

38

1

原创

算法导论-----递归算法的时间复杂度求解

2016-12-04 19:31:11 GNG 阅读数 45935 更多

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/so_geili/article/details/53444816

目录

- 1.算法设计与分析概述
- 2.非递归算法分析
- 3.递归算法分析
 - 3.1利用数列知识
 - 3.2代入法
 - 3.3递归树
 - 3.4主方法求解递推式
- 4.参考资料

1.算法设计与分析概述

在总结递归算法的时间复杂度分析之前，应该明确几组概念。

算法仅仅是求解问题的解决方案，这个解决方案本身并不是问题的答案，而是能获得答案的指令序列。只有通过执行算法才可以获得求解问题的答案。从算法是否递归调用的角度看，算法可以分为非递归算法和递归算法。

非递归算法时间复杂度分析较为简单，通常是计算算法中基本语句执行次数，一般都是一个关于问题规模n的表达式，然后用渐近符号 Θ 、 O 、 o 、 Ω 、 ω 表示出算法的时间复杂度。

递归算法是采用分治的方法，把一个“大问题”分解出若干个相似的“小问题”求解。在分析算法复杂度时，关键是根据递归过程建立递推关系式，得到算法执行的时间表达式（一般都与问题规模n相关），最后用渐近符号 Θ 、 O 、 o 、 Ω 、 ω 表示出算法的时间复杂度。

在《算法导论》、《算法设计与分析》这2门课中，我们已经学习一些通用的算法设计技术，如增量法、分治法、贪心法、动态规划、线性规划、| 限界法等；在算法设计完成后，对算法的复杂度进行分析是必然的，所以本篇的中心将围绕算法时间复杂度展开。

2.非递归算法分析

例1：如果算法的执行时间不随着问题规模n的增加而增长，它的基本语句执行的次数是固定的，总的时间由一个常数来限界。此类算法的时间复杂度是

例2：当有若干个循环语句时，时间复杂度是由嵌套层数最多的循环语句中的基本语句的执行次数决定。如下

```
1 void fun(int n){
2     int x=0;
3     for(int i=1;i<=n;i++){
4         for(int j=1;j<=i;j++){
5             for(int k=1;k<=j;k++){
6                 x++;           //基本语句
7             }
8         }
9     }
10 }
```

解：该算法的基本语句是x++;所以

$$f(n) = \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^j 1 = \sum_{i=1}^n \sum_{j=1}^i j = \sum_{i=1}^n \frac{i(i+1)}{2} = \dots = O(n^3), \text{ 时间复杂度为 } O(n^3)。$$

虽然非递归算法的时间复杂度比较好分析，但往往需要用到多项式的求和技巧和放缩技巧，如：



1. 等差数列 $\{a_k\}$ 求和: $\sum_{k=1}^n a_k = \frac{n(a_1+a_n)}{2}$
2. 等比数列 $\{aq^k\}$ 求和: $\sum_{k=0}^n aq^k = \frac{a(1-q^{n+1})}{1-q}$
3. 调和级数 $\{\frac{1}{k}\}$ 求和: $\sum_{k=1}^n \frac{1}{k} = \ln n + O(1)$ (需要用微积分知识证明)
4. 对数级数 $\lg 1 + \lg 2 + \dots + \lg n = \lg(n!) = \Theta(n \lg n)$ (利用Stirling公式证明)
5. 放缩1: 用序列中的最大项代替序列中的每个项, 这种方法可以表示为: $\sum_{k=1}^n a_k \leq na_{max}$
6. 放缩2: 在等比数列中, 假设存在常数 $r < 1$, 使得 $\frac{a_{k+1}}{a_k} \leq r$ 对一切 $k \geq 0$ 成立, 那么有:

$$\sum_{k=0}^n a_k \leq \sum_{k=0}^{\infty} a_0 r^k = a_0 \sum_{k=0}^{\infty} r^k = \frac{a_0}{1-r}$$

7. 放缩3: 对每个 $b > 1$ 和每个 $a > 0$, 都有 $\log_b n = o(n^a)$
8. $1 + 3 + 5 + \dots + (2n-1) = n^2$
9. $1^2 + 2^2 + 3^2 + \dots + n^2 = \frac{1}{6}n(n+1)(2n+1) = O(n^3)$ (幂方级数, 上限比幂次高出一阶)
10. $1^3 + 2^3 + 3^3 + \dots + n^3 = [\frac{1}{2}n(n+1)]^2 = O(n^4)$
11. 各种收敛级数, 不再一一列举。

👍
38

👤

💬
1

📖

🔖

📱

<

>

👑

小结: 以上这些求和及放缩技巧在下面的递归树求总代价、主定理证明过程中也要用到, 非常的重要!

