

## 个人资料



SURE\_bd

访问: 187962次

积分: 2756

等级: [BLOG > 5](#)

排名: 第7873名

原创: 77篇

转载: 125篇

译文: 1篇

评论: 53条

## 文章搜索

## 文章分类

[C/C++基础知识 \(35\)](#)[Linux \(23\)](#)[TC/OJ/ACM \(7\)](#)[字符串 \(4\)](#)[嵌入式 \(5\)](#)[数据结构 \(5\)](#)[面试题目精选 \(45\)](#)[数学 \(4\)](#)[网络&编程 \(3\)](#)[面试题转载消化系列 \(57\)](#)[娱乐下 \(2\)](#)[职业规划等 \(3\)](#)[C++Primer \(yang\) \(12\)](#)[操作系统 \(1\)](#)[Hadoop \(1\)](#)

## 文章存档

[2014年12月 \(3\)](#)[2014年03月 \(1\)](#)[2012年11月 \(4\)](#)[Bitbucket 让 pull request 变得更强大](#), 可即刻提升团队代码质量 [云计算行业圆桌论坛](#) [前端精品课程免费看](#), 写课评赢心动大礼!

## 两个有序数组的中位数

标签: [random](#) [c](#) [测试](#)

2011-09-22 21:59

6459人阅读

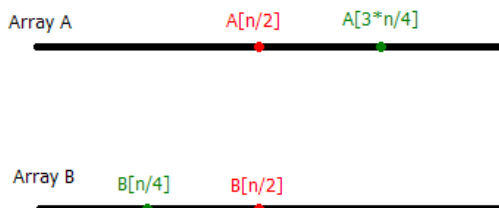
[评论\(7\)](#)[收藏](#)[举报](#)[分类:](#) [面试题目精选 \(44\)](#)[目录\(?\)](#)[\[+\]](#)<http://blog.csdn.net/hhygcy/article/details/4584064>

## 2个有序数组求合并后的中位数

第一步: 假设两个有序数组 (已经各自排序完成了) 长度相等, 试写函数找出两个数组合并后的中位数。第二步: 假设两个有序数组长度不等, 一样的求出中位数

解析: 这个题目看起来非常简单。第一题的话: 假设数组长度为 $n$ , 那么我就把数组1和数组2直接合并, 然后再直接找到中间元素。对于这样的方案, 第一题和第一题就没有什么区别了。这样的话时间复杂度就是 $O(n)$ 。通常在这样的情况下, 那些mentor类型的达人就会循循善诱道: “你还有更好的办法吗?” 如果比线性更高效, 直接能想到的就是对数 $O(\log(n))$ , 这个时间复杂度在这里可能吗? 当然还是可能的。来继续看看下面的分析。

先找来了一个图(自己画的, 简陋了点)



我们先来分析看看: 想到对数的效率, 首先想到的就是二分查找, 对于这个题目二分查找的意义在哪里呢?

我们找到了 $A[n/2]$  和  $B[n/2]$ 来比较,

如果他们相等, 那样的话, 我们的搜索结束了, 因为答案已经找到了 $A[n/2]$ 就肯定是排序后的中位数了。

如果我们发现 $B[n/2] > A[n/2]$ , 说明什么, 这个数字应该在  $A[n/2] \rightarrow A[n]$ 这个序列里面, 或者在  $B[1] \rightarrow B[n/4]$ 这里面。或者, 这里的或者是很重要的, 我们可以说, 我们已经成功的把问题变成了在排序完成的数组 $A[n/2] \rightarrow A[n]$ 和 $B[0] \rightarrow B[n/2]$ 里面找到合并以后的中位数, 显然递归是个不错的选择了。

类似的, 如果 $B[n/2] < A[n/2]$ 呢? 显然就是在 $A[0] \rightarrow A[n/2]$ 和 $B[n/2] \rightarrow B[n]$ 里面寻找了。

