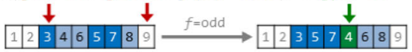


nth_element(@first, @nth, @last)

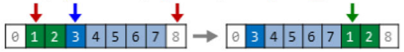


element at nth position → element that would be in that position in a sorted sequence

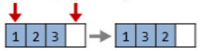
partition(@first, @last, f(○)↦bool) → @part2



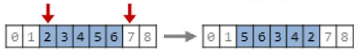
rotate(@first, @newfst, @last) → @old_first



next_permutation(@first, @last) → true if new permutation is lexicographically greater

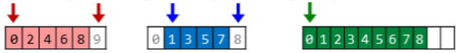


shuffle(@first, @last, random_engine)

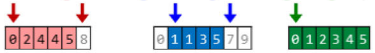


on *sorted* sequences ⇒ $\mathcal{O}(n)$

merge(@fst1,@lst1, @fst2,@lst2, @target)

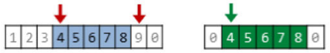


set_union(@fst1, @lst1, @fst2,@lst2, @target)

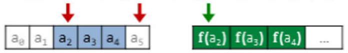


that change values of elements

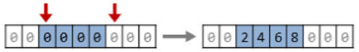
copy(@first, @last, @target)
move(@first, @last, @target)



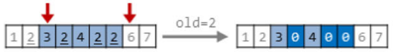
transform(@first, @last, @target, f(○)↦■)



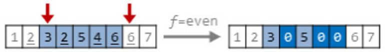
generate(@first, @last, f()↦■)



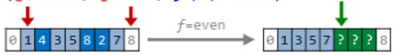
replace(@first, @last, old, new)



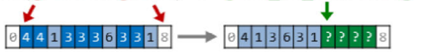
replace_if(@first, @last, f(○)↦bool, new)



remove (@first, @last, value) → @end_of_
remove_if(@first, @last, f(○)↦bool) → remaining

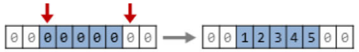


unique(@first, @last) → @end_of_remaining

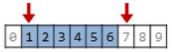


Numeric Algorithms <numeric>

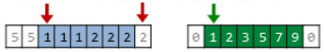
iota(@first, @last, start_value)



accumulate(@first, @last, init_value) → sum

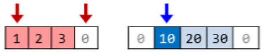


partial_sum(@first, @last, @target)



prefix sums

inner_product(@fst1, @lst1, @fst2, init) → Π



Π = init + 1·10 + 2·20 + 3·30

h/cpp hackingcpp.com

Sequence Queries

all_of (C++11)
any_of (C++11)
none_of (C++11)
count count_if
find find_if
find_if_not (C++11)
find_end
find_first_of
adjacent_find
for_each
for_each_n (C++17)
sample (C++20)
equal
mismatch
search
search_n
lexicographical_compare
compare_3way (C++20)
lexicographical_compare_3way (C++20)

Reordering Elements

reverse reverse_copy
rotate rotate_copy
shift_left shift_right (C++20)
shuffle (C++11)
swap
swap_ranges
iter_swap

Partitioning

is_partitioned
partition
stable_partition
partition_copy
partition_point

Permutations

is_permutation (C++11)
next_permutation prev_permutation

Sorting

sort
stable_sort
partial_sort
partial_sort_copy
is_sorted
is_sorted_until
nth_element

Changing Elements

copy
copy_backward
copy_if (C++11)
copy_n (C++11)
move (C++11)
move_backward (C++11)
fill
fill_n
generate
generate_n
transform
replace replace_if
replace_copy replace_copy_if
remove remove_if
remove_copy remove_copy_if
unique
unique_copy

Binary Search on Sorted Ranges

binary_search
lower_bound upper_bound
equal_range
includes

Merging of Sorted Ranges

merge
inplace_merge
set_union
set_intersection
set_difference
set_symmetric_difference

Heaps

make_heap
sort_heap
push_heap
pop_heap
is_heap (C++11)
is_heap_until (C++11)

Minimum/Maximum

min max
min_element max_element
minmax (C++11)
minmax_element (C++11)
clamp (C++17)

Numeric #include <numeric>

accumulate
adjacent_difference
inner_product
partial_sum
iota (C++11)
reduce (C++17)
exclusive_scan (C++17)
inclusive_scan (C++17)
transform_reduce (C++17)
transform_exclusive_scan (C++17)
transform_inclusive_scan (C++17)

v2.5