3.递归算法分析

3.1利用数列知识

1. 累加法: 递推关系式为 $a_{n+1} - a_n = f(n)$ 采用累加法。
2. 累乘法: 递推关系式为 $\frac{a_{n+1}}{a_n} = f(n)$ 采用累乘法。
3. 构造法: 递推关系式为 (1) $a_{n+1} = pa_n + q$, (2) $a_{n+1} = pa_n + q^n$, 都可以通过恒等变形, 构造出等差或等比数列, 利用等差或等比数列的定其中的构造方法可通过待定系数法来进行。
4. 和化项法: 递推公式为 $S_n = f(n)$ 或 $S_n = f(a_n)$ 一般利用

$$a_n = \begin{cases} S_1, & \text{当 } n = 1 \\ S_n - S_{n-1}, & \text{当 } n \geq 2 \end{cases}$$

5. 用特征方程求解递推方程 (感觉比较生僻, 不做解释)
6. **迭代法:** 从原始递推方程开始, 反复将对于递推方程左边的函数用右边的等式代入, 直到得到初值, 然后将所得的结果进行化简。
例如在调用归并排序mergeSort(a,0,n-1)对数组a[0...n-1]排序时, 执行时间 $T(n)$ 的递推关系式为:

$$T(n) = \begin{cases} O(1), & \text{当 } n = 1 \\ 2T(\frac{n}{2}) + O(n), & \text{当 } n \geq 2 \end{cases}$$

其中, $O(n)$ 为merge()所需要的时间, 设为 cn (c 为正常量)。因此:

$$\begin{aligned} T(n) &= 2T(\frac{n}{2}) + cn = 2(2T(\frac{n}{4}) + \frac{cn}{2}) + cn = 2^2T(\frac{n}{4}) + 2cn \\ &= 2^3T(\frac{n}{8}) + 3cn = \dots = 2^kT(\frac{n}{2^k}) + kcn \\ &= nO(1) + cn \log_2 n \\ &= O(n \log_2 n), (\text{假设 } n = 2^k, \text{ 则 } k = \log_2 n) \end{aligned}$$

忽略求解细节。在我们求解递归式时, 因为最终是要求得一个时间上限, 所以在求解时常常省略一些细节。比如mergeSort(a,0,n-1)运行时间的实是:

$$T(n) = \begin{cases} O(1), & \text{当 } n = 1 \\ T(\lceil \frac{n}{2} \rceil) + T(\lfloor \frac{n}{2} \rfloor) + O(n), & \text{当 } n \geq 2 \end{cases}$$

👤

👑

但我们忽略这些上取整、下取整以及边界条件, 甚至假设问题规模 $n = 2^k$, 这都是为方便求解而忽略的细节。经验和一些定理告诉我们, 这些细节不会影响算法时间复杂度

类似的，我们也可以用迭代法求解汉诺塔递归求解时的时间复杂度。但遗憾的是，迭代法一般适用于一阶的递推方程。对于二阶及以上（即 $T(n)$ 依赖它前面更多个递归项）的递推方程，迭代法将导致迭代后的项太多，从而使得求和公式过于复杂，因此需要将递推方程化简，利用差消法等推方程化为一阶递推方程。如在求快速排序算法平均时间复杂度 $T(n)$ 的递推方程， $T(n)$ 依赖 $T(n-1)$ 、 $T(n-2)$ 、 \dots 、 $T(1)$ 等所有的项，这样称为全部历史递推方程。（这里省略快速排序算法平均复杂度 $T(n)$ 的求解过程）

小结：上面6种递推关系是高中、本科知识，在此重点介绍了迭代法，其它几种方法虽未在本篇中使用，但可以加深递推式求解的认识。

3.2代入法

代入法实质上就是数学归纳法，因此求递推式分为两步：

- 1. 猜测解的形式；
- 2. 用数学归纳法求出解中的常数，并证明解是正确的。

遗憾的是并不存在通用的方法来猜测递归式的正确解，需要凭借经验，偶尔还需要创造力。即使猜出了递归式的解，也需要用数学归纳法验证其正确性。正是由于该方法技术细节较为难掌握，因此这个方法不适合用来求解递归方程，反而比较适合作为其他方法的辅助验证手段。在此不做总结。可以翻阅《算法导论》进行学习。

