

Task 3 Containers

Sequence container

- What is C++ STL(Standard Template Library)? Briefly introduce three STL containers that you know of.
- `vector` is the most commonly used container in the STL. Please learn about `vector` independently and outline its differences from C++ arrays.
- What is an `iterator`? How do you access a `vector` using an `iterator`? How do you iterate through a `vector` using an `iterator`?
- Learn how to initialize and access `vector`, as well as common `vector` methods and functions, to complete the following example question.
 - Implement a function that takes a `vector` of n positive integers as input. The function should first remove all odd numbers from the `vector`. Then, for each remaining even number, the function should do two things: keep the original number in the `vector`, and also add into the `vector` the result of dividing that even number by 2. Finally, the function should return the `vector` sorted in ascending order.
- Self-study other sequence containers (such as `deque` and `priority_queue`), choose one for introduction.

Associative container

- Learn about two types of associative containers: `set` and `map`. What data structures are used to implement them, and why not simply use linear arrays to maintain these containers?
- Introduce the characteristics of the `set` in the STL. What is the relationship between these characteristics and sets in mathematics?
- How do you iterate through a map using an `iterator`, and what are the differences between iterators for maps and vectors?
- Learn how to initialize and access `map` and `set`, as well as common methods and functions for `map` and `set`. Use a `set` and a `map` respectively to complete the following example questions.
 - Given n positive integers, remove duplicates and keep only the first occurrence of each number.

Sample Input

11

1 2 18 3 3 19 2 3 6 5 4

Sample Output

1 2 18 3 19 6 5 4

- Given n strings (strings containing numbers, lowercase and uppercase letters, case sensitive), find out how many unique strings there are among the n strings. Then, output these n strings along with their frequencies of occurrence.

Sample Input

```
5
abc
aaaa
abc
abcc
12345
```

Sample Output

```
4 //only four
unique strings
abc 2 //There are 2
abc among n strings
aaaa 1
abc 2
abcc 1
12345 1
```

- In the STL, there are also a series of "unordered sequential containers," such as `unordered_set` and `unordered_map`. What are they? What functionalities do they provide? Are they implemented using red-black trees? What are the differences in performance and functionality compared to `set` and `map`?