STL, Sorting Large Amounts of Data

Chapter 7

C++ Templates (Java: Generics)

Compile time **polymorphism**

```
template <typename Type> // could be <class classname>
Type max(Type a, Type b)
{
    Type result = b;
    if( a > b )
    {
       result = a;
    }
    return result;
}

int main()
{
  std::cout << max<double>(3, 7.0) << std::endl;
  return 0;
}
```

https://en.wikipedia.org/wiki/Template_(programming)

Standard Template Library

- Four important parts
  - Containers Functional
  - Iterators Algorithms

http://en.cppreference.com/w/cpp

http://www.cplusplus.com/references/stl

http://www.sgi.com/tech/stl/table_of_contents.html

http://gcc.gnu.org/onlinedocs/libstdc++ *chapter 9*

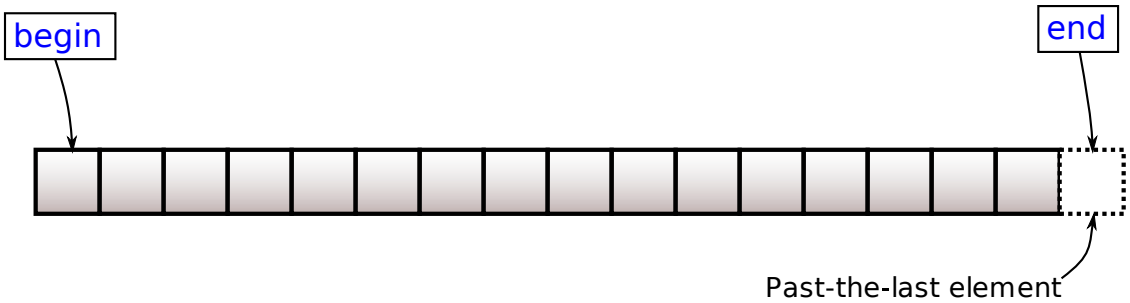
http://msdn.microsoft.com/en-us/library

Containers

- Associative
  - set
  - multiset
  - map
  - multimap
- Sequential
  - vector
  - dequeue
  - list

Iterators

- Iterators are objects that behave like cursors
- Used to access items stored in containers
- Each STL container object provides two member functions:
  - .begin()
  - .end()



http://en.cppreference.com/w/cpp/iterator/end

Using Iterators

- To define an iterator object for vectors:

```
vector<int>::iterator iter;
```

Example

```
#include <iostream>
#include <vector>  // needed to use vectors

int main()
{
   std::vector<int> vect;   // Create a vector of int
   for (int x = 0; x < 10; x++)
   {
     vect.push_back(x*x);
   }

   //print everything using iterators.
   std::vector<int>::iterator iter = vect.begin();
   while (iter != vect.end()) // !!!
   {
     std::cout <<  *iter << " ";
     iter ++;
   }
   return 0;
}
```

Algorithms in STL

- Many algorithms including:
  - binary_search
  - count
  - for_each
  - find
  - max_element
  - min_element
  - random_shuffle
  - sort

Example

```
#include <vector>
#include <iostream>
#include <algorithm>

int main()
{
    std::vector<int> v({1,2,3});
    std::reverse(std::begin(v), std::end(v));
    std::cout << v[0] << v[1] << v[2] << '\n';

    int a[] = {4, 5, 6, 7};
    std::reverse(&a[0], &a[4]);
    std::cout << a[0] << a[1] << a[2] << a[3] << '\n';
}
```

http://en.cppreference.com/w/cpp/algorithm/reverse

Project

- Disk Scheduling

Project

4.3 Program Design Example: Rating the Field

Pretty Polly has no shortage of gentlemen suitors who come a' courting. Indeed, her biggest problem is keeping track of who the best ones are. She is smart enough to realize that a program which ranks the men from most to least desirable would simplify her life. She is also persuasive enough to have talked you into writing the program.

Polly really likes to dance, and has determined the optimal partner height is 180 centimeters tall. Her first criteria is finding someone who is as close as possible to this height; whether they are a little taller or shorter doesn't matter. Among all candidates of the same height, she wants someone as close as possible to 75 kilograms without going over. If all equal-height candidates are over this limit, she will take the lightest of the bunch. If two or more people are identical by all these characteristics, sort them by last name, then by first name if it is necessary to break the tie.

Polly is only interested in seeing the candidates ranked by name, so the input file:

```
George Bush          195    110
Harry Truman         180    75
Bill Clinton         180    75
John Kennedy         180    65
Ronald Reagan        165    110
Richard Nixon        170    70
Jimmy Carter         180    77
```

yields the following output:

```
Clinton, Bill
Truman, Harry
Kennedy, John
Carter, Jimmy
Nixon, Richard
Bush, George
Reagan, Ronald
```

Problem of the Day

- The *nuts and bolts problem* is defined as follows. You are given a collection of *n* bolts of different widths, and n corresponding nuts.
- You can test whether a given nut and bolt go together, from which you learn whether the nut is too large, too small, or an exact match for the bolt.
- The differences in size between pairs of nuts or bolts can be too small to see by eye, so you cannot rely on comparing the sizes of two nuts or two bolts directly.
- You are to match each bolt to each nut.

Problem of the Day

1. Give an $O(n^2)$ algorithm to solve the nuts and bolts problem.
2. Suppose that instead of matching all of the nuts and bolts, you wish to find the smallest bolt and its corresponding nut. Show that this can be done in only 2n - 2 comparisons.
3. Match the nuts and bolts in expected O(n lg n) time.