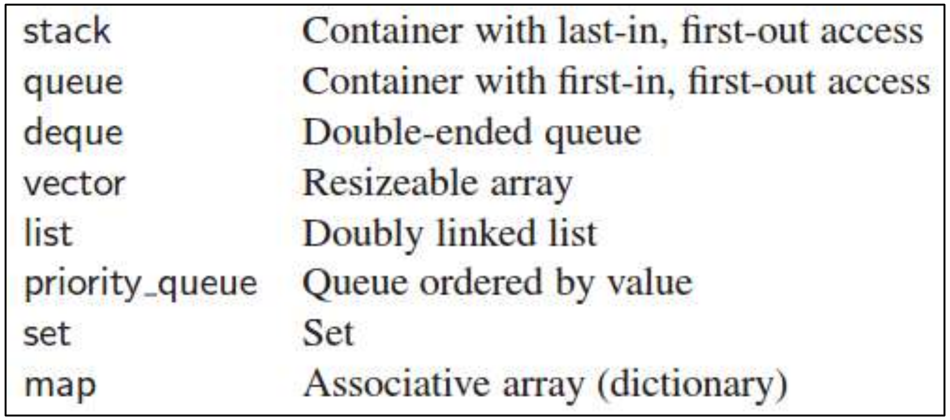
Standard Template Library: Vectors

Jyh-Shing Roger Jang (張智星)
CSIE Dept, National Taiwan University

Intro. to Standard Template Library

- Standard Template Library (STL)
  - A collection of useful classes for common data structures
- STL provides data structures for standard containers



stack	Container with last-in, first-out access
queue	Container with first-in, first-out access
deque	Double-ended queue
vector	Resizeable array
list	Doubly linked list
priority_queue	Queue ordered by value
set	Set
map	Associative array (dictionary)

- Each type of STL can store objects of any kinds.
- FAQ for STL
  - http://cs.smu.ca/~porter/csc/ref/stl/faq.html

