

Algorithms

Algorithms are a fundamental part of the Standard Template Library. Algorithms do not work with containers themselves but rather with iterators. Therefore, the same algorithm can be used by most if not all of the STL containers. This section discusses the conventions and terminology of the STL algorithms.

Remarks

The descriptions of the algorithm template functions employ several shorthand phrases:

- The phrase "in the range $[A, B)$ " means the sequence of zero or more discrete values beginning with A up to but not including B . A range is valid only if B is reachable from A ; you can store A in an object N ($N = A$), increment the object zero or more times ($++N$), and have the object compare equal to B after a finite number of increments ($N == B$).
- The phrase "each N in the range $[A, B)$ " means that N begins with the value A and is incremented zero or more times until it equals the value B . The case $N == B$ is not in the range.
- The phrase "the lowest value of N in the range $[A, B)$ such that X " means that the condition X is determined for each N in the range $[A, B)$ until the condition X is met.
- The phrase "the highest value of N in the range $[A, B)$ such that X " means that X is determined for each N in the range $[A, B)$. The function stores in K a copy of N each time the condition X is met. If any such store occurs, the function replaces the final value of N , which equals B , with the value of K . For a bidirectional or random-access iterator, however, it can also mean that N begins with the highest value in the range and is decremented over the range until the condition X is met.
- Expressions such as $X - Y$, where X and Y can be iterators other than random-access iterators, are intended in the mathematical sense. The function does not necessarily evaluate $operator-$ if it must determine such a value. The same is also true for expressions such as $X + N$ and $X - N$, where N is an integer type.

Several algorithms make use of a predicate that performs a pairwise comparison, such as with **operator==**, to yield a bool result. The predicate function **operator==**, or any replacement for it, must not alter either of its operands. It must yield the same bool result every time it is evaluated, and it must yield the same result if a copy of either operand is substituted for the operand.

Several algorithms make use of a predicate that must impose a strict weak ordering on pairs of elements from a sequence. For the predicate $pr(X, Y)$:

- Strict means that $pr(X, X)$ is false.

- Weak means that X and Y have an equivalent ordering if $!pr(X, Y) \ \&\& \ !pr(Y, X)$ ($X == Y$ does not need to be defined).
- Ordering means that $pr(X, Y) \ \&\& \ pr(Y, Z)$ implies $pr(X, Z)$.

Some of these algorithms implicitly use the predicate $X < Y$. Other predicates that typically satisfy the strict weak ordering requirement are $X > Y$, **less**(X, Y), and **greater**(X, Y). Note, however, that predicates such as $X \leq Y$ and $X \geq Y$ do not satisfy this requirement.

A sequence of elements designated by iterators in the range $[First, Last)$ is a sequence ordered by **operator<** if, for each N in the range $[0, Last - First)$ and for each M in the range $(N, Last - First)$ the predicate $!(*First + M) < *(First + N)$ is true. (Note that the elements are sorted in ascending order.) The predicate function **operator<**, or any replacement for it, must not alter either of its operands. It must yield the same bool result every time it is evaluated, and it must yield the same result if a copy of either operand is substituted for the operand. Moreover, it must impose a strict weak ordering on the operands it compares.

A sequence of elements designated by iterators in the range $[First, Last)$ is a heap ordered by **operator<** if, for each N in the range $[1, Last - First)$ the predicate $!(*First < *(First + N))$ is true. (The first element is the largest.) Its internal structure is otherwise known only to the template functions `make_heap`, `pop_heap`, and `push_heap`. As with an ordered sequence, the predicate function **operator<**, or any replacement for it, must not alter either of its operands, and it must impose a strict weak ordering on the operands it compares. It must yield the same bool result every time it is evaluated, and it must yield the same result if a copy of either operand is substituted for the operand.

The STL algorithms are located in the `<algorithm>` and `<numeric>` header files.