

Topic 7. «Set and Map Iterators»



How to declare?

```
set <char> T; // set declaration  
set <char> :: iterator sit; // iterator  
                        // declaration  
sit = T.begin(); // iterator initialization
```

```
map <int, int> A; // map declaration  
map <int, int> :: iterator it; // iterator  
                        // declaration  
it = A.begin(); // iterator initialization
```

How to step through set?

```
set <int> A; // still declaration

for (set <int> :: iterator it = A.begin();
     it != A.end(); it++)
{
    int value = *it; // dereferencing the
                    // iterator
}
```

How to step through map?

```
map <int, int> A; // still declaration

for (map <int, int> :: iterator it = A.begin();
     it != A.end(); it++)
{
    int key = (*it).first; // obtaining
                        // the key value
    int value = (*it).second; // obtaining
                            // the mapped value
}
```

Why do I need iterators?

```
set <int> A; // still declaration
int x = 100500; // some integer value

set <int> :: iterator it = A.find(x);
// iterator to the specified value

if (it == A.end())
    cout << ":" (" << endl;
// do something if there is no such value
```

Why do I need iterators?

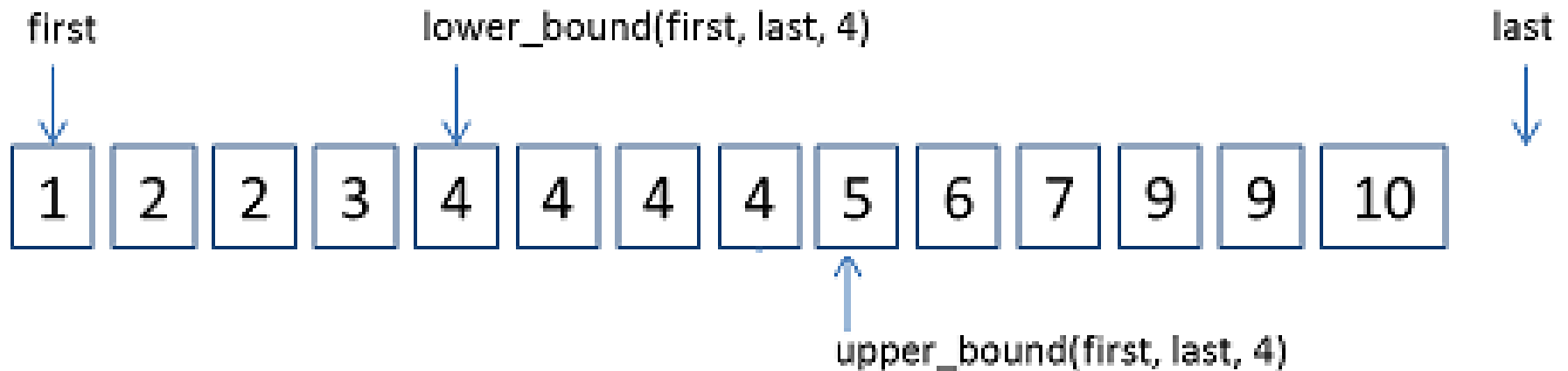
```
set <int> :: iterator it = A.lower_bound(x);  
// iterator to the maximum value that is not  
less than the specified value
```

```
set <int> :: iterator it2 = A.upper_bound(x);  
// iterator to the minimum value that is  
greater than the specified value
```

```
if (it == A.end() || it2 == A.end())  
    cout << "O_o" << endl;  
// do something if there is no such value
```

Lower and Upper Bounds

- Defined not only for sets and maps:



For those who still don't love iterators

```
set <int> A; // still declaration
int x = 100500; // some integer value

set <int> :: iterator it = A.find(x);
// iterator to the specified value

A.erase(it);
// erasing by iterator works faster than by
value
```


Questions?

