

思维决定一切

嵌入式linux

目录视图 摘要视图 **RSS** 订阅

个人资料



zxzy1988

访问： 119496次
积分： 1668
等级：
排名： 第19366名

原创： 46篇 转载： 15篇
译文： 0篇 评论： 39条

文章搜索

文章存档

- 2013年07月 (1)
- 2013年04月 (14)
- 2013年02月 (24)
- 2012年11月 (4)
- 2012年08月 (1)

展开

阅读排行

- Leetcode 4 Median of Tv (18802)
- Leetcode 127 Word Ladder (13041)
- Leetcode 37 Sudoku Solver (6178)
- Leetcode 75 Sort Colors (4703)
- 关于冯诺依曼结构、哈佛 (3983)
- 拆解诺基亚E71，解决莫 (3676)
- Leetcode 128 Longest Common Subsequence (3603)
- 关于int *p[n]和int (*p)[n] (3210)
- 虚拟内存介绍及MMU工作原理 (2777)
- 在ARM linux mini2440上 (2571)

评论排行

- Leetcode 4 Median of Tv (6)
- 关于冯诺依曼结构、哈佛 (5)
- 在mini2440上面搞定CC2 (5)

CSDN 2016博客之星评选结果公布

【系列直播】零基础学习微信小程序！

“我的2016”主题征文活动

博客的神秘功能

Leetcode 4 Median of Two Sorted Arrays

2013-02-18 11:20

18804人阅读

评论(0) 收藏 分享

版权声明：本文为博主原创文章，未经博主允许不得转载。

今天发现了leetcode上面一道题，觉得非常经典，记录之。

题目是这样的：给定两个已经排序好的数组（可能为空），找到两者所有元素中第k大的元素。另外一种更加具体的形式是，找到所有元素的中位数。本篇文章我们只讨论更加一般性的问题：如何找到两个数组中第k大的元素？不过，测试是用的两个数组的中位数的题目，Leetcode第4题 Median of Two Sorted Arrays
方案1：假设两个数组总共有n个元素，那么显然我们有O(n)时间和O(n)空间的方法：用merge sort的思路排序，排序好的数组取出下标为k-1的元素就是我们需要的答案。

这个方法比较容易想到，但是有没有更好的方法呢？

方案2：我们可以发现，现在我们是需要“排序”这么复杂的操作的，因为我们仅仅需要第k大的元素。我们可以用一个计数器，记录当前已经找到第m大的元素了。同时我们使用两个指针pA和pB，分别指向A和B数组的第一个元素。使用类似于merge sort的原理，如果数组A当前元素小，那么pA++，同时m++。如果数组B当前元素小，那么pB++，同时m++。最终当m等于k的时候，就得到了我们的答案——O(k)时间，O(1)空间。
但是，当k很接近于n的时候，这个方法还是很费时间的。当然，我们可以判断一下，如果k比n/2大的话，我们可以从最大的元素开始找。但是如果我们要找所有元素的中位数呢？时间还是O(n/2)=O(n)的。有没有更好的方案呢？

我们可以考虑从k入手。如果我们每次都能够剔除一个一定在第k大元素之前的元素，那么我们需要进行k次。但是如果每次我们都剔除一半呢？所以用这种类似于二分的思想，我们可以这样考虑：

Assume that the number of elements in A and B are both larger than k/2, and if we compare the k/2-th smallest element in A(i.e. A[k/2-1]) and the k-th smallest element in B(i.e. B[k/2 - 1]), there are three results:

(Because k can be odd or even number, so we assume k is even number here for simplicity. The following is also true when k is an odd number.)

A[k/2-1] = B[k/2-1]

A[k/2-1] > B[k/2-1]

A[k/2-1] < B[k/2-1]

if A[k/2-1] < B[k/2-1], that means all the elements from A[0] to A[k/2-1](i.e. the k/2 smallest elements in A) are in the range of k smallest elements in the union of A and B. Or, in the other word, A[k/2 - 1] can never be larger than the k-th smallest element in the union of A and B.

Why?