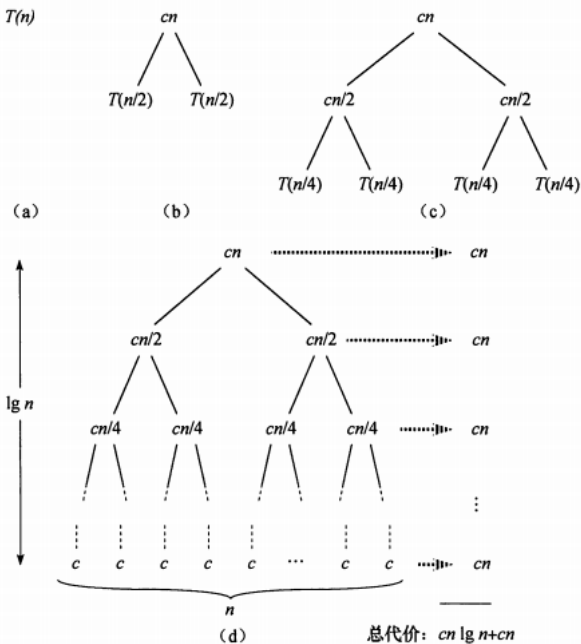
3.3递归树

递归树是一棵结点带权值的树。初始的递归树只有一个结点，它的权标记为 $T(n)$ ；然后按照递归树的迭代规则不断进行迭代，每迭代一次递归树直到树中不再含有权值为函数的结点（即叶结点都为 $T(1)$ ）。下面以递归方程

$$T(n) = \begin{cases} O(1), & \text{当 } n = 1 \\ 2T(\frac{n}{2}) + O(n), & \text{当 } n \geq 2; (\text{假设 } n = 2^k, \text{ 则 } k = \log_2 n) \end{cases}$$

来讲递归树的迭代规则。

- 第一步：把根结点 $T(n)$ 用根是 cn 、左结点为 $T(\frac{n}{2})$ 、右结点为 $T(\frac{n}{2})$ 的子树代替（即：以分解、合并子问题需要的代价为根，分解得到的子问题为叶的子树。其求解规模为1的问题所需的时间）；（如下如(a) \rightarrow (b)）
- 第二步：把叶结点按照“第一步”的方式展开； $T(\frac{n}{2})$ 用根是 $cn/2$ 、左结点为 $T(\frac{n}{4})$ 、右结点为 $T(\frac{n}{4})$ 的子树代替。（如下如(b) \rightarrow (c)）
- 第三步：反复按照“第一步”的方式迭代，每迭代一次递归树就增加一层，直到树中不再含有权值为函数的结点（即叶结点都为 $T(1)$ ）。（如下如(c) \rightarrow (d)）



在得到递归树后，将树中每层中的代价求和，得到每层代价，然后将所有层的代价求和，得到所有层次的递归树的总代价。在上图(d)部分中，递归树高度为 $\lg n$ （树高为根结点到叶结点最长简单路径上边的数目），所有递归树具有 $\lg n + 1$ 层，所以总代价为 $cn(\lg n + 1)$ ，所有时间复杂度为 $O(n \lg n)$ 。

总结：递归树模型求解递归方程，本质上就是迭代思想的应用，利用递归方程迭代展开过程构造对应的递归树，然后把每层的时间代价进行求和。模型更直观，同时递归树也克服了二阶及更高阶递推方程不方便迭代展开的痛点。

3.4主方法求解递推式

主方法为如下形式的递归式提供了一种“菜谱”式的求解方法，如下所示

T(n) = aT(n/b) + f(n)

其中a ≥ 1和b > 1是常数，f(n)是渐近正函数。这个递推式将规模为n的问题分解为a个子问题，每个子问题的规模为n/b，a个子问题递归地求解，然后合并子问题的解，合并的代价为f(n)。函数f(n)包含了问题分解和子问题解合并的代价。同样，这个递归式也没有考虑上取整、下取整、边界条件等，结果不会影响递归式的渐近复杂度。

定理4.1(主定理) 令a ≥ 1和b > 1是常数，f(n)是一个函数，T(n)是定义在非负整数上的递归式：

T(n) = aT(n/b) + f(n)