在继续想, 这个递归什么时候收敛呢? 当然一个case就是相等的值出现, 如果不出现等到这个 $n==1$ 的时候也就结束了。

照着这样的思路, 我们比较容易写出如下的代码, 当然边界的值需要自己思量一下, 前面的想法只是想法而已。

马上有人说那不定长的怎么办呢? 一样的, 我们还是来画个图看看: (我的画图水平肯定提高了)

```
[cpp] view plain copy print ?
01. int find_median_equal_length( int a[], int b[], int length)
02. {
03.     if (length == 1)
04.         return a[0];
```

2012年10月 (2)

2012年08月 (4)

展开

#### 阅读排行

转载: c++内存泄露机制 (7678)

两个有序数组的中位数 (6453)

二叉树最近公共父亲节点 (4952)

浮点数的表示方法 (4763)

leal (4553)

Linux中的task\_struct和P (4269)

100个数, 范围是0(1)~9! (4032)

腾讯面试题: 海量数据中 (3645)

bitset (2982)

单词反转 (2949)

#### 评论排行

2路插入排序 (13)

100个数, 范围是0(1)~9! (8)

两个有序数组的中位数 (7)

腾讯面试题: 海量数据中 (6)

浮点数的表示方法 (5)

盛大笔试: 两个有序数组 (2)

Using auto\_ptr Effectivel (2)

N个元素取出最大(小)的K (1)

计算字符串的相似度(编程 (1)

二叉树最近公共父亲节点 (1)

#### 推荐文章

\*Networking Named Content 全文翻译

\* 边缘检测与图像分割

\* 数据库性能优化之SQL语句优化

\* 阿里巴巴发布《2015移动安全漏洞年报》

\* Java经典设计模式之七大结构型模式 (附实例和详解)

\* 网络性能评价方法

#### 最新评论

2路插入排序

up\_destiny: 要是  
一定要用插入排序法, 推荐使用  
二分插入排序。如下: void  
b\_inse...

2路插入排序

up\_destiny: @up\_destiny: //把已  
经排序好的两个归并到一起 int k  
= 0; int start...

2路插入排序

up\_destiny: @up\_destiny: 例如  
这样: void two\_insert\_sort(int  
array[],...

2路插入排序

up\_destiny: @up\_destiny: 所  
以, 在2路排序完毕之后, 应该使  
用归并法将两个数据列归并在一  
起就能应付各种场...

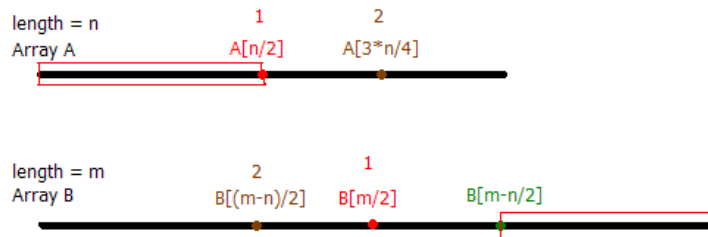
2路插入排序

up\_destiny: 这个算法是有些错误  
的, 为什么这么说。且看我举个  
例子: 3,6,5,3,4,9,4,10,1,2这是  
个...

浮点数的表示方法

chuiwei: 小数部分有更轻松的计  
算方法吗

```
05.
06.     int i = length/2;
07.     if (a[i] == b[i])
08.         return a[i];
09.     else if (a[i]<b[i])
10.         return find_median_equal_length( &a[i], &b[0], length-i );
11.     else
12.         return find_median_equal_length( &a[0], &b[i], length-i );
13. }
```