We can use a proof by contradiction. Since A[k/2 - 1] is larger than the k-th smallest element in the union of A and B, then we assume it is the (k+1)-th smallest one. Since it is smaller than B[k/2 - 1], then B[k/2 - 1] should be at least the (k+2)-th smallest one. So there are at most (k/2-1) elements smaller than A[k/2-1] in A, and at most (k/2 - 1) elements smaller than A[k/2-1] in B. So the total number is k/2+k/2-2, which, no matter when k is odd or even, is surely smaller than k(since A[k/2-1] is the (k+1)-th smallest element). So A[k/2-1] can never be larger than the k-th smallest element in the union of A and B if A[k/2-1]<B[k/2-1];

Since there is such an important conclusion, we can safely drop the first k/2 element in A, which are definitely smaller than k-th element in the union of A and B. This is also true for the A[k/2-1] > B[k/2-1] condition, which we

关闭

Leetcode 127 Word Ladder I	(4)
关于int *p[n]和int (*p)[n]	(3)
拆解诺基亚E71，解决莫	(2)
Leetcode 37 Sudoku Solver	(2)
Leetcode 128 Longest Consecutive Sequence	(2)
关于linux下访问寄存器	(2)
虚拟内存介绍及MMU工作原理	(2)

推荐文章

- * Android 反编译初探 应用是如何被注入广告的
- * 凭兴趣求职80%会失败，为什么
- * 安卓微信自动抢红包插件优化和实现
- * 【游戏设计模式】之四 《游戏编程模式》全书内容提炼总结
- * 带你开发一款给Apk中自动注入代码工具icodetools(完善篇)

最新评论

- Leetcode 4 Median of Two Sorted Arrays
kidd4212: @biill: $O(\lg(m) + \lg(n)) = O(\lg(mn)) \leq O(2\lg(m+n))$...
- Leetcode 4 Median of Two Sorted Arrays
biill: 楼主啊，这个答案很简洁没错，可你确定它是 $O(\lg(m+n))$ 的吗？你想想每次就只去掉一个数组（不...
- Leetcode 4 Median of Two Sorted Arrays
answer3y: 博主，确定这个算法是 $O(\lg(m+n))$ 的吗？应该是 $\lg(m) + \lg(n) = \lg...$
- Leetcode 127 Word Ladder I
HEVC_CJL: 分析得很好，受益匪浅。有点瑕疵就是DFS算法在返回的时候有个遗漏，在对current值进行修改后，...
- Leetcode 4 Median of Two Sorted Arrays
WillenWu: 感谢，不过好像可以优化一下：if (a < b) return findKth(a, pa, b + ...
- Leetcode 127 Word Ladder I
chenliehuo7: 非常好，感谢分享这么详细
- Leetcode 127 Word Ladder I
newton0214: 这道题网上能找到的分析里面，这篇应该是分析的最好的
- Leetcode 75 Sort Colors
cane1991: 赞
- Leetcode 37 Sudoku Solver
jegeyom: 有点儿奇怪，solveSudoku()是recursion但又没把当前的位置传进去，下一个recur...
- Leetcode 128 Longest Consecutive Sequence
Eleven_Adam: @karaya:人家用的是unordered_map,内部存储结构是哈希表，和map的红黑树不一样，...

About Me

嵌入式爱好者
zhuxiaoyongsdu@163.com

友情链接

http://space.chinaaet.com
/jihceng0622 http://blog.csdn.net/youhaipeng

should drop the elements in B.

When $A[k/2-1] = B[k/2-1]$, then we have found the k-th smallest element, that is the equal element, we can call it m. There are each $(k/2-1)$ numbers smaller than m in A and B, so m must be the k-th smallest number. So we can call a function recursively, when $A[k/2-1] < B[k/2-1]$, we drop the elements in A, else we drop the elements in B.

We should also consider the edge case, that is, when should we stop?

