Bounds and Ranges

This topic teaches the bounds and range algorithms by demonstrating the usage including lower_bound, upper_bound and equal_range and explaining the algorithms

OPKINS | Engineering for Professionals

• Figure 16.11 demonstrates functions lower_bound, upper_bound and equal_range.

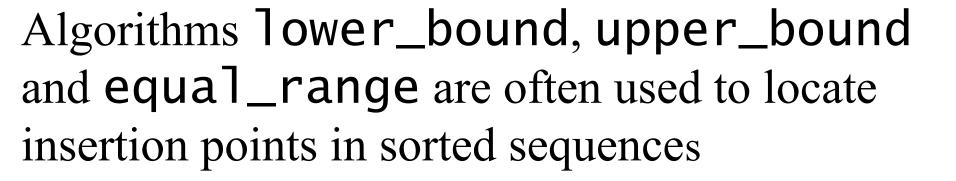
- auto lower = lower_bound
 (a.cbegin(), a.cend(), 6);
- Uses the <u>lower_bound</u> algorithm to find the first location in a sorted sequence of values at which the third argument could be inserted in the sequence such that the sequence would still be *sorted in ascending order*
- The first two iterator arguments must be at least forward iterators
- The third argument is the value for which to determine the lower bound

- The algorithm returns a *forward iterator* pointing to the position at which the insert can occur
- A second version of lower_bound takes as a fourth argument a binary predicate function indicating the order in which the elements were originally sorted

auto upper = upper bound (a.cbegin(), a.cend(), 6);

- Uses the upper bound algorithm to find the last location in a sorted sequence of values at which the third argument could be inserted in the sequence such that the sequence would still be sorted in ascending order
- The first two iterator arguments must be at least forward iterators
- The third argument is the value for which to determine the upper bound
- The algorithm returns a *forward iterator* pointing to the position at which the insert can occur
- A second version of upper_bound takes as a fourth argument a binary predicate function indicating the order in which the elements were originally sorted

- auto eq = equal_range
 (a.cbegin(), a.cend(), 6);
- Uses the equal_range algorithm to return a pair of *forward iterators* containing the results of performing both a lower_bound and an upper_bound operation
- The first two arguments must be at least *forward iterators*
- The third is the value for which to locate the equal range
- Returns a pair of *forward iterators* for the lower bound (eq.first) and upper bound (eq.second), respectively



- lower = lower_bound(a.cbegin(),a.cend(), 5);
 - Uses lower_bound to locate the first point at which 5 can be inserted in order in a

```
upper = upper_bound(a.cbegin(), a.cend(), 7);
```

- Uses upper_bound to locate the last point at which 7 can be inserted in order in a
- eq = equal_range(a.cbegin(), a.cend(), 5);
 - Uses equal_range to locate the first and last points at which 5 can be inserted in order in a

This topic taught the bounds and range algorithms by demonstrating the usage including lower_bound, upper_bound and equal_range and explaining the algorithms