

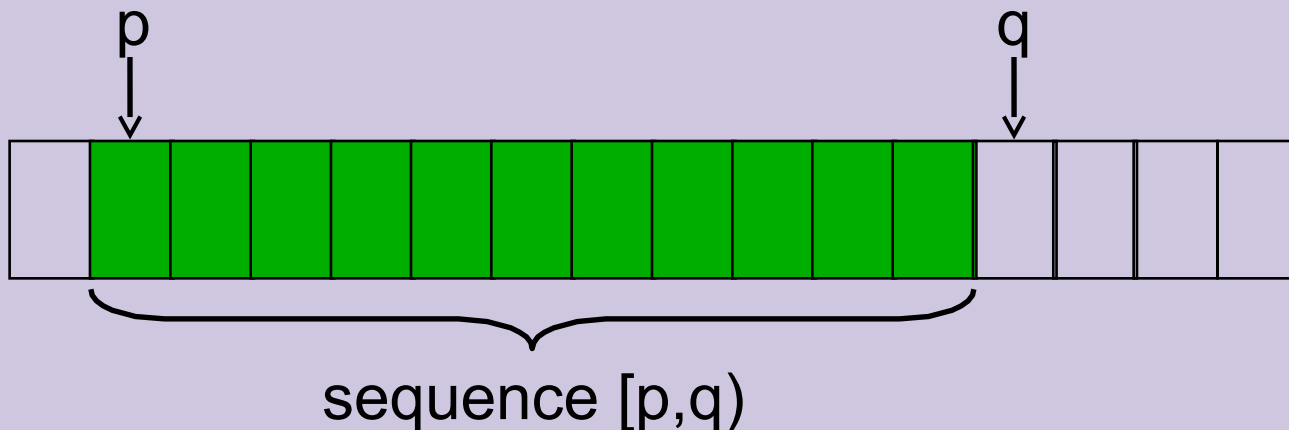
The Standard C++ Library – Algorithms

Version 1: Dr. Ofir Pele

Version 2: Dr. Erel Segal-Halevi

Algorithms

- Most STL algorithms work on sequences
- Sequences are passed as two iterators:
 - beginning element
 - element one after last



- Algorithms depend on iterator type
not on container type

Example – merge documentation

copy

```
template< typename In, typename Out>
Out copy(In first, In last, Out res)
{
    while (first != last)
        *res++ = *first++;
    return res;
}
```

copy

```
template< typename In, typename Out>
Out copy(In first, In last, Out res)
{
    while (first != last)
        *res++ = *first++;
    return res;
}
```

What's wrong with this ?

```
void foo(const vector<char>& v) {
    vector<char> v2;
    ...
    copy(v.begin(), v.end(), v2.begin());
}
```

copy

```
template< typename In, typename Out>
Out copy(In first, In last, Out res)
{
    while (first != last)
        *res++ = *first++;
    return res;
}
```

What's wrong with this ?

```
void foo(const vector<char>& v) {
    vector<char> v2;
    ...
    copy(v.begin(), v.end(), v2.begin());
}
```



OUT OF BOUND

So how can we copy and insert ?

Solution #1: Use insert explicitly

```
void foo(const vector<char>& v) {  
    vector<char> v2;  
    ...  
    v2.insert(v2.end(), v.begin(), v.end());  
}
```

So how can we copy and insert ?

Solution #2: Use `back_inserter`, which returns an iterator that knows to “push_back”. See folder 6.

```
void foo(const vector<char>& v) {  
    vector<char> v2;  
    ...  
    copy(v.begin(), v.end(), back_inserter(v2));  
}
```


generate, bind

generate –

- Accepts two iterators and a function without arguments;
- Fills the space between the iterators by iteratively calling the function.

bind -

- Accepts a function f and an argument x .
- Returns a functor without arguments that returns $f(x)$.

Application example:

- Fill a vector with random numbers (*folder 7*).

sort – using operator <

```
template <class RandomAccessIterator>
void sort(RandomAccessIterator first,
          RandomAccessIterator last);
```

Example usage(the hard way):

```
sort< vector<int>::iterator >
      (vec.begin(), vec.end()) ;
```

sort – using operator <

```
template <class RandomAccessIterator>
void sort(RandomAccessIterator first,
          RandomAccessIterator last);
```

Example usage:

```
sort(vec.begin(), vec.end());
```

sort – using operator <

```
template <class RandomAccessIterator>
void sort(RandomAccessIterator first,
          RandomAccessIterator last);
```

Example usage with primitive arrays:

```
int arr[5];
...
sort(arr, arr+5);
```

sort – using operator <

```
template <class RandomAccessIterator>
void sort(RandomAccessIterator first,
          RandomAccessIterator last);
```

Example usage with primitive arrays (C++11):

```
int arr[5];
...
sort(begin(arr), end(arr));
```

sort – using comparator

```
template <class RandomAccessIterator,  
          class StrictWeakOrdering>  
void sort(RandomAccessIterator first,  
          RandomAccessIterator last,  
          StrictWeakOrdering comp) ;
```