一样的, 我们还是把这个两个数组来比较一下, 不失一般性, 我们假定B数组比A数组长一点。A的长度为n, B的长度为m。比较A[n/2]和B[m/2]时候。类似的, 我们还是分成几种情况来讨论:

a. 如果A[n/2] == B[m/2], 那么很显然, 我们的讨论结束了。A[n/2]就已经是中位数, 这个和他们各自的长度是奇数或者偶数无关。

b. 如果A[n/2] < B[m/2], 那么, 我们可以知道这个中位数肯定不在[A[0], A[n/2]]这个区间内, 同时也不在[B[m/2], B[m]]这个区间里面。这个时候, 我们不能冲动地把[A[0], A[n/2]]和[B[m/2], B[m]]全部扔掉。我们只需要把[B[m-n/2], B[m]]和[A[0], A[n/2]]扔掉就可以了。(如图所示的红色框框), 这样我们就把我们的问题成功转换成了如何在A[n/2]->A[n]这个长度为n/2的数组和B[1]-B[m-n/2]这个长度为m-n/2的数组里面找中位数了。问题复杂度即可下降了。

c. 只剩下A[n/2] > B[m/2], 和b类似的, 我们可以把A[n/2]->A[n]这块以及B[1]-B[n/2]这块扔掉了就行, 然后继续递归。

我们也可以写下如下的代码:

```
[cpp] view plain copy print ?
01. int find_median_random_length( int a[], int lengtha, int b[], int lengthb)
02. {
03.
04.     int ma = lengtha/2;
05.     int nb = lengthb/2;
06.     int l = ma <= nb ? ma : nb;
07.     if (lengtha == 1)
08.     {
09.         if (lengthb%2==0)
10.         {
11.             if (a[0] >= b[nb])
12.                 return b[nb];
13.             else if (a[0]<=b[nb-1])
14.                 return b[nb-1];
15.             return a[0];
16.         }
17.         else
18.             return b[nb];
19.     }
20.     else if (lengtha == 1)
21.     {
22.         if (lengthb%2==0)
23.         {
24.             if (b[0] >= a[ma])
25.                 return a[ma];
26.             else if (b[0]<=a[ma-1])
27.                 return a[ma-1];
28.             return b[0];
29.         }
30.         else
31.             return a[ma];
32.     }
33.     if ( a[ma] == b[nb] )
34.         return a[ma];
35.     else if ( a[ma] < b[nb] )
36.         return find_median_random_length(&a[ma], lengtha-1, &b[0], lengthb-1);
```

100个数，范围是0(1)~99，其中qq\_31729843: 最前边两种方法，i<size 为什么没有定义一个size的值呢？

浮点数的表示方法

Dablelv: 说的挺好的，但是没有涉及到移码。

两个有序数组的中位数

Alps1992: 楼主你图画的有问题，在B数组里面，第2步的时候，位置是(m/2 - n/4) 不是(m-n)/2

浮点数的表示方法

ixiaoluosiding: 楼主，指数部分是2^7-1再加移位的

```
37.         else
38.             return find_median_random_length(&a[0],lengtha-1,&b[nb],lengthb-1);
39.     }
```

在一些特定的case下面测试了一下，结果还是正确的，下面是用的testcase

```
[cpp] view plain copy print ?
01. int _tmain(int argc, _TCHAR* argv[])
02. {
03.
04.     int a[] = {1,2,3,6,8} ;
05.     int b[] = {6,7,8,9,10};
06.     std::cout<<"median for equal length is : "
<<find_median_equal_length(a, b, sizeof(a)/sizeof(a[0]))<<std::endl; ;
07.     int c[] = {1,3,5,7,9,11} ;
08.     int d[] = {2,4,6,8,10,12};
09.
10.     std::cout<<"median for equal length is : "
<<find_median_equal_length(c, d, sizeof(c)/sizeof(c[0]))<<std::endl;;
11.
12.
13.     int A[]={1,3,5,7,8,9,10};
14.     int B[]={2,4,6,10,11,12,13,14,17,19,20};
15.     int sizeA = sizeof(A)/sizeof(int);
16.     int sizeB = sizeof(B)/sizeof(int);
17.
18.     std::cout<<"median : "<<find_median_random_length(A,sizeA,B,sizeB)<<std::endl;;
19.
20.     int C[] = {1, 2, 3, 4, 5, 6, 7};
21.     int D[] = {5, 6, 7, 8, 9};
22.     std::cout<<"median : "
<<find_median_random_length(C,sizeof(C)/sizeof(C[0]),D,sizeof(D)/sizeof(D[0]))
<<std::endl;;
23.     system("pause");
24.     return 0;
25. }
```