其中我们将n/b解释为⌊n/b⌋或⌈n/b⌉。那么T(n)有如下渐近界：

- 1. 若对某个常数ε > 0有f(n) = O(n^{(log_b a) - ε})，则T(n) = Θ(n^{log_b a})
- 2. 若f(n) = Θ(n^{log_b a})，则T(n) = Θ(n^{log_b a} lg n)。
- 3. 若对某个常数ε > 0有f(n) = Ω(n^{(log_b a) + ε})，且对某个常数c < 1和所有足够大的n有af(n/b) ≤ cf(n)，则T(n) = Θ(f(n))

在使用主定理之前，要比较f(n)和(n^{log_b a})的大小，这个大小不是算术意义上的大小比较，而是要在多项式意义上比较。以上三种情况在多项式意义上覆盖了f(n)的所有可能性。情况1和情况2之间有一定间隙；情况2和情况3之间也有一定间隙。如果f(n)落在这两个间隙中，或者情况3中正则条件不成立，主方法求递推式。

如在递归式：T(n) = 2T(n/2) + n lg n中，因为n^{log₂ 2} = n < f(n) = n lg n，但是f(n)并不大于n一个多项式因子n^ε，因为对于给定的ε > 0，均有n^ε > lg n。所以找不到这样ε > 0，该递归式落入了情况2和情况3之间的间隙，不能使用主定理。

最后给出主定理应用的几个练习题：

Ex.

T(n) = 4T(n/2) + n
a = 4, b = 2 ⇒ n^{log_b a} = n²; f(n) = n.
CASE 1: f(n) = O(n^{2 - ε}) for ε = 1.
∴ T(n) = Θ(n²).

Ex.

T(n) = 4T(n/2) + n²
a = 4, b = 2 ⇒ n^{log_b a} = n²; f(n) = n².
CASE 2: f(n) = Θ(n² lg⁰ n), that is, k = 0.
∴ T(n) = Θ(n² lg n).

Ex.

T(n) = 4T(n/2) + n³
a = 4, b = 2 ⇒ n^{log_b a} = n²; f(n) = n³.
CASE 3: f(n) = Ω(n^{2 + ε}) for ε = 1
and 4(cn/2)³ ≤ cn³ (reg. cond.) for c = 1/2.
∴ T(n) = Θ(n³).

Ex.

T(n) = 4T(n/2) + n²/lg n
a = 4, b = 2 ⇒ n^{log_b a} = n²; f(n) = n²/lg n.
Master method does not apply. In particular,
for every constant ε > 0, we have n^ε = ω(lg n).

4.参考资料

- 1. 《算法导论》第三版
- 2. 《算法设计与分析》屈婉玲著，清华大学出版社
- 3. 《算法设计与分析》李春葆著，清华大学出版社



人脸识别主要算法原理

人脸识别
1011阅读

38



1



想对作者说点什么

递归算法的时间复杂度分析

在算法分析中，当一个算法中包含递归调用时，其时间复杂度的分析会转化为一个递归方程求解。实际上，这个问题... 博文

阅读数 2万+

青青的专栏--C/C++



阅读数 1万+

w36680130的博客



阅读数 5万+

budapest的专栏



递归函数时间复杂度分析

递归函数时间复杂度分析------(1)递归执行过程

博文

递归程序的时间复杂度分析

阅读数 973

一，递归的四大基本法则：①基准情形基准情形是指那些不需要递归（不需要经过函数调用）之后就能退出的情况。... 博文

来自： qq_36327491的博客

男子发现妻子上班穿短裙腿上毫无遮拦，妻子：工作需要，老总要求

搜狐·猎媒

递归树求解递归算法的时间复杂度(看这篇博客看懂的)

