



## 09、将两个有序顺序表合并为一个新的有序顺序表

24计算机考研成员一战成硕！



题目描述：

将两个有序(升序)顺序表合并为一个新的有序顺序表，并由函数返回结果顺序表



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. 本题是采用辅助空间的方式，也就是空间换时间，顺序表的扩容实际很耗费时间。
2. 在 `mergedSequence[k++] = sequence2[i++]`；中，`k` 和 `i` 都会在赋值操作之后各自加1，以指向下一个位置。

**时间复杂度：** $O(m + n)$ ，其中 $m$ 和 $n$ 分别为两个有序顺序表的长度。因为需要遍历两个顺序表中的元素。

**空间复杂度：** $O(m + n)$ ，需要创建一个新的顺序表来存储合并后的结果，该顺序表的长度为两个输入顺序表的长度之和。

### 2、算法题

思路

1. 创建一个新的顺序表，用于存储合并后的结果。

2. 定义两个指针分别指向两个有序顺序表的起始位置。
3. 比较两个指针所指元素的值，将较小的元素插入到新的顺序表中，并移动指针到下一个位置。
4. 重复步骤3，直到其中一个顺序表的元素全部插入到新的顺序表中。
5. 将另一个顺序表中剩余的元素依次插入到新的顺序表中。
6. 返回合并后的新顺序表作为结果。

#### 基本实现-C

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int* mergerSortedSequence(int* sequence1, int length1, int* sequence2, int length2, int* mergedLength) {
5     int* mergedSequence = (int*)malloc((length1 + length2) * sizeof(int)); // 合并后的新顺序表
6     int i = 0, j = 0, k = 0; // 分别为两个有序顺序表和新的合并顺序表的指针
7
8     // 合并两个有序顺序表
9     while (i < length1 && j < length2) {
10         if (sequence1[i] <= sequence2[j]) {
11             mergedSequence[k++] = sequence1[i++];
12         } else {
13             mergedSequence[k++] = sequence2[j++];
14         }
15     }
16
17     // 将剩余未插入的元素添加到新的顺序表中
18     while (i < length1) {
19         mergedSequence[k++] = sequence1[i++];
20     }
21
22     while (j < length2) {
23         mergedSequence[k++] = sequence2[j++];
24     }
25
26     *mergedLength = k; // 更新合并后的顺序表的长度
27
28     return mergedSequence;
29 }
```

#### 基本实现C++

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 vector<int> mergeSortedLists(vector<int>& sequence1, vector<int>& sequence2) {
6     vector<int> mergedSequence; // 合并后的新顺序表
7     int i = 0; // 表1的指针
8     int j = 0; // 表2的指针
9
10    // 合并两个有序顺序表
11    while (i < sequence1.size() && j < sequence2.size()) {
12        if (sequence1[i] <= sequence2[j]) {
13            mergedSequence.push_back(sequence1[i]);
14            i++;
15        } else {
```

```
16         mergedList.push_back(sequence2[j]);
17         j++;
18     }
19 }
20
21 // 将剩余未插入的元素添加到新顺序表中
22 while (i < sequence1.size()) {
23     mergedList.push_back(sequence1[i]);
24     i++;
25 }
26
27 while (j < sequence2.size()) {
28     mergedList.push_back(sequence2[j]);
29     j++;
30 }
31
32 return mergedList;
33 }
```

### 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。