

Введение

Вы уже сталкивались с алгоритмами поиска и методами контейнеров `find` и `count`. Алгоритм `find` и его родственники `find_if` и `find_if_not` ищут элемент в контейнере. Алгоритмы `count` и `count_if` возвращают количество элементов, равных переданному параметру, и количество элементов, для которых выполняется некое условие.

Когда у контейнера есть собственные методы поиска, лучше использовать их, а не общий алгоритм. Например, при поиске в словаре или множестве метод `find` будет эффективнее, чем алгоритм `find`:

CPP

```
set<int> some_set;
for (int i = 0; i < 100000; ++i) {
    some_set.insert(i);
}

// Результаты этих двух поисков будут одинаковы,
// но первый будет сделан более эффективно.
auto it_more_efficient = some_set.find(50000);
auto it = find(some_set.begin(), some_set.end(), 50000);

assert(it_more_efficient == it);
```

Простой поиск элемента, проверка его существования и подсчёт элементов — не единственные алгоритмы поиска. В этой теме вы научитесь:

- искать первый элемент, равный данному или больший,
- искать первый элемент, больший, чем данный,
- делать всё это в отсортированном и неотсортированном векторе, строке, в словаре и множестве.

Чтобы систематизировать знания, вы будете отмечать методы и функции в таблице.