阅读数 5227

递归算法时间复杂度的计算方程式一个递归方程：

在引入递归树之前可以考虑一个例子：

$T(n)=2T(n/2)...$

博文

来自： onlyoncelove的博客

算法导论-----渐近记号 Θ 、 O 、 o 、 Ω 、 ω 详解

阅读数 2万+

算法设计与分析这门课学了很久了，竟然对 Θ 、 O 、 o 、 Ω 、 ω 还没有一个清晰的认识，是总结一下... 博文

来自： 麦田里的码农

递归树分析递归算法的时间复杂度

阅读数 646

$T(n)=T(n/3)+T(2n/3)+n$ 其递归树如下图所示：

可见每层的值都为n，从根到叶节点的最长路径是： ... 博文

来自： qq_27840621的博客

递归式时间复杂度分析 《算法导论》

阅读数 2222

算法导论中递归式求解时间复杂度的三种方法：（一）代换法： 1.猜测解的形式； 2.用数学归纳法求出解中... 博文

来自： ninesun的博客

我花了一夜用数据结构给女朋友写个H5走迷宫游戏

阅读数 11万+

起因又到深夜了，我按照以往在csdn和公众号写着数据结构！这占用了我大量的时间！我的超越妹妹严重缺乏陪伴而... 博文

来自： bigsai

算法导论---递归算法的时间复杂度求解 - CSDN博客

递归树求解递归算法的时间复杂度(看这篇博客看懂的) - ..._CSDN博客

反转！“只问了1个框架，就给了35K的Python岗”

学Python的程序员建议收藏！

学会了这些技术，你离BAT大厂不远了

阅读数 9万+

每一个程序员都有一个梦想，梦想着能够进入阿里、腾讯、字节跳动、百度等一线互联网公司，由于身边的环境等原... 博文

来自： 平头哥的技术博文

算法导论---递归算法的时间复杂度求解 - passer's blog - CSDN博客

迭代复杂度分析 - czl_Lynn的博客 - CSDN博客



阅读数 1792

递推公式求时间复杂度

例子1: 斐波那契数列的定义如下: $F_1 = 1, F_2 = 1, F_n = F_{n-1} + F_{n-2} (n \geq 3)$ 。如果用递归算法计算斐波那契数列的第n项... 博文

来自： huangjin_1995的...



556篇文章
排名:3000+

关注



391篇文章
排名:千里之外

关注



183篇文章
排名:8000+

关注



44篇文章
排名:千里之外

关注

算法导论---递归算法的时间复杂度求解 - 朗朗晨光的专..._CSDN博客

算法导论---递归算法的时间复杂度求解 - onlyoncelove..._CSDN博客

递归算法时间复杂度计算

https://blog.csdn.net/so_geili/article/details/53444816

别再翻了，面试二叉树看这 11 个就够了~

写在前边数据结构与算法：不知道你有没有这种困惑，虽然刷了很多算法题，当我去面试的时候，面试官让你手写一...

算法导论中求解时间复杂度的三种方法 - 坚持 - CSDN博客

算法:递归 - caoPengFlying的博客 - CSDN博客

算法的时间复杂度 递推

算法的时间复杂度 递推1. NOIP2013提高组初赛7.斐波那契数列的定义如下： $F_1=1, F_2=1, F_n=F_{n-1}+F_{n-2}(n \geq 3)$ 。如...

年度榜单：Python三连冠，碾压Java！你怎么看？

IEEE Spectrum近日发布了2019年度编程语言排行榜，令人些许意外的是，Python连续三年问鼎巅峰，你怎么看？

递归算法的时间复杂度 - rock_joker的博客 - CSDN博客

递归算法时间复杂度分析

递归算法时间复杂度分析时间复杂度：一般情况下，算法中基本操作重复的次数就是问题规模n的某个函数...

分享靠写代码赚钱的一些门路

作者mezod，译者josephchang10如今，通过自己的代码去赚钱变得越来越简单，不过对很多人来说依然还是很难...

让程序员崩溃的瞬间（非程序员勿入）

今天给大家带来点快乐，程序员才能看懂。来源：<https://zhuanlan.zhihu.com/p/470665211>.公司实习生找Bug2....

关于递归程序的时间复杂度

关于递归程序的时间复杂度主定理递归中，一个规模为n的问题分成a个规模为n/b的问题，额外计算复杂度为 $c \cdot n^d$...

算法导论-----最长公共子序列LCS（动态规划）

目录一.概念梳理二.最长公共子序列解决方案方案1：蛮力搜索策略方案2：动态规划策略三、C代码实现实现1实现2(...

一本可陪伴一辈子的笔记本

可擦可写，可循环利用，支持OCR识别，让你的笔记本变得智能

技术一旦被用来作恶，究竟会有多可怕？

技术一直都在被用来作恶。作为与经常与黑客、攻击者打交道的我们，熟知各种用技术作恶的手段。这篇就作为简单...

GitHub开源的10个超棒后台管理面板

目录1、AdminLTE2、vue-Element-Admin3、tabler4、Gentelella5、ng2-admin6、ant-design-pro7、blur-ad...

