

首页 0 题广场 > 对n个记录的线性表进行快速排序为减少算法的递归深度,以下叙述...

▼ [单选题]

对n个记录的线性表进行快速排序为减少算法的递归深度,以下叙述正确的是()

- A. 每次分区后,先处理较短的部分
- B. 每次分区后,先处理较长的部分
- C. 与算法每次分区后的处理顺序无关
- D. 以上三者都不对

添加笔记 | 邀请回答 | 收藏(2321) | 分享

问题信息

数组 链表 排序 递归

难度：

24条回答 2321收藏 8232浏览

热门推荐



相关试题

一个有向无环图的拓扑排序序列(...

图 排序

评论 (5)

以下属于逻辑结构的是()

链表 队列 哈希

评论

下述哪一条是顺序存储结构的优

数组

评论

设有关键字 $n=2^k$...

树 查找

对有序数组{2、11、15

腾讯 数组 查找



扫一扫下载牛客APP

23个回答

添加回答

83 牛客444334号

▼ A
在快速排序中,需要使用递
查看全部

编辑于 2015-02-10 14:44:37

回复(14)

159 notlie

看到很多人在别的答案下说看不懂,那我就来举个栗子。
现在有这么个序列:123456789
假设每次划分出短序列的长度为1
即第一次划分
短序列:1
长序列:23456789
如果优先处理短序列1
则栈中仅用保存23456789,深度为1
然后23456789出栈,划分
短序列:2
长序列:3456789
同样的先处理短序列
栈中保存3456789,深度为1
类推下去,处理完整个序列,栈的最大深度都为1

假如每次划分出的短序列长度为2呢
短序列:12
长序列:3456789
优先处理短序列12
栈中保存3456789 深度为1

刷真题、补算法、看面经、得内推

使用第三方账号直接登录使用吧：

更多

然后2 出栈 处理
0 接着3456789出栈
划分为短序列34
长序列 56789
处理短序列34
栈中保存56789
类推下去，处理完整个序列，栈的最大深度都为2

也就是说栈的最大深度取决于划分出来的短序列的长度（前提是先处理短序列）

那么先处理长序列呢

短序列：1

长序列：23456789

如果优先处理长序列序列23456789 短序列入栈，长序列划分为2和3456789

2入栈，3456789划分

...

8入栈，9处理

此时栈中有12345678 深度为8

短序列：12

长序列：3456789

12入栈 3456789划分为34和56789

34入栈 56789划分为56和789

56入栈 789划分为78和9

9入栈 78划分为7和8

7入栈 8处理

此时栈中有12 34 56 9 7 深度为5

很明显先处理长序列 栈的深度要大于 先处理短序列栈的深度。

编辑于 2016-08-07 16:36:27


回复(24)

19  纽克牛客

觉得还是要弄清递归深度和栈深度的区别吧 看了那么多回复，感觉这个题问的是递归深度（也就是自己调用自己的次数）栈深度就是这些空间复杂度，对于栈深度，不论是先处理长序列还是先处理短序列，空间复杂度都是相同的，都是元素的总个数，但是递归深度就不同了，上面举得例子说，极端情况下，每次短序列长度均为1，那么递归深度始终是1，（优先处理短序列），如果先处理长序列，那么递归的深度就要大于1，因为长序列可能要多次递归，才能出栈，最终长序列递归完之后，还要再递归一次短序列（长序列优先），所以说，长序列优先的递归深度要大于短序列优先

发表于 2015-11-15 15:29:18

回复(2)

25  笑颜如玉

考虑一种极端情况，如果每次确定pivot后短的部分都为1，先处理短的部分，短的部分处理完后就会出栈，那么辅助栈的深度为 $O(1)$ 即可，同理，如果先处理长的部分，较长的部分不断递归压栈，辅助栈的深度就是 $O(n)$ 。所以，如果每次都先处理较短的部分，那么长而化小，栈区的深度较小，反之，较长的部分不断递归压栈，栈区的长度就会大。



扫一扫下载
牛客APP



编程小白

0

一看这题没有出处我就知道可能要错了。只要你是用函数递归的方法做快排，而且不是手工栈模拟递归，函数递归几层栈就是几层。其实你可以把函数调用过程想象成一棵树，不管你前面是怎么dfs的，到某个点为止，在这个点处的递归深度(函数递归了几层或者说栈叠了几层)只取决于该点的深度。也许这个题目是建立在算法某种特定的实现方式上时才是对的。对于评论区里那个详细的例子我只想说，123456789到12345678到1234567，递归层数也是在增加的除非你自己特定实现下，就好比一棵没有左节点的树。要是按链访问深度为1(事实上没有递归)，按dfs递归访问那深度就是n了。。

发表于 2017-02-25 09:06:56

回复(1)