顶 踩  
0 0

[上一篇 倒排索引](#)

[下一篇 字典树算法](#)

我的同类文章

#### 面试题目精选（44）

- |                    |            |         |                               |            |         |
|--------------------|------------|---------|-------------------------------|------------|---------|
| • 三扇门的问题           | 2012-10-25 | 阅读 2743 | • 猴子分桃                        | 2012-10-14 | 阅读 475  |
| • 输入一个正数 n，输出所有... | 2012-08-12 | 阅读 1927 | • C++ STL vector: sizeof(v... | 2012-08-12 | 阅读 2398 |
| • 并查集              | 2011-10-12 | 阅读 353  | • 一个大小为N的数组，里面...             | 2011-10-12 | 阅读 1629 |
| • 楼层扔鸡蛋问题          | 2011-10-08 | 阅读 596  | • 用最少的并行交换完成排序                | 2011-10-08 | 阅读 271  |
| • 生成全排列            | 2011-10-01 | 阅读 968  | • 收集的面试题                      | 2011-09-30 | 阅读 792  |

[更多文章](#)

#### 猜你在找

[性能测试之Apache性能调优](#)

[性能测试之tomcat性能调优](#)

C/C++单元测试培训  
ArcGIS for JavaScript  
测试环境搭建实战集合



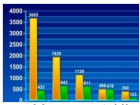
奥迪a9



成都七中



孵化机



计算机学院排



日语学习



oa系统



阿里巴巴

[查看评论](#)

7楼 [Alps1992](#) 2015-11-03 21:07发表



楼主你图画的问题，在B数组里面，第2步的时候，位置是 $(m/2 - n/4)$  不是 $(m-n)/2$

6楼 [码到功成](#) 2015-04-25 11:34发表



又是bug很多的代码，根本没法用，楼主你有好好测试过吗？

5楼 [keyou226](#) 2015-04-08 02:28发表



```
int a[] = {1, 4, 5, 11, 12};
int b[] = {2};
直接返回了一个很大的负数
```

```
int a[] = {1, 4, 5, 11, 12};
int b[] = {2, 3, 6, 7, 18, 19};
返回7 能无语么？ 不应是6么
```

4楼 [静水流风](#) 2014-11-08 18:11发表



test case A:  
int A[]={2};  
int B[]={1,3,5};

result: 3 （返回上中位数）

test case B:  
int C[] = {4};  
int D[] = {1,3,5};  
result: 3 （返回下中位数）

上两个测试case说明，该程序有可能返回上中位数，又有可能返回下中位数。行为不是很确定。

3楼 [zwjcxj](#) 2014-09-19 19:11发表



好像不对吧。楼主。

2楼 [hotmocha](#) 2013-03-07 20:10发表



假设开始的时候A长度为n，B长度为m，可不可能出现在 $m/2$ ， $(m-n)/2$ 呢，现在假设出现在这段区域，因为 $A[n/2] < B[n/2]$ ，所以当中位数在 $[m/2, (m-n)/2]$ ，那前面数的个数也就是 $n/2+m/2+...$ 这样显然是不对的？不知道我有没有想错

1楼 [hotmocha](#) 2013-03-07 20:02发表



这个时候，我们不能冲动地把 $A[0], A[n/2]$ 和 $B[m/2], B[m]$ 全部扔掉。？？这个为什么这么说？？能不能解释一下，看不懂

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker  
OpenStack VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC  
WAP jQuery BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML  
LBS Unity Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra  
CloudStack FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App  
SpringSide Maemo Compuware 大数据 aptech Perl Tomado Ruby Hibernate ThinkPHP  
HBase Pure Solr Angular Cloud Foundry Redis Scala Django Bootstrap