算法导论-----快速排序QuickSort

目录：一、快速排序思想介绍二、实现的三步骤（分解、子问题求解、合并）三、C代码实现3.1快速排序双向扫描法...

时间复杂度分析：递归算法

斐波那契数列，又称黄金分割数列，指的是这样一个数列：0、1、1、2、3、5、8、13、21、34、.....在数学上，斐...

一般递归时间复杂度分析

使用：master公式计算。当满足 $T(n)=a \cdot T(n/b)+O(n^d)$ 时。其中 $T(n)$ 为样本量的时间复杂度，a为子过程发生多少...



38



1



阅读数 266

来自: fnjeikd的博客

阅读数 5万+

来自: 一个不甘平凡的码农

阅读数 4742

来自: mrcrack的博客

阅读数 8647

来自: AcceptedLin的博客

阅读数 3万+

来自: Python之禅的专栏

阅读数 15万+

来自: strongerHuang

阅读数 1005

来自: chy2003的博客

阅读数 2万+

来自: 麦田里的码农

阅读数 8538

来自: 知道创宇KCSC

阅读数 3万+

来自: 不脱发的程序猿

阅读数 1780

来自: 麦田里的码农

阅读数 2985

来自: huayyu的博客

阅读数 354

来自: weixin_43971252...

广告

阿里资深工程师教你如何优化 Java 代码！

作者 | 王超 责编 | 伍杏玲 明代王阳明先生在《传习录》谈为学之道时说： 私欲日生，如地上尘，一日不扫，便又有...

对计算机专业来说学历真的重要吗？

我本科学校是渣渣二本，研究生学校是985，现在毕业五年，校招笔试、面试，社招面试参加了两年了，就我个人的...

C语言实现推箱子游戏

很早就想过做点小游戏了，但是一直没有机会动手。今天闲来无事，动起手来。过程还是蛮顺利的，代码也不是非常...

面试官，不要再问我三次握手和四次挥手

三次握手和四次挥手是各个公司常见的考点，也具有一定的水平区分度，也被一些面试官作为热身题。很多小伙伴说...

德国 IT 薪酬大揭秘！

作者 |德国IT那些事责编 | 伍杏玲 “所有脱离工龄、级别、职位、经验、城市以及裙带关系来谈论工资，都是耍流氓！ ...



人脸识别主要算法原理

人脸识别

1523阅读

据说中台凉了？唔，真香

全文长度: 2200字 阅读时间: 8分钟 TL;DR(too long don'tread) 1、业务中台就是流程模板+扩展点 2、没法很好抽...

500行代码，教你用python写个微信飞机大战

这几天在重温微信小游戏的飞机大战，玩着玩着就在思考人生了，这飞机大战怎么就可以做的那么好，操作简单，简...

史上最全的中高级JAVA工程师-面试题汇总

史上最全的java工程师面试题汇总，纯个人总结，精准无误。适合中高级JAVA工程师。...

别在学习框架了，那些让你起飞的计算机基础知识。

我之前里的文章，写的大部分都是与计算机基础知识相关的，这些基础知识，就像我们的内功，如果在未来想要走的...

Docker技术(容器虚拟化技术)

Docker虚拟化容器技术 第一章 Docker简介诞生背景Docker 介绍虚拟机技术容器虚拟化技术官方网址第二章 Docke...



人脸识别主要算法原理

人脸识别

1523阅读

MySQL数据库—SQL汇总

一、准备 下文整理常见SQL语句的用法，使用MySQL5.7测试，参考了尚硅谷MySQL教程及用例。用例sql： 链接: h...

java秀发入门到优雅秃头路线导航【教学视频+博客+书籍整理】

在博主认为，学习java的最佳学习方法莫过于视频+博客+书籍+总结，前三者博主淋漓尽致地挥毫于这篇博客文章...

不就是SELECT COUNT语句吗，竟然能被面试官虐的体无完肤

数据库查询相信很多人都不陌生，所有经常有人调侃程序员就是CRUD专员，这所谓的CRUD指的就是数据库的增删...

程序员必须掌握的核心算法有哪些？

由于我之前一直强调数据结构以及算法学习的重要性，所以就有一些读者经常问我，数据结构与算法应该要学习到哪...