Example usage:

```
sort(vec.begin() , vec.end() , greater<int>() ) ;
```

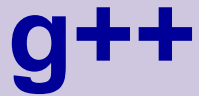
sort – compile error

```
list<int> l(nums, nums+SIZE);  
sort(l.begin(), l.end());
```

sort – compile error

```
list<int> l(nums, nums+SIZE);  
sort(l.begin(), l.end());
```

list iterators are bidirectional and not random access



```
/usr/lib/gcc/i486-linux-gnu/4.1.2/../../../../include/c++/4.1.2/bits/stl_algo.h: In function 'void  
std::sort(_RandomAccessIterator, _RandomAccessIterator) [with _RandomAccessIterator =  
std::_List_iterator<int>]':
```

```
Main.cpp:17: instantiated from here
```

```
/usr/lib/gcc/i486-linux-gnu/4.1.2/../../../../include/c++/4.1.2/bits/stl_algo.h:2713: error: no match for  
'operator-' in '___last - ___first'
```

```
/usr/lib/gcc/i486-linux-gnu/4.1.2/../../../../include/c++/4.1.2/bits/stl_bvector.h:182: note: candidates are:  
ptrdiff_t std::operator-(const std::_Bit_iterator_base&, const std::_Bit_iterator_base&)
```

```
/usr/lib/gcc/i486-linux-gnu/4.1.2/../../../../include/c++/4.1.2/bits/stl_algo.h: In function 'void  
std::__final_insertion_sort(_RandomAccessIterator, _RandomAccessIterator) [with  
_RandomAccessIterator = std::_List_iterator<int>]':
```

```
/usr/lib/gcc/i486-linux-gnu/4.1.2/../../../../include/c++/4.1.2/bits/stl_algo.h:2714: instantiated from 'void  
std::sort(_RandomAccessIterator, _RandomAccessIterator) [with _RandomAccessIterator =  
std::_List_iterator<int>]'
```

```
Main.cpp:17: instantiated from here
```

```
/usr/lib/gcc/i486-linux-gnu/4.1.2/../../../../include/c++/4.1.2/bits/stl_algo.h:2357: error: no match for  
'operator-' in '___last - ___first'
```

g++

```
/usr/lib/gcc/i486-linux-gnu/4.1.2/../../../../include/c++/4.1.2/bits/stl_algo.h: In function 'void  
std::sort(_RandomAccessIterator, _RandomAccessIterator) [with _RandomAccessIterator =  
std::_List_iterator<int>]':
```

```
Main.cpp:17: instantiated from here
```

```
/usr/lib/gcc/i486-linux-gnu/4.1.2/../../../../include/c++/4.1.2/bits/stl_algo.h:2713: error: no match for  
'operator-' in '___last
```

```
/usr/lib/gcc/i486-linux-gnu/4.1.2/../../../../include/c++/4.1.2/bits/stl_algo.h:2713: note: candidates are:  
ptrdiff_t std::operator
```

```
/usr/lib/gcc/i486-linux-gnu/4.1.2/../../../../include/c++/4.1.2/bits/stl_algo.h:2713: note: candidates are:  
std::__final_insertion  
_RandomAccessIter
```

???

```
82: note: candidates are:  
ator_base&)  
unction 'void  
ator) [with
```

```
/usr/lib/gcc/i486-linux-gnu/4.1.2/../../../../include/c++/4.1.2/bits/stl_algo.h:2714: instantiated from 'void  
std::sort(_RandomAccessIterator, _RandomAccessIterator) [with _RandomAccessIterator =  
std::_List_iterator<int>]'
```

```
Main.cpp:17: instantiated from here
```

```
/usr/lib/gcc/i486-linux-gnu/4.1.2/../../../../include/c++/4.1.2/bits/stl_algo.h:2357: error: no match for  
'operator-' in '___last - ___first'
```

g++

```
/usr/lib/gcc/i486-linux-  
gnu/4.1.2/../../../../include/c+  
+/4.1.2/bits/stl_algo.h: In function 'void  
std::sort(_RandomAccessIterator,  
_RandomAccessIterator) [with  
_RandomAccessIterator =  
std::_List_iterator<int>]':
```

Main.cpp:17: instantiated from here

```
/usr/lib/gcc/i486-linux-gnu/4.1.2/../../../../include/c++/4.1.2/bits/stl_algo.h:2713: error: no match for  
'operator-' in '___last - ___first'
```

...

g++ -D_GLIBCXX_CONCEPT_CHECKS and STLfilt

BD Software STL Message Decryptor v2.47a for gcc

stl_algo.h: In function 'void sort(_List_iterator<int>, _List_iterator<int>)':

Main.cpp:17: instantiated from here

stl_algo.h:2713: error: no match for 'operator-' in '__last - __first'

stl_algo.h: In function 'void __final_insertion_sort(_List_iterator<int>, _List_iterator<int>)':

stl_algo.h:2714: instantiated from 'void sort(_List_iterator<int>, _List_iterator<int>)'

Main.cpp:17: instantiated from here

...

g++ -D_GLIBCXX_CONCEPT_CHECKS and STL Filt

...

Main.cpp:17: instantiated from here

boost_concept_check.h:223: error: **conversion
from ‘**

**bidirectional_iterator_tag’ to non-scalar
type ‘**

random_access_iterator_tag’ requested

Cryptic error messages

STLFilt:

An STL Error Message Decryptor for C++:

<http://www.bdsoft.com/tools/stlfilt.html>

Cryptic error messages

Different compilers:

clang++ (free)

intel c++ (not free)

More?

- Lots of other features, especially in c++11 (threads,...)
- Other libraries:
 - Boost
 - opencv, dlib, armadillo, zlib, ...