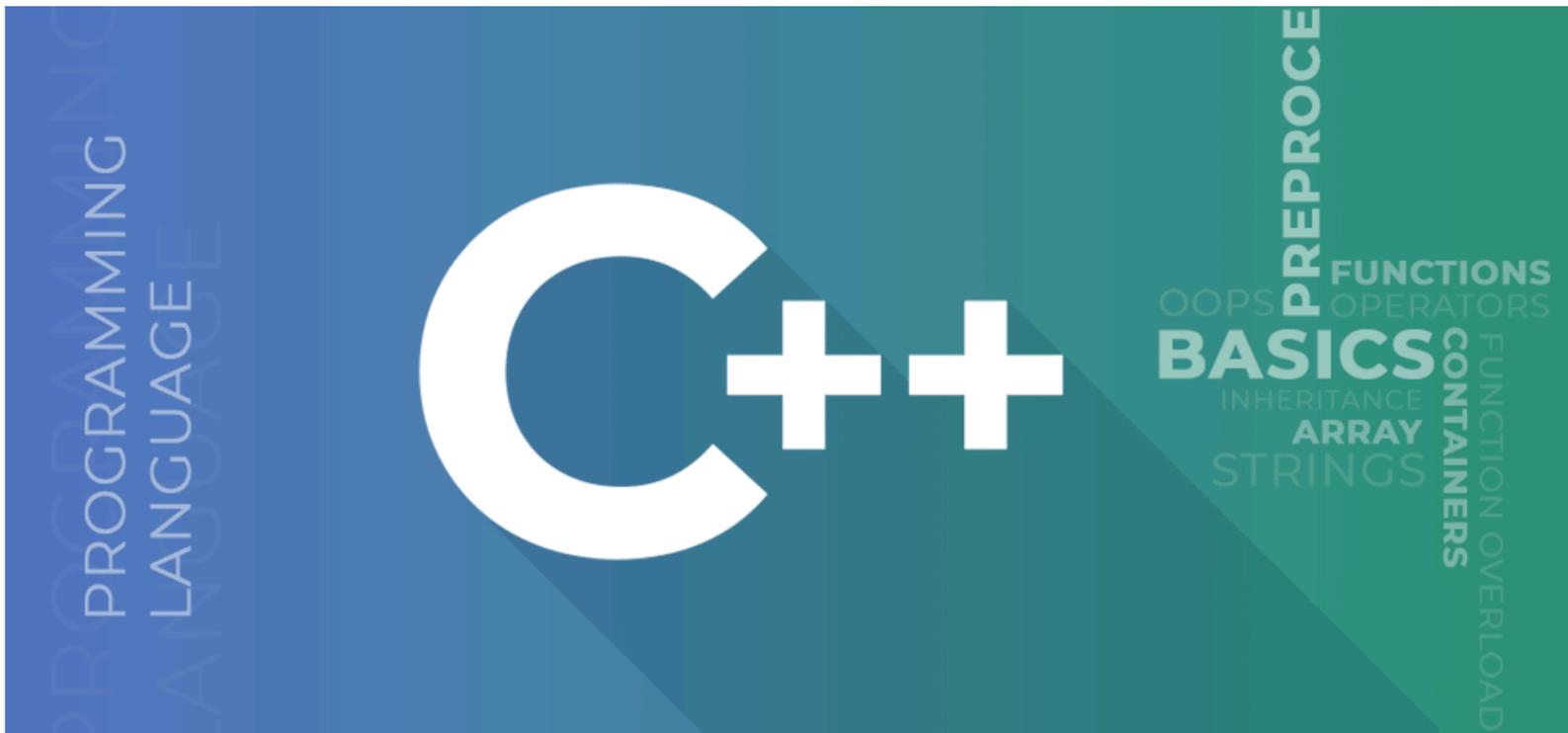


C++矢量操作技巧：擦除重复项、排序轻松搞定，让你的代码更高效！

作者：微信公众号：【架构师老卢】

1-9 8:24

446



**概述：**在C++中，通过使用`std::unique`和`std::sort`算法，可以高效擦除矢量中的重复项并进行排序。先使用`std::sort`排序，再利用`std::unique`擦除相邻的重复元素，是处理矢量操作的常用方法。

在C++中，擦除重复项和对矢量进行排序的有效方法分别是使用`std::unique`和`std::sort`算法。`std::unique`用于在序列中移除相邻的重复元素，而`std::sort`用于对序列进行排序。

## 方法

1. **擦除重复项：**
  - 使用`std::sort`对矢量排序。
  - 使用`std::unique`擦除相邻的重复项。
2. **对矢量进行排序：**
  - 使用`std::sort`对矢量进行排序。

## 步骤

1. **擦除重复项：**
  - 使用`std::sort`对矢量排序。
  - 使用`std::unique`擦除相邻的重复项。
2. **对矢量进行排序：**
  - 使用`std::sort`对矢量进行排序。

## 实例源代码

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4
5 int main() {
6     // 示例矢量
7     std::vector<int> myVector = {3, 1, 2, 2, 4, 3, 5, 4, 6, 6};
8
9     // 擦除重复项
10    std::sort(myVector.begin(), myVector.end()); // 先排序
11    myVector.erase(std::unique(myVector.begin(), myVector.end()), myVector.end());
12
13    // 打印擦除重复项后的矢量
14    std::cout << "擦除重复项后的矢量: ";
15    for (int num : myVector) {
16        std::cout << num << " ";
17    }
18
19    // 对矢量进行排序
20    std::sort(myVector.begin(), myVector.end());
21
22    // 打印排序后的矢量
23    std::cout << "\n排序后的矢量: ";
24    for (int num : myVector) {
25        std::cout << num << " ";
26    }
27
28    return 0;
29 }
```

## 注意事项

1. **擦除重复项顺序：**在使用`std::unique`前需要先将矢量进行排序。
2. **排序算法选择：**`std::sort`使用快速排序，对于小型矢量可以考虑使用`std::sort`，对于大型矢量可考虑其他排序算法。
3. **擦除重复项后的容量：**`std::unique`擦除重复项后，矢量的容量不会改变，可以使用`std::vector<int>(myVector.begin(), it)`构造一个新矢量来释放多余内存。

在C++中，使用`std::unique`和`std::sort`是擦除重复项和对矢量进行排序的有效方法。确保在使用`std::unique`前对矢量进行排序，根据需求选择合适的排序算法，同时注意擦除重复项后的矢量容量。