



20、求得中位数

24计算机考研成员一战成硕！



题目描述：

【2011统考真题】一个长度为 $L(L \geq 1)$ 的升序序列 S ，处在第 $\lfloor L/2 \rfloor$ 个位置的数称为 S 的中位数。例如，若序列 $S(1)=(11, 13, 15, 17, 19)$ ，则 $S(1)$ 的中位数是15，两个序列的中位数是含它们所有元素的升序序列的中位数。例如，若 $S(2)=(2, 4, 6, 8, 20)$ ，则 $S(1)$ 和 $S(2)$ 的中位数是11。现在有两个**等长升序**序列 A 和 B ，试设计一个在**时间和空间**两方面都尽可能高效的算法，找出两个序列 A 和 B 的中位数。要求：

- 1) 给出算法的基本设计思想。
- 2) 根据设计思想，采用C或C++或Java语言描述算法，关键之处给出注释。
- 3) 说明你所设计算法的时间复杂度和空间复杂度。

i	2	4	6	8	20
---	---	---	---	---	----

j	11	13	15	17	19
---	----	----	----	----	----

2	4	6	8	11	13	15	17	19	20
---	---	---	---	----	----	----	----	----	----

相关内容



算法题解

思路

- 做本题最简单直接的一种想法大概就是将两个升序序列合并成一个长度为 $2L$ 的升序序列，中位数就是第 L 个元素。但是这样的暴力解法时间和空间复杂度较高，所以本题我在暴力解法的基础上做一些**优化**降低时间和空间复杂度。
- 其实我们没有必要把两个长度为 L 的升序序列合并，因为我们只需要确定前 L 个元素的位置即可，后 L 个元素则不需要考虑，基于此我们可以使用双指针的方式，两个指针分别指向两个数组当前最小值，每次比较指针所指元素的值，更小的一方指针向前移动一步，并且计数器加1，当计数器的值等于 L 的时候说明我们已经找到了中位数。

具体步骤：

1. 循环确定前 L 个元素的位置，其中每次指针所指元素更小的一方，让其指针向前移动一步，同时计数器加1。
2. 当计数器的值等于 L 的时候就可以返回中位数。
3. 如果循环结束仍然没有返回中位数，则返回0，属于未知错误

```
1 int Search(int arr1[], int arr2[], int size) {
2     int median = 0, count = 0; // 中位数, 计数器
3     int idx1 = 0, idx2 = 0;
```

```

4 // 确定升序序列中第n位元素
5 while (idx1 < size && idx2 < size) {
6     if (arr1[idx1] < arr2[idx2]) {
7         if (++count == size) {
8             median = arr1[idx1];
9         }
10        idx1++; // 指针加1
11    } else {
12        if (++count == size) {
13            median = arr2[idx2];
14        }
15        idx2++; // 指针加1
16    }
17    if (count >= size) {
18        return median; // 优化循环次数，一旦确定了第n位元素，立即返回
19    }
20 }
21 return 0; // 未知错误，确保健壮性
22 }

```

```

1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 // 从两个升序序列中找出中位数
6 int Search(vector<int> arr1 , vector<int> arr2) {
7     int median = 0 , cout = 0; //中位数 ,计数器
8     int idx1 = 0 , idx2 = 0 , l = arr1.size(); // 分别为arr1和arr2设置指针,数组长度
9     // 确定升序序列中第n位元素
10    while( idx1 < l && idx2 < l ) {
11        if( arr1[idx1] < arr2[idx2] ) {
12            // 注意这个为什么可以这样写++count，当然推荐你这样分开写count++;
13            // 这里我只是为了代码更加简洁!!!
14            if( ++cout == l ) median = arr1[idx1];
15            idx1++; // 指针加1
16        }
17        else {
18            if( ++cout == l ) median = arr2[idx2];
19            idx2++; // 指针加1
20        }
21        if( cout >= l ) return median; // 优化循环次数，一旦确定了第n位元素，立即返回
22    }
23    return 0; //未知错误，确保健壮性

```

```
24 }
25
26 void main() {
27     //测试以下
28     //手动定义两个数组
29     vector<int> arr1 = { 2, 4, 13, 16, 20 } , arr2 = { 11, 13, 15, 17 ,19 };
30     int res = Search(arr1 , arr2);
31     cout << "中位数为：" << res;
32 }
```



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。