目录

[一、容器 1](#_Toc201234977)

[二、函数 1](#_Toc201234978)

[2.1查找函数 1](#_Toc201234979)

[2.1.1 顺序查找函数 1](#_Toc201234980)

[2.1.2计数相关 2](#_Toc201234981)

[2.1.3区域查找 2](#_Toc201234982)

[2.1.4邻近元素查找 2](#_Toc201234983)

[2.1.5二分查找系列函数（适用于升序排列容器） 2](#_Toc201234984)

# 一、容器

# 二、函数

## 2.1查找函数

### 2.1.1 顺序查找函数

**🔹 find**

功能：顺序查找第一个等于指定值的元素迭代器。

std::find(vec.begin(), vec.end(), 5); // 时间复杂度：O(n)

**🔹 find\_if**

功能：查找第一个使得一元谓词为 true 的元素迭代器。

std::find\_if(vec.begin(), vec.end(), [](int x){ return x % 2 == 0; });

**🔹 find\_if\_not**

功能：查找第一个使得一元谓词为 false 的元素迭代器。

std::find\_if\_not(vec.begin(), vec.end(), [](int x){ return x < 10; });

### 2.1.2计数相关

**🔹 count**

功能：统计容器中等于指定值的元素个数。

int cnt = std::count(vec.begin(), vec.end(), 3);

**🔹 count\_if**

功能：统计容器中满足谓词的元素个数。

int even = std::count\_if(vec.begin(), vec.end(), [](int x){ return x % 2 == 0; });

### 2.1.3区域查找

**🔹 search**

功能：查找连续子序列在范围中的首次出现。

std::search(v1.begin(), v1.end(), v2.begin(), v2.end());

**🔹 find\_end**

功能：查找连续子序列在范围中的最后一次出现。

std::find\_end(v1.begin(), v1.end(), v2.begin(), v2.end());

### 2.1.4邻近元素查找

**🔹 adjacent\_find**

功能：查找第一个相邻的两个相等元素。

std::adjacent\_find(vec.begin(), vec.end());

### 2.1.5二分查找系列函数（适用于升序排列容器）

🔹 **binary\_search**

功能：判断某个值是否存在（返回 true 或 false）

示例：

std::binary\_search(vec.begin(), vec.end(), 5); // true or false

🔹 **lower\_bound**

功能：返回第一个 大于等于 value 的元素迭代器

示例：

auto it = std::lower\_bound(vec.begin(), vec.end(), 5);// \*it >= 5

🔹 **upper\_bound**

功能：返回第一个 大于 value 的元素迭代器

示例：

auto it = std::upper\_bound(vec.begin(), vec.end(), 5);// \*it > 5

🔹 **equal\_range**

功能：返回一个 pair（lower\_bound, upper\_bound）

auto range = std::equal\_range(vec.begin(), vec.end(), 5);

示例：

for (auto it = range.first; it != range.second; ++it)

std::cout << \*it << " "; // 输出所有等于5的元素

🔄 等价于：

std::make\_pair(

std::lower\_bound(first, last, value),

std::upper\_bound(first, last, value)

);