



05、将顺序表L中小于表头元素放前半部分，大于表头放后半部分

24计算机考研成员一战成硕！

💡 题目描述：

有个存放整数类型的顺序表L，设计算法将L中所有小于等于表头元素的整数放在前半部分，大于表头元素的整数放在后半部分

🕒 倒计时

1、知识点及难度



解人：多动症男孩

难度：简单

知识点 & 注意点：

1. 交换元素时，右指针指向的元素要小于等于表头元素，左指针指向的元素要大于表头元素。这样可以保证交换后的顺序表仍然满足题目要求：将小于等于表头元素的整数放在前半部分，大于表头元素的整数放在后半部分。
2. 外层循环中 $left \leq right$ 的条件，其中=是为了保证循环结束的时候right指向左边最后一个小于表头值的元素，方便最后的时候一个把表头元素放到合适的位置（即左边都是小于它的元素，右边都是大于它的元素）

时间复杂度： $O(n)$ ，其中n为顺序表中元素的个数。因为我们需要遍历整个顺序表，并且只进行值的交换操作。

空间复杂度： $O(1)$ ，即常数级别。算法并没有使用额外的数据结构来存储元素。

2、算法题

思路

1. 定义两个指针，一个指针从表头开始向后遍历，找到大于表头元素的整数；另一个指针从表尾开始向前遍历，找到小于等于表头元素的整数。
2. 在遍历的过程中，如果左指针指向的元素大于表头元素，右指针指向的元素时小于等于表头元素的指针，则交换两个指针指向的元素（这里只需要交换值即可）。
3. 重复步骤2,直到两个指针相遇（此时表示整个顺序表已经遍历完毕）。

基本实现-C

```
1 #include<stdio.h>
2
3 void partition(int* sequence,int length) {
4     if (sequence == NULL || length == 0) {
5         return -1; // 错误返回
6     }
7
8     int pivot = sequence[0]; // 表头元素的值作为比较的枢轴
9     int left = 1; // 左指针初始指向第二个元素(索引)
10    int right = length - 1 // 右指针初始指向最后一个元素 (索引)
11
12    while(left <= right) {
13        // 左指针指向大于pivot的元素
14        while (sequence[left] <= pivot) {
15            left++;
16        }
17        // 右指针指向小于pivot的元素
18        while (sequence[right] > pivot) {
19            right--;
20        }
21        if(left < right) {
22            // 交换元素
23            int tmp = sequence[left]; // 中间变量
24            sequence[left] = sequence[right];
25            sequence[right] = tmp;
26            left++;
27            right--;
28        }
29    }
30
31    //将表头元素放到合适的位置 (即左边都是小于它的元素，右边都是大于它的元素)
32    int temp = sequence[0];
33    sequence[0] = sequence[right];
34    sequence[right] = temp;
35 }
```

基本实现C++

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 void partition(vector<int>& sequence) {
6     if (sequence.empty()) {
7         return;
8     }
```

```
9
10     int pivot = sequence[0]; // 表头元素
11     int left = 1; // 左指针从第二个元素开始(索引)
12     int right = sequence.size() - 1; // 右指针从最后一个元素开始(索引)
13
14     while (left <= right) {
15         while (left <= right && sequence[left] <= pivot) {
16             left++;
17         }
18         while (left <= right && sequence[right] > pivot) {
19             right--;
20         }
21         if (left < right) {
22             swap(sequence[left], sequence[right]); // 交换元素
23             left++;
24             right--;
25         }
26     }
27     swap(sequence[0], sequence[right]); // 将表头元素放到合适的位置
28 }
29
```

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。