Cookie1997

2

我觉得题目把递归深度改为栈深度比较好。

发表于 2018-03-20 21:02:12

回复(0)



data_poper

2

长序列先进行递归时，只需将短序列递归地址压入栈中，
短序列先进行递归时，只需将长序列递归地址压入栈中，
两个地址所用栈深度是一样的
因此这个问题可以等价
为长序列递归和短序列递归哪个序列的递归深度深一点，明显是长序列

发表于 2015-09-09 10:45:03

回复(0)



pioneer

1

递归深度应该不会受处理顺序的影响吧。按照其他人的解析，是说调用栈的深度受处理顺序影响。

编辑于 2017-07-26 21:30:17

回复(0)



Mr.DoubleU

1

我觉得是这样的,总递归次数,二者完全一样,但是最大递归深度,由于总是要递归长的那一边,所以最大递归深度要大一,而总是递归短的一边,没递归几次就返回了,虽然递归总次数一样,但是最大递归深度不一样

发表于 2018-04-05 20:36:02

回复(0)



跳腾的优秀咸鱼

0

递归深度不会改变，但堆栈的使用量增大

发表于 2018-04-04 22:08:14

回复(0)



牛客_1

0

如果递归先处理长的字符串，长的字符串肯定要再次划分，每一次划分都导致递归的深度加1
而如果处理短的字符串，则需要划分的次数比长的少
因此短的字符串先处理，会导致递归的深度小

发表于 2017-09-25 21:22:54

回复(0)



扫一扫下载
牛客APP

Are you sure??

0
发表于 2015-08-07 16:20:55
回复(0)

0
牛客103459723号

快速排序想要减少递归深度,需要先处理短的序列:因为长的序列,短的序列,在栈中存的都是指针,大小是一样的,但是如果先处理长的,因为只能一个提个的处理,所以必须要不断的分解为很多小的,然后不断地存入栈中,这样就会加大栈的深度,也就是算法的递归深度(这里的递归深度,不是指递归树的深度,而是指栈的深度)

发表于 2018-09-14 14:30:14
回复(0)

0
album

快速排序时,左右子段均衡对于排序效率是最高的
短的递归优先,可以减少递归深度

发表于 2018-05-04 15:32:04
回复(0)

0
suoga

这题坑爹啊

发表于 2015-03-08 21:21:48
回复(0)

0
ufo008ahw

扯淡

发表于 2015-07-27 01:06:49
回复(0)

0
水平的天平

最快优先原则,好像是计算机系统的知识

发表于 2016-10-28 16:42:19
回复(0)

0
凌梦初辰

每次快速排序总会把排序的那段分成两段,一般来说这两段会有长短之分(但也不排除有长度相同的可能,这个无所谓)。如果每一段先处理较长的那段,那么如此递归下来,递归的深度会很长,毕竟越到最后每个小区间的长度越短,所以深度肯定会很大。

发表于 2017-07-27 18:03:21
回复(0)

0
牛客8034813号

递归深度:当前递归函数的frame数目,或者是函数实例的数目 运行树:任意时刻对应一条根到特定节点的路径

编辑于 2017-04-27 14:36:09
回复(0)

暖灯

小白拙见,还望大神指正:

首先我觉得这个题目的意思不是很明确。什么叫“减少算法的递归深度”?

下面的分析基于对“算法的递归深度”的如下定义——一个算法的递归深度指的是它的最大递归深度。



扫一扫下载
牛客APP

- 第一次快排后得到1和2,3,4两段。
- 0 不管先处理谁，下面的两个函数都要被调到
- quicksort(src, indexShorter, 1);
- quicksort(src, indexLonger, 3);
- 因此，最大的递归深度总是3。

发表于 2016-01-18 17:16:13

回复(0)



- 个人认为这和算法的具体实现有关。
- 如果使用递归的方式实现，那么就算先处理较长部分，也不意味着要把较短部分的相关信息压栈，因为处理较短部分的函数调用语句还未执行，不会占用系统栈空间导致其深度相对先处理较短部分的情况更大。这时选C是对的。
- 如果是使用迭代+辅助栈实现算法，这道题选A是合理的。

发表于 2016-08-31 21:42:14

回复(0)

首页 上一页 1 2 下一页 末页

代码语言

MD编辑器

请输入正确的答案及解题思路

提交观点



扫一扫，把题目装进口袋

牛客网，程序员必备求职神器

关于我们 加入我们 意见反馈 企业服务 联系我们
免责声明 友情链接

公司地址：北京市朝阳区大屯路东金泉时代3-2708北京牛客科技有.....]

联系方式：010-60728802(电话) admin@nowcoder.com

牛客科技©2018 All rights reserved

京ICP备14055008号-4

京公网安备 11010502036488号



扫一扫下载
牛客APP