<algorithms> library in C++ STL

1. **std :: all_of** : Test condition on all elements in range
2. **std :: any_of** : Test if any element in range fulfills condition
3. **std :: none_of** : Test if no elements fulfill condition
4. std :: for_each : Apply function to range
5. **std :: find** : Find value in range
6. **std :: find_if** : Find element in range
7. **std :: find_if_not** : Find element in range (negative condition)
8. **std :: find_end :** Find last subsequence in range
9. **std :: find_first_of** : Find element from set in range
10. **std :: adjacent_find** : Find equal adjacent elements in range
11. **std :: count** : Count appearances of value in range
12. std :: count_if : Return number of elements in range satisfying condition
13. **std :: mismatch :** Return first position where two ranges differ
14. **std::equal** : Test whether the elements in two ranges are equal
15. **std :: is_permutation :** Test whether range is permutation of another
16. **std :: search :** Search range for subsequence
17. **std :: search_n :** Search range for element

Modifying sequence operations

1. **std :: copy :** Copy range of elements
2. **std :: copy_n :** Copy elements
3. **std :: copy_if :** Copy certain elements of range
4. **std :: copy_backward :** Copy range of elements backward
5. **std::move :** Move range of elements
6. **std :: move_backward :** Move range of elements backward
7. std :: swap : Exchange values of two objects
8. std ::swap_ranges : Exchange values of two ranges
9. **std :: iter_swap :** Exchange values of objects pointed to by two iterators
10. **std ::transform :** Transform range
11. **std ::replace :** Replace value in range
12. **std ::replace_if :** Replace values in range
13. **std :: replace_copy :** Copy range replacing value
14. **std :: replace_copy_if :** Copy range replacing value
15. **std ::fill :** Fill range with value
16. **std :: fill_n :** Fill sequence with value
17. **std ::generate :** Generate values for range with function
18. **std ::generate_n :** Generate values for sequence with function
19. **std ::remove :** Remove value from range

20. **std :: remove_if :** Remove elements from range
21. remove_copy : Copy range removing value
22. remove_copy_if : Copy range removing values
23. **std ::unique :** Remove consecutive duplicates in range
24. **std :: unique_copy :** Copy range removing duplicates
25. **std ::reverse :** Reverse range
26. std :: reverse_copy : Copy range reversed
27. std :: rotate : Rotate left the elements in range
28. std :: rotate_copy : Copy range rotated left
29. std :: random_shuffle : Randomly rearrange elements in range
30. std :: shuffle : Randomly rearrange elements in range using generator

Partition Operations

1. **std :: is_partitioned :** Test whether range is partitioned
2. **std :: partition :** Partition range in two
3. **std :: stable_partition :** Partition range in two – stable ordering
4. partition_copy : Partition range into two
5. partition_point : Get partition point

Sorting

1. **std :: sort :** Sort elements in range
2. **std :: stable_sort :** Sort elements preserving order of equivalents
3. **std :: partial_sort :** Partially sort elements in range
4. **std :: partial_sort_copy :** Copy and partially sort range
5. **std :: is_sorted :** Check whether range is sorted
6. **std :: is_sorted_until :** Find first unsorted element in range
7. **std :: nth_element :** Sort element in range

Binary search (operating on partitioned/sorted ranges)

1. **std :: lower_bound :** Return iterator to lower bound
2. **std :: upper_bound :** Return iterator to upper bound

3. **std :: equal_range :** Get subrange of equal elements
4. **std :: binary_search :** Test if value exists in sorted sequence

Merge (operating on sorted ranges)

1. **std :: merge :** Merge sorted ranges
2. **std :: inplace_merge :** Merge consecutive sorted ranges
3. **std :: includes :** Test whether sorted range includes another sorted range
4. **std :: set_union :** Union of two sorted ranges
5. **std :: set_intersection :** Intersection of two sorted ranges
6. **std :: set_difference :** Difference of two sorted ranges
7. **std :: set_symmetric_difference :** Symmetric difference of two sorted ranges

Heap Operations

1. **std :: push_heap :** Push element into heap range
2. **std :: pop_heap :** Pop element from heap range
3. **std :: make_heap :** Make heap from range
4. **std :: sort_heap :** Sort elements of heap
5. **std :: is_heap :** Test if range is heap
6. **std :: is_heap_until :** Find first element not in heap order
7. **std :: max :** Return the largest
8. **std :: minmax :** Return smallest and largest elements
9. **std :: min_element :** Return smallest element in range
10. **std :: max_element :** Return largest element in range
11. **std :: minmax_element :** Return smallest and largest elements in range

Other Operations

1. **std :: lexicographical_compare :** Lexicographical less-than comparison
2. **std :: next_permutation :** Transform range to next permutation
3. **std :: prev_permutation :** Transform range to previous permutation

All STL articles of C++

Recommended Posts:

<iterator> library in C++ STL
SDL library in C/C++ with examples
<regex> library in C++ STL
<numeric> library in C++ STL
<strings> library in C++ STL
snprintf() in C library
Pattern Searching using C++ library
Any datatype in C++ boost library
boost::split in C++ library
Set in C++ Standard Template Library (STL)
difftime() C library function
Advanced C++ with boost library
The C++ Standard Template Library (STL)
Map in C++ Standard Template Library (STL)
boost::algorithm::none_of_equal() in C++ library

Article Tags : C++ cpp-advanced cpp-algorithm-library cpp-containers-library STL

Practice Tags : STL CPP

14

2.6

☐ To-do ☐ Done

Based on **8** vote(s)

Feedback/ Suggest Improvement Add Notes Improve Article

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments