

跟小D每日学口语

昵称: David Luo

园龄: 6年1个月

粉丝: 11

关注: 2

+加关注

2016年3月						
<	一	二	三	四	五	>
28	29	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

更多链接

我的标签

C++(2)

世界杯(2)

虚继承(1)

字符编码(1)

操作符重载(1)

动态规划(1)

骑行(1)

人生感悟(1)

随笔档案

2013年1月 (1)

2011年1月 (1)

2010年7月 (5)

文章分类

[ACM-DS-Algorithm](25)

[C/C++](21)

[Database]

[Distribute - System](1)

[Life](4)

[Linux](2)

[Python]

[Research-Paper]

interview(23)

羽毛球

相册

2010.7.2~4秦皇岛之旅(21)

Life(1)

羽动天空(2)

最新评论

1. Re:多源最短路径 弗洛伊德算法理解和实现

这种时间复杂度的 算法也可以拿出来，醉

两个有序数组中找中位数或者第K大的元素

RT,在两个有序数组中找中位数或者第K大的元素。

假设两个数组为A, B长度分别为m,n.分别是递增顺序。

可以采用的算法有很多：

首先想到的是类似MergeSort的方式，合并的同时找第K大元素，这个基本没难度，复杂度O(m + n)。

不过此算法并不是最优，还有Log级别复杂度的算法，此算法其实很简单，远没有很多网站的代码那么玄乎，以下——道来：

首先明白几个前提：

1.如果是求中位数，(m + n)是奇数还是偶数对结果是很有影响的，具体的如果(m + n)是奇数，中位数唯一，如果是偶数就有两个中位数，怎么取舍就看要求了。

2.如果找到的第k大数(中位数类似)是 X，如果X排在A中的 第Ax位置，X排序在B中的Bx位置，那么(Ax + Bx - 1) == k 是恒成立的。

明白了2中的前提后，我们就可以得到一个算法，在A数组中枚举X，加入在A中是第Ax个，那么可以反推B中第(k + 1 - Ax)个以及相邻元素和X的大小关系就可以得到一个Log级别复杂度的算法：

简单点我们可以这么想：

1)先假设第k大数在A中，我们首先从A中第(m/(m + n)) * (k - 1)个元素开始检查其是否是第k个元素，假设其值为A1,然后看B中第(k + 1 - (m/(m + n)) * (k - 1))个元素(B1)和A1是否相等，或者 大于B中第(k + 1 - (m/(m + n)) * (k - 1))个元素，小于B中第(k + 1 - (m/(m + n)) * (k - 1))+ 1个元素。满足及可以知道A1即为所求。如果两个条件都不满足，请看2。

2)如果两个条件都不满足，那么需要判断第k个元素是处于 A1的左边还是右边，这个就是典型的分治思想。具体的来说：

if A1 > B1 那么k可以排除肯定不在A[0, (m/(m + n)) * (k - 1)]以及B[(k + 1 - (m/(m + n)) * (k - 1))+ 1, n


中


if A1< B1 那么k可以排除肯定不在A[(m/(m + n)) * (k - 1), m]以及B[0, (k + 1 - (m/(m + n)) * (k - 1))]中

注意下临界条件(corner condition may not stastify, but the method is right)

第K个元素有可能在B中，同理可以假设在B中，然后再搜索一遍就可以查到。复杂度 log(m)+ log(n)

当然也可以两个数组一起找，总体代码如下：

```

1  int kthsmallest(int *a,int m,int *b,int n,int k) {
2      if (m == 0) {
3          return b[k - 1];
4      }
5      if (n == 0) {
6          return a[k - 1];
7      }
8      if (k == 1) {
9          return (a[0] < b[0])?a[0]:b[0];
10     }
11     if (k == m + n) {
12         return (a[m - 1] > b[n - 1])?a[m - 1]:b[n - 1];
13     }
14     int i = ((double) m) / (m + n) * (k - 1);
15     int j = k - 1 - i;
16     if (j >= n) {
17         j = n - 1;
18         i = k - n;
19     }
20     if (((i == 0) || (a[i - 1] <= b[j])) && (b[j] <= a[i])) {
21         return b[j];
22     }
23     if (((j == 0) || (b[j - 1] <= a[i])) && (a[i] <= b[j])) {
24         return a[i];
25     }
26     if (a[i] <= b[j]) {
27         return kthsmallest(a + i + 1, m - i - 1, b, j, k - i - 1);
28     } else {
29         return kthsmallest(a, i, b + j + 1, n - j - 1, k - j - 1);
30     }
31 }
32
```



阅读排行榜

1. 读《编程珠玑》有感(847)
2. 希望是真的-徐金梧校长在元旦晚会上的讲话(652)
3. [转]人不成熟的六个特征(334)
4. 堆排序复习(267)
5. [转]成大事必备9种能力、9种手段、9种心态(178)

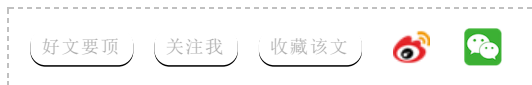
评论排行榜

1. [转]人不成熟的六个特征(1)
2. [转]成大事必备9种能力、9种手段、9种心态(1)

推荐排行榜

1. 读《编程珠玑》有感(1)

分类: [\[ACM-DS-Algorithm\]](#), [interview](#)





David Luo
关注 - 2
粉丝 - 11
[+加关注](#)

0 0

(请您对文章做出评价)

« 上一篇: [希望是真的-徐金梧校长在元旦晚会上的讲话](#)
» 下一篇: [堆排序复习](#)

posted @ 2012-12-26 23:42 David Luo 阅读(4566) 评论(1) 编辑 收藏

评论列表

#1楼 2014-12-15 01:54 NightElf


有两处问题:
1. $Ax + Bx - 1 == k$ 是恒成立的?
好像应该是 $Ax + Bx + 1 == k$
2. if $A1 > B1...$
if $A1 < B1...$
这两处好像说反了。
另外, 代码和描述有不符, 主要体现在这上述两处。
代码应该是求第k个数, 不是第k大数。

支持(1) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问网站首页](#)。

- 【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
- 【推荐】融云即时通讯云—专注为 App 开发者提供IM云服务
- 【推荐】UCloud开年大礼, 充5000返1000; 买云主机送CDN, 详情点击



jQuery MiniUI

快速开发WEB软件, 支持
Java、.Net

最新IT新闻:

- 传360总裁齐向东将率“360新闻团队”出走, 与周鸿祎分道扬镳
 - Ubuntu 16.04 LTS将移除私有的AMD催化剂驱动
 - 微软发布开源的R Tools for Visual Studio
 - Let's Encrypt客户端将由EFF管理
 - 那些只收钱不发货、或者光发地摊货的众筹团队, 后来都怎么样了?
- » 更多新闻...



最新知识库文章:

- 架构漫谈 (一): 什么是架构?
 - 架构的本质
 - 谷歌背后的数学
 - Medium开发团队谈架构设计
 - 理解“渐进增强(Progressive Enhancement)”
- » 更多知识库文章...