TCP三次握手原理

TCP协议 TCP报文格式

面试最后一问：你有什么问题想问我？

尽管，我们之前分享了这么多关于面试的主题： 高薪必备的一些Spring Boot高级面试题 面试必问：设计模式遵循的...

python 程序员进阶之路：从新手到高手的100个模块

在知乎和CSDN的圈子里，经常看到、听到一些 python 初学者说，学完基础语法后，不知道该学什么，学了也不知...

阅读数 1万+	博文
阅读数 5万+	博文
阅读数 6万+	博文
阅读数 10万+	博文
阅读数 3391	博文
阅读数 1万+	博文
阅读数 3万+	博文
阅读数 3万+	博文
阅读数 1万+	博文
阅读数 1381	博文
阅读数 1930	博文
阅读数 1万+	博文
阅读数 4万+	博文
阅读数 526	博文
阅读数 2万+	博文
阅读数 2万+	博文

大学四年，看过的优质书籍推荐

有时有些读者问我，数据结构与算法该怎么学？有书籍推荐的吗？Java 初学者该怎么学等等。今天我就给大家介绍一...

Python——画一棵漂亮的樱花树（不同种樱花+玫瑰+圣诞树喔）

最近翻到一篇知乎，上面有不少用Python（大多是turtle库）绘制的树图，感觉很漂亮，我整理了一下，挑了一些我...

还在收集资料？我这里有github汇总

原创：小姐姐味道（微信公众号ID：xjjdog），欢迎分享，转载请保留出处。国内程序员都喜欢收集资料，但是又不...

程序员不懂浪漫？胡扯！

程序员男朋友你的程序员男朋友为你做过什么暖心的事情呢？我的男朋友是一个程序员，他有很多大家在网络上吐槽...

程序员成长的四个简单技巧，你 get 了吗？

最近拜读了“阿里工程师的自我修养”手册，12 位技术专家分享生涯感悟来帮助我们这些菜鸡更好的成长，度过中...

Hadoop技术(一)分布式文件系统HDFS

Hadoop介绍 HDFS理论 HDFS集群搭建 CLI使用 API使用 第一章 统一思维 单机大数据处理实现以及问题 问题引入 ...

Java 8：一文掌握 Lambda 表达式

本文将介绍 Java 8 新增的 Lambda 表达式，包括 Lambda 表达式的常见用法以及方法引用的用法，并对 Lambda ...

Excel，python全都靠边站，这才是数据分析应该有的样子

前段时间，老同学居然和我说她在学编程。当时我很不理解，她是市场专员，又不是程序员，为啥要学编程？直到上...

HTTP协议详解(基础概念 方法 状态码 首部 连接 Cookie 新特性 安全)

一、基础概念 URI URI 包含 URL 和 URN。请求和响应报文 1. 请求报文 2. 响应报文 二、HTTP 方法 客户端发送的...

Python中几个非常有趣的模块

最近学习Python，发现了许多有趣的模块。感觉开启了新世界的大门，因为我也不是对所有模块都熟悉，所以今天不...

2020年秋招后端面经

主要包含华为，网易互娱，广联达，科大讯飞，浦发，中兴，上海农商行这些已经拿offer的，还有京东（不小心把二...

一图看懂py2/py3编码

无论是py2还是py3，都使用unicode作为内存编码，简称内码。保存在python解释器内存中的文本，输出到屏幕、...

Linux/C/C++ 不可错过的好书

来源：公众号【编程珠玑】作者：守望先生 ID：shouwangxiansheng 前言 经常有读者让我推荐书籍，这次我就把...

Python自动化完成tb喵币任务

2019双十一，tb推出了新的活动，商店喵币，看了一下每天都有几个任务来领取喵币，从而升级店铺赚钱，然而我既...

Python 爬虫实现增加播客访问量

一、序言： 世界 1024 程序员节日不加班，闲着没事儿。。。随手写了个播客访问量爬虫玩玩，每个步骤注释都...

十年磨一剑? Linux QQ 终于发布了新版本！

前言 十年磨一剑！Linux QQ 终于发布了 2.0.0 Beta 版本。今天 腾讯QQ 悄然发布了 Linux QQ 测试版 。不过好歹...

从月薪3K的中专生，到身家千万的CTO！人生最大的对手，就是自己

关注 “技术领导力” 博客，独家大厂干货推送 文/Daniel.W David坐在我对面，窗外是梦境般的外滩夜景，繁星点点...

