**一. 概述**

移除性算法是在一区间内移除某些元素，这些算法并**不能改变元素的数量**，它们只是以逻辑上的思考，将原本置于后面的“不移除元素”向前移动，覆盖那些被移除元素而已。它们都返回新区间的逻辑终点。

|  |  |
| --- | --- |
| remove() | 将等于某特定值的元素全部移除 |
| remove\_if() | 将满足某准则的元素全部移除 |
| remove\_copy() | 将不等于某特定值的元素全部复制到它处 |
| remove\_copy()\_if() | 将不满足某准则的元素全部复制到它处 |
| unique() | 移除毗邻的重复元素 |
| unique\_copy() | 移除毗邻的重复元素，并复制到它处 |

**二. 移除某些特定元素**

**1. 移除某序列内的元素**

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/lwbeyond/article/details/7329577)

1. //移除区间[beg, end)中每一个与value相等的元素
2. ForwardIterator
3. remove (ForwardIterator beg, ForwardIterator end,
4. const T& value)
5. //移除区间[beg, end)中每一个令op(elem)为true的元素
6. ForwardIterator
7. remove\_if (ForwardIterator beg, ForwardIterator end,
8. UnaryPredicate op)

//移除区间[beg, end)中每一个与value相等的元素

ForwardIterator

remove (ForwardIterator beg, ForwardIterator end,

const T& value)

//移除区间[beg, end)中每一个令op(elem)为true的元素

ForwardIterator

remove\_if (ForwardIterator beg, ForwardIterator end,

UnaryPredicate op)

**2. 复制时一并移除元素**

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/lwbeyond/article/details/7329577)

1. //将源区间[beg, end)内所有元素复制到以destBeg为起点的目标区间去，
2. //并在复制过程中移除与value相等的所有元素
3. OutputIterator
4. remove\_copy (InputIterator sourceBeg, InputIterator sourceEnd,
5. OutputIterator destBeg,
6. const T& value)
7. //两个算法都返回目标区间中最后一个被复制元素的下一位置
8. OutputIterator
9. remove\_copy\_if (InputIterator sourceBeg, InputIterator sourceEnd,
10. OutputIterator destBeg,
11. UnaryPredicate op)

//将源区间[beg, end)内所有元素复制到以destBeg为起点的目标区间去，

//并在复制过程中移除与value相等的所有元素

OutputIterator

remove\_copy (InputIterator sourceBeg, InputIterator sourceEnd,

OutputIterator destBeg,

const T& value)

//两个算法都返回目标区间中最后一个被复制元素的下一位置

OutputIterator

remove\_copy\_if (InputIterator sourceBeg, InputIterator sourceEnd,

OutputIterator destBeg,

UnaryPredicate op)

**三. 移除重复元素**

**1. 移除连续重复元素**

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/lwbeyond/article/details/7329577)

1. //移除连续重复的元素，源序列必须是已排序的
2. void
3. unique (ForwardIterator beg, ForwardIterator end)
4. void
5. unique (Forwardlterator beg, ForwardIterator end,
6. BinaryPredicate op)

//移除连续重复的元素，源序列必须是已排序的

void

unique (ForwardIterator beg, ForwardIterator end)

void

unique (Forwardlterator beg, ForwardIterator end,

BinaryPredicate op)

**2. 复制过程中移除重复元素**

**[cpp]** [view plaincopyprint?](http://blog.csdn.net/lwbeyond/article/details/7329577)

1. //将源区间[sourceBeg, sourceEnd)内的元素复制到以destBeg起始的目标区间
2. //并移除重复元素
3. OutputIterator
4. unique\_copy (InputIterator sourceBeg, InputIterator sourceEnd,
5. OutputIterator destBeg)
6. //两个算法都返回目标区间内最后一个被复制的元素的下一个位置
7. OutputIterator
8. unique\_copy (InputIterator sourceBeg, InputIterator sourceEnd,
9. OutputIterator destBeg,
10. BinaryPredicate op)