

Felix's Blog

Time goes on, life goes on.

首页

登入

RSS

注册

留言

链接

归档

关于

Search...

日志标题 ▼

站内

谷歌

联系我

felix021[#]gmail.com

最新评论

1.KeepAlive机制很多情况无法...  
do\_accept是否存在问题? sock...  
芝麻糊LOL  
主板?  
我记得我以前那个台式机在某...  
查看上下月这种操作... 我都...  
我在想额外的两个问题: 1. 能...  
我一般是 持续一段时间吃一套...  
喝多了不见得对男人好. 女的...  
少喝豆浆, 建议完毕.

分类

杂碎 [12]   
IT [706]   
操作系统 [125]   
Python [9]   
探索设计模式 [5]   
软件 [84]   
硬件 [40]   
手机 [12]   
程序设计 [162]   
网络 [196]   
数据库 [18]   
病毒 [7]   
其他 [48]

其他

登入  
注册  
RSS: 日志 | 评论  
编码: UTF-8  
XHTML 1.0

统计

访问次数 2098821  
今日访问 478  
日志数量 2099  
评论数量 2291  
引用数量 1  
留言数量 143  
注册用户 295  
在线人数 27

链接

默认链接组  
WHU微软俱乐部  
朋友们的据点  
czyhd's Blog  
姜南的BLOG  
Kid的原创Blog  
[GCC]Feli  
彼岸·花开·荼靡  
谁见过风?  
不敢流泪  
Liuw's Thinkpad

廉颇老矣 (装B一下)

字符串的Hash

★ 最长递增子序列 O(NlogN)算法

类别: IT » 程序设计 | felix021 @ 2009-5-13 04:15 | 评论(19) | 阅读(23607) | 大 | 中 | 小

今天回顾W0J1398, 发现了这个当时没有理解透彻的算法。  
看了好久好久, 现在终于想明白了。  
试着把它写下来, 让自己更明白。

最长递增子序列, Longest Increasing Subsequence 下面我们简记为 LIS。  
排序+LCS算法 以及 DP算法就忽略了, 这两个太容易理解了。

假设存在一个序列d[1..9] = 2 1 5 3 6 4 8 9 7, 可以看出来它的LIS长度为5。  
下面一步一步试着找出它。  
我们定义一个序列B, 然后令 i = 1 to 9 逐个考察这个序列。  
此外, 我们用一变量Len来记录现在最长算到多少了

首先, 把d[1]有序地放到B里, 令B[1] = 2, 就是说当只有1一个数字2的时候, 长度为1的LIS的最小末尾是2。这时Len=1

然后, 把d[2]有序地放到B里, 令B[1] = 1, 就是说长度为1的LIS的最小末尾是1, d[1]=2已经没用了, 很容易理解吧。这时Len=1

接着, d[3] = 5, d[3]>B[1], 所以令B[1+1]=B[2]=d[3]=5, 就是说长度为2的LIS的最小末尾是5, 很容易理解吧。这时候B[1..2] = 1, 5, Len=2

再来, d[4] = 3, 它正好加在1,5之间, 放在1的位置显然不合适, 因为1小于3, 长度为1的LIS最小末尾应该是1, 这样很容易推知, 长度为2的LIS最小末尾是3, 于是可以把5淘汰掉, 这时候B[1..2] = 1, 3, Len = 2

继续, d[5] = 6, 它在3后面, 因为B[2] = 3, 而6在3后面, 于是很容易可以推知B[3] = 6, 这时B[1..3] = 1, 3, 6, 还是很容易理解吧? Len = 3 了噢。

第6个, d[6] = 4, 你看它在3和6之间, 于是我们就可以把6替换掉, 得到B[3] = 4。B[1..3] = 1, 3, 4, Len继续等于3

第7个, d[7] = 8, 它很大, 比4大, 嗯。于是B[4] = 8。Len变成4了

第8个, d[8] = 9, 得到B[5] = 9, 嗯。Len继续增大, 到5了。

最后一个, d[9] = 7, 它在B[3] = 4和B[4] = 8之间, 所以我们知道, 最新的B[4] =7, B[1..5] = 1, 3, 4, 7, 9, Len = 5。

于是我们知道了LIS的长度为5。

!!!! 注意。这个1,3,4,7,9不是LIS, 它只是存储的对应长度LIS的最小末尾。有了这个末尾, 我们就可以一个一个地插入数据。虽然最后一个d[9] = 7更新进去对于这组数据没有什么意义, 但是如果后面再出现两个数字 8 和 9, 那么就可以把8更新到d[5], 9更新到d[6], 得出LIS的长度为6。

然后应该发现一件事情了: 在B中插入数据是有序的, 而且是进行替换而不需要挪动——也就是说, 我们可以使用二分查找, 将每一个数字的插入时间优化到O(logN)~~~~~于是算法的时间复杂度就降低到了O(NlogN)~!

代码如下:

```
int LIS(int d[], int n){  
    int *B = new int[n];  
    int i, left, right, mid, len = 1;  
    B[0] = d[1]; //为了和上面的一致, 我们从1开始计数吧 :)  
    for(i = 2; i <= n; ++i){  
        left = 0, right = len;  
        while(left <= right){  
            mid = (left + right) / 2;  
            if(B[mid] < d[i]) left = mid + 1; //二分查找d[i]的插入位置  
            else right = mid - 1;  
        }  
        B[left] = d[i];  
        len++;  
    }  
    return len;  
}
```

http://www.felix021.com/blog/read.php?1587

1/3

- Pumpkin's
- FOUR
- 快乐云の小屋
- 张磊的blog
- Felix's Blog
- Zhang Jiuan's notes
- Frank's Blog
- intijk
- PortWatcher's Blog
- 链链接接^\_^
- mlzy
- LiuYu's LOGS
- dutor

```
        B[left] = d[i]; //插入
        if(left > len) len++; //d[i]比现有的所有数字都大，所以left 才会大于 len。
    }
    delete[] B;
    return len;
}
```

--

转载请注明出自 <http://www.felix021.com/blog/read.php?1587> ，如是转载文则注明原出处，谢谢:)  
RSS订阅地址: <http://www.felix021.com/blog/feed.php> 。

chasuner 2013-4-15 14:00

代码好像有点小问题吧，二分查找的right = len - 1才对吧，然后下面left比较的时候也应该和len - 1 比较！

北叶青藤 2011-11-10 11:40

这是一条隐藏评论或留言。您需要以合适的身份登录后才能看到。

Felix021 2011-11-5 19:50

这是一条隐藏评论或留言。您需要以合适的身份登录后才能看到。

Felix021 2011-11-5 19:50

这是一条隐藏评论或留言。您需要以合适的身份登录后才能看到。

树大 2011-11-5 18:48

这是一条隐藏评论或留言。您需要以合适的身份登录后才能看到。

Felix021 2011-11-4 18:15

这是一条隐藏评论或留言。您需要以合适的身份登录后才能看到。


bigrabbit 2011-3-8 11:26

为什么插入 B[left] = d[i]; 时候，他们不需要比较呢？觉得如果B[left]>d[i],就不需要插入了。。

wwy 2010-7-25 11:18

很好，很强大! very good!


11111 2010-4-30 19:45



xg1990 2010-3-1 16:11

学到了~多谢学长~

Iloveyou 2009-8-2 16:47



felix021 回复于 2009-8-2 16:49

。。这是哪位阿。。。

LN 2009-7-23 09:50

经典

felix021 回复于 2009-7-23 16:19