JDK8 新特性：判空遍历写法

在工作中，经常会处理各种变量，但往往会在使用变量的时候，要进行空判断，不然要报错。Java 8 提供了判空写...

Java HashMap 透析

HashMap 相关概念 HashTab、HashMap、TreeMap 均以键值对像是存储或操作数据元素。HashTab继承自Dicti...

Python全栈（一）基础之6.条件控制语句练习

主要对条件控制语句进行了更深入地讲解，包括一些典型的应用，如求素数、打印九九乘法表等，并在文章最后分析...

阅读数 2万+

博文

38

阅读数 7478

博文

1

阅读数 9399

博文

1

阅读数 8806

博文

1

阅读数 1万+

博文

1

阅读数 745

博文

1

阅读数 8026

博文

阅读数 9470

博文

阅读数 3980

博文

阅读数 1万+

博文

阅读数 1671

博文

阅读数 6074

博文

阅读数 1651

博文

阅读数 5739

博文

阅读数 937

博文

阅读数 1万+

博文

阅读数 8225

博文

阅读数 5208

博文

阅读数 5956

博文

阅读数 4803

博文

1

1

堆排序c# c# 标注对象字段后 c# 抓包修改 c#窗体模版 c#让线程重启 c#外链发短信 c#索引器的实现过程 c#函数名后尖括号 c# 外部类
调用控件报错 c# 接口 接收大量数据



38



1



©2019 CSDN 皮肤主题: 编程工作室 设计师: CSDN官方博客



GNG

TA的个人主页 >

私信

关注

原创

117

粉丝

263

获赞

353

评论

116

访问

64万+

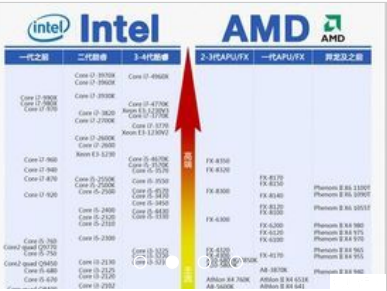
等级: 博客 6

周排名: 2290

积分: 5026

总排名: 8931

勋章:



至强cpu天梯图

最新文章

Mac开发环境搭建

寻找工作的意义——《你在为谁工作》读后感

单例模式的7种实现总结

面经——2019求职修行记

深入理解volatile关键字

归档

2019年10月	2篇
2019年9月	8篇
2019年8月	2篇
2019年6月	3篇
2019年5月	4篇
2019年4月	3篇
2019年1月	6篇
2018年12月	1篇

展开

分类专栏

	MySQL数据库	7篇
	《什么是数学》	7篇
	《算法导论》笔记	8篇
	spring学习总结	



算法导论

10篇

展开

热门文章

"Windows7 内部版本7601 此Windows副本不是正版"解决办法

阅读数 77549

Visual Studio 2015中ucrtbased.dll丢失

阅读数 58159

算法导论-----递归算法的时间复杂度求解

阅读数 45882

Notepad++的实用技巧

阅读数 36449

windows7下TexLive2016的安装和配置

阅读数 28007

最新评论

算法导论-----最长公共子序列L...

qq_20013667: 如果要求取全部最长子序列, 而且同时有多个等长的最长子序列该怎么办?

Notepad++的实用技巧

qq_32573251: 赞

Visual Studio 201...

wxqitwizard: 谢谢! 用方案三完美解决

算法导论-----渐近记号O、O...

so_geili: [reply]likeyou1314918273[/reply] 谢谢反馈, 写错了。已修复。

算法导论-----渐近记号O、O...

likeyou1314918273: 第一个 渐近紧确界记号 T2 (n)你写的是n的四次方。难道不是三次方吗? ...

賢い経営者は、カードで差

■ 経費精算の効率化

■ キャッシュフロー改善

■ 魅力的な特典・優待

■ コンシェルジュサービス

BLAC

5214 1234

000000

A.L. TAYLOR

LUXURY CARD.

n

詳しくはこちらから >

EDU

学院

CSDN学院

JOB

CSDN企业招聘

QQ客服

kefu@csdn.net

客服论坛

400-660-0108

工作时间 8:30-22:00

关于我们

招聘

广告服务

网站地图

百度提供站内搜索

京ICP备19004658号

©1999-2019 北京创新乐知网络技术有限公司

网络110报警服务

经营性网站备案信息

北京互联网违法和不良信息举报中心

中国互联网举报中心

家长监护

版权申诉

38

1

赏