STL Odds and Ends

Jim Fawcett

CSE687 – Object Oriented Design

Spring 2002

Removing Element Values

- To remove all values, t, from a vector, v, use:

```
v.erase(remove(v.begin(),v.end(),t),v.end());
```

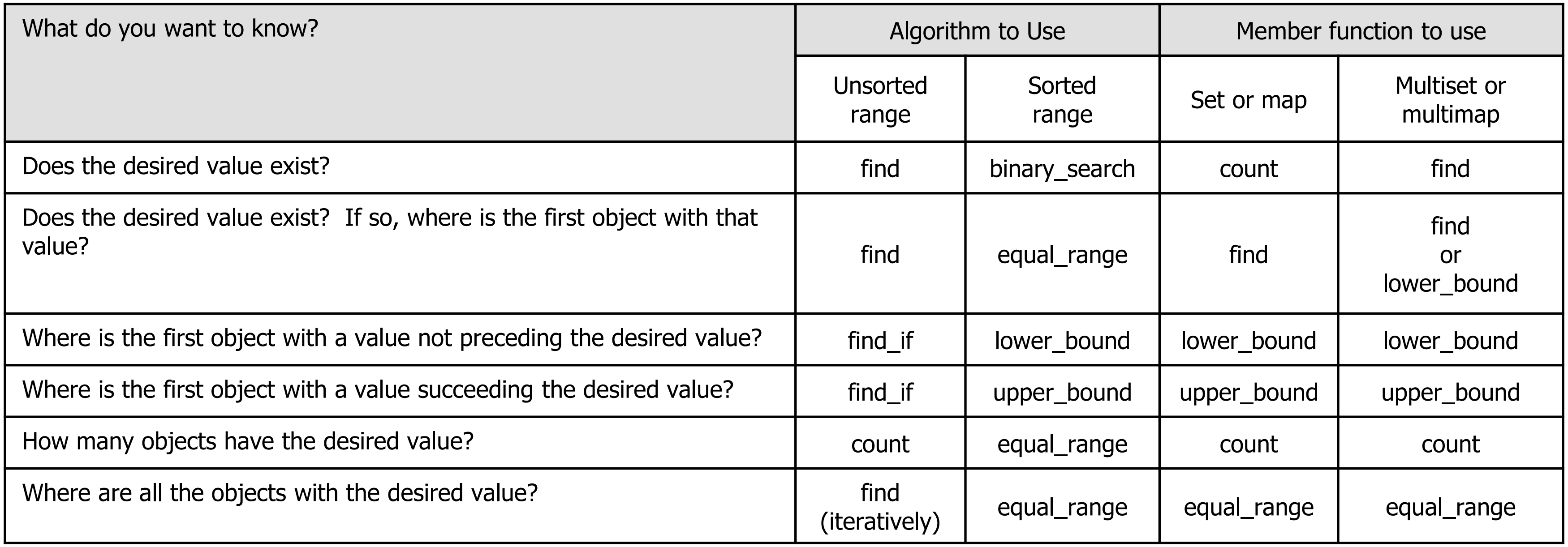
- To remove all values, t, from a list, l, use:

```
l.remove(t);      // note: this remove is a member function
```

- To remove all values, t, from an associative container, c, use:

```
c.erase(t);       // remove corrupts an associative container
```

Finding and Counting Values



What do you want to know?	Algorithm to Use		Member function to use	
	Unsorted range	Sorted range	Set or map	Multiset or multimap
Does the desired value exist?	find	binary_search	count	find
Does the desired value exist? If so, where is the first object with that value?	find	equal_range	find	find or lower_bound
Where is the first object with a value not preceding the desired value?	find_if	lower_bound	lower_bound	lower_bound
Where is the first object with a value succeeding the desired value?	find_if	upper_bound	upper_bound	upper_bound
How many objects have the desired value?	count	equal_range	count	count
Where are all the objects with the desired value?	find (iteratively)	equal_range	equal_range	equal_range

Effective STL, Scott Meyers, Addison Wesley, 2001

Functor Notes

- STL Algorithms pass functors by value, so all your functor designs should properly support copying and assignment, perhaps by simply allowing compiler to generate those operations.
- Predicate functors should be pure functions, e.g., they always return a boolean result as a function of their argument(s) only, e.g., internal state plays no role in determining the return value.

  The usual way to do this is simply to avoid giving predicate functors any state or declare them const.

Code Examples

- Inserter – demonstrates use of transform algorithm using an inserter iterator.
- Sorting – demonstrates use of sort, find_if, upper_bound, and lower_bound