1. When A or B is empty, we return $B[k-1]$ (or $A[k-1]$), respectively;
2. When k is 1 (when A and B are both not empty), we return the smaller one of $A[0]$ and $B[0]$
3. When $A[k/2-1] = B[k/2-1]$, we should return one of them

In the code, we check if m is larger than n to guarantee that we always know the smaller array, for coding simplicity.

```
[cpp]

01. double findKth(int a[], int m, int b[], int n, int k)
02. {
03.     //always assume that m is equal or smaller than n
04.     if (m > n)
05.         return findKth(b, n, a, m, k);
06.     if (m == 0)
07.         return b[k - 1];
08.     if (k == 1)
09.         return min(a[0], b[0]);
10.     //divide k into two parts
11.     int pa = min(k / 2, m), pb = k - pa;
12.     if (a[pa - 1] < b[pb - 1])
13.         return findKth(a + pa, m - pa, b, n, k - pa);
14.     else if (a[pa - 1] > b[pb - 1])
15.         return findKth(a, m, b + pb, n - pb, k - pb);
16.     else
17.         return a[pa - 1];
18. }

19.
20. class Solution
21. {
22. public:
23.     double findMedianSortedArrays(int A[], int m, int B[], int n)
24.     {
25.         int total = m + n;
26.         if (total & 0x1)
27.             return findKth(A, m, B, n, total / 2 + 1);
28.         else
29.             return (findKth(A, m, B, n, total / 2)
30.                 + findKth(A, m, B, n, total / 2 + 1)) / 2;
31.     }
32. };
```

顶

0

踩

0

关闭

上一篇 Leetcode 37 Sudoku Solver

下一篇 Leetcode 19 Remove Nth Node From End of List



猜你在找

- C++ 单元测试 (GoogleTest)
- Median of Two Sorted Arrays_leetcode_#4
- 渗透测试工程师学习-征途
- LeetCode 4Median of Two Sorted Arrays
- 微服务场景下的自动化测试
- LeetCode 4 Median of Two Sorted Arrays
- Web无障碍——测试与工具
- LeetCode 4 Median of Two Sorted Arrays
- iOS测试
- leetcode 数组4 Median of Two Sorted Arrays

查看评论

4楼 [biill](#) 2015-09-01 11:05发表



楼主啊， 这个答案很简洁没错，可你确定它是O(log(m+n))的么？你想想每次就只去掉一个数组（不是两个）的某一部分。用O(log(m))在A里查找，O(log(n))在B里查找。是O(log(m) + log(n)) = O(log(mn))复杂度。测试出来是O(log(mn))而不是O(log(m+n))。

Re: [kidd4212](#) 2016-06-10 20:39发表



回复biill：O(lg(m) + lg(n)) = O(lg(mn)) <= O(2lg(m+n)) = O(lg(m+n))

3楼 [answer3y](#) 2014-09-14 17:48发表



博主，确定这个算法是O(log(m+n))的吗？

应该是log(m) +log(n) = log(mn) 吧？

我也测试过，我m和n分别为10000，数组里面填的是随机数，findKth函数有30次调用，正好吻合log(100000000)

2楼 [WillenWu](#) 2014-07-01 17:11发表



感谢，不过好像可以优化一下：

```
if (a[pa - 1] < b[pb - 1])
return findKth(a + pa, m - pa, b, pb, k - pa);
else if (a[pa - 1] > b[pb - 1])
return findKth(a, pa, b + pb, n - pb, k - pb);
```

因为a[pa - 1] < b[pb - 1]时，一定不会出现在b数组的pb位置之后，a[pa - 1] > b[pb - 1]时，一定不会出现在数组a的pa位置之后。

优化后可以快大概100ms。

1楼 [apprentice89](#) 2013-09-09 21:19发表



求问博主：这个算法最坏情况下的时间复杂度是多少？？感觉是个常量时间（与|A|,|B|,k都没关系）？

Re: [Henry要当个小码农](#) 2013-10-22 13:28发表



回复apprentice89：如果每次都是缩减的a的长度， if (a[pa - 1] < b[pb - 1]) 那么很快a的m就变为0，直接b[k-1]得到结果。这不是最坏情况。最坏情况是k一直以k=k/2的速度递减，而且始终在a和b的长度范围内，这样直到k减为1，递归才停止。所以最坏情况时间复杂度O(log(k))=O(log((m+n)/2))=O(log(m+n))

Re: [apprentice89](#) 2014-01-06 18:16发表



回复Henry要当个小码农：多谢！

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


关闭

Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved 

关闭