




07、从有序顺序表中删除其值重复的元素，使表中所有元素值不同

24计算机考研成员一战成硕！

 题目描述：
从有序(升序)顺序表中删除所有其值重复的元素，使表中所有元素的值均不同

 倒计时

1、知识点及难度



解人：多动症男孩

难度：简单

知识点：

1. 在删除重复元素的过程中，需要确保顺序表是有序的。只有在有序的情况下，才能正确判断元素是否重复，并进行删除操作。另外，删除操作需要注意顺序表的长度的更新，以确保只保留了不重复的元素。
2. 这里并没有采用顺序表中“删除位置为i的元素”这样一个基本操作，而是通过if条件中的赋值操作实现将不重复元素同过时间复杂度为 $O(1)$ 的赋值操作实现前移，最后遍历完成的时候重复元素都在表尾，所以只需要更改表长即可实现删除操作。
3. 这里length使用指针类型，是因为删除操作需要更新表长。

时间复杂度： $O(n)$ ，其中n为顺序表中元素的个数。由于算法需要遍历整个顺序表一次，时间复杂度为线性级别。

空间复杂度： $O(1)$ ，即常数级别。算法并没有使用额外的辅助空间。

2、算法题

思路

1. 本题依旧采用双指针法，定义两个指针，一个指向当前元素，一个指向下一个元素。
2. 遍历顺序表，比较两个指针所指元素的值。
3. 如果当前元素的值等于下一个元素的值，则将下一个元素删除，如果不相等则将两指针同步移动到下一个位置。
4. 重复步骤2、3,直到遍历完这个链表。

基本实现-C

```
1 #include <stdio.h>
2 void removeDuplicates(int* sequence, int* length) {
3     if(sequence == NULL || length == 0) {
4         return -1; // 返回一个错误标识
5     }
6
7     int currentIndex = 0; //初始时当前指针指向第一个元素
8
9     for (int i = 1; i < *length; i++) {
10         // 将不重复元素移动到顺序表前端
11         if (sequence[i] != sequence[currentIndex]) {
12             currentIndex++;
13             sequence[currentIndex] = sequence[i];
14         }
15     }
16
17     // 更新顺序表的长度
18     *length = currentIndex + 1; // 这里currentIndex是数组下标
19 }
20
```

基本实现C++

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 void removeDuplicates(vector<int>& sequence) {
6     if (sequence.empty()) {
7         return -1; // 返回一个错误标识
8     }
9
10    int currentIndex = 0; //初始时当前指针指向第一个元素
11
12    for (int i = 1; i < sequence.size(); i++) {
13        if (sequence[i] != sequence[currentIndex]) {
14            currentIndex++;
15            sequence[currentIndex] = sequence[i];
16        }
17    }
18
19    // 更新顺序表的长度
```

```
20     sequence.resize(currentIndex + 1);
21 }
```

3、总结



总结栏

蓝蓝B站首页: [蓝蓝希望你上岸呀B站首页](#)

蓝蓝公众号: [算法训练营9分计划](#)

蓝蓝知识星球介绍: [📖 关于知识星球的权益](#)

如何在星球打卡记录:

- 计算机考研数据结构算法专项day[1/60]:
- 学习内容: 最好能发出自己写的图片
- 遇到的问题: 如果无就不用写了
- 小结: 这部分一周写一次即可。