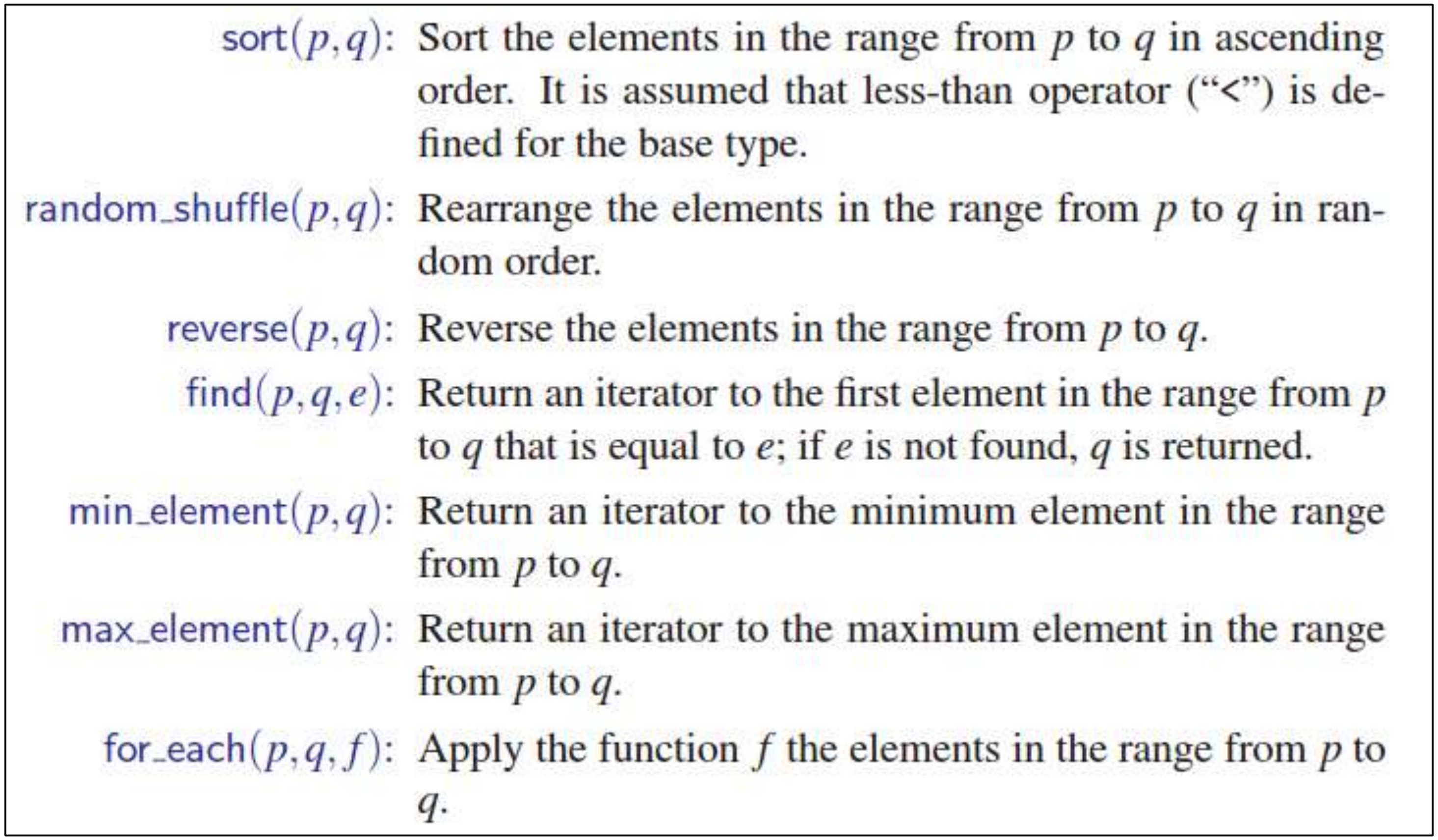
Strength and Weakness of STL vectors

Quiz!

- Advantages of STL vectors (over standard C/C++ arrays)
  - Flexible element access
    - vec[i] ➔ No range check, but more efficient
    - vec.at(i) ➔ With range check
  - Dynamic growth of arrays
    - Memory are automatic allocated (and reallocated)
  - Less likely to have memory leak
    - No need to delete/free memory explicitly
  - Built-in methods for common array operations
- Disadvantages of STL vectors
  - Not as efficient as standard C/C++ arrays
- Comprehensive comparison
  - http://cs.smu.ca/~porter/csc/ref/stl/tutorial_intro.html

STL Vectors and Algorithms

- #include <algorithm>



sort(p,q):	Sort the elements in the range from p to q in ascending order. It is assumed that less-than operator ("<") is defined for the base type.
random_shuffle(p,q):	Rearrange the elements in the range from p to q in random order.
reverse(p,q):	Reverse the elements in the range from p to q.
find(p,q,e):	Return an iterator to the first element in the range from p to q that is equal to e; if e is not found, q is returned.
min_element(p,q):	Return an iterator to the minimum element in the range from p to q.
max_element(p,q):	Return an iterator to the maximum element in the range from p to q.
for_each(p,q,f):	Apply the function f the elements in the range from p to q.

Examples of STL Vectors

- Some example of STL vectors is here:
  - http://mirlab.org/jang/courses/dsa/example
- Memory of STL vectors is allocated implicitly
  - You can reserve a vector of size n by “x.reserve(n)”.
  - You can keep on pushing back to go beyond n.
  - Once it go explodes, a new size of k*n is allocated implicitly.

Compiler dependent!

- Quiz: Given an STL vector x…
  - What does "x.reserve(25)" mean?
  - What is the difference between x[i] and x.at(i)?
  - What is the difference between x.size() and x.capacity()?

Quiz!

Resources & References

- Member functions of STL vectors
  - http://www.cplusplus.com/reference/vector/vector/
- Algorithms that can be used for STL vectors
  - http://www.cplusplus.com/reference/algorithm
  - http://en.cppreference.com/w/cpp/algorithm
- A comprehensive site for STL
  - http://cs.smu.ca/~porter/csc/ref/stl/

Check the list before you go!
Don't reinvent the wheel!