天勤计算机考研高分笔记系列
天勤计算机考研系列

2013NIAN
JISUANJI ZHUANYE JICHU
ZONGHE KAOSHI
XITI XIANGJIE Ⅱ

2013年
计算机专业基础
综合考试习题详解Ⅱ

（数据结构+操作系统）

周伟　率辉　王征勇　刘泱◎等编
清航考研培训教学组老师◎审核

天勤论坛

天勤论坛，取名自古训“天道酬勤”，意为考研路上，困苦实多，然而天自有道，勤恳付出者，必有应得之酬劳。天勤论坛由浙大、北航等多所计算机专业名校的研究生创办，团队所有成员皆亲身经历过计算机专业考研的磨炼，于是本着为考生服务的热情，共同搭建了此交流平台。

由天勤论坛组编的高分笔记系列计算机考研辅导书，融入了论坛答疑的精华内容，论坛组织了高分考生进行勘误，不断完善此套书籍。考生在书中遇到疑问[redacted]流。

为提高考生算法设计能力，团队[redacted]考研学子的在线算法测试平台——ACM俱乐部（acmclub.com），希望能借此帮助考生提高复习效率。

更多计算机
考研和学习交流
尽在www.csbiji.com

机械工业出版社
CHINA MACHINE PRESS

天勤计算机考研系列

天勤计算机考研高分笔记系列

2013年计算机专业基础综合考试习题详解Ⅱ

（数据结构+操作系统）

周 伟 率 辉 王征勇 刘 泱 等编

机械工业出版社

本书分为Ⅰ、Ⅱ两册，第Ⅰ册包括计算机组成原理和计算机网络两个科目；第Ⅱ册包含数据结构和操作系统两个科目。本书所选习题，紧密围绕教育部考试中心发布的考试大纲，并以梯度的形式呈现给读者（从基础题进阶到拔高题），使考生的学习更具有针对性。

另外，本书作者对统考近四年来的真题所考查的知识点进行了深入剖析，在每章的最前面都给出了本章节的考点大预测，使得考生可以有重点地进行复习，提高复习效率。

考生对此书有任何疑问都可以通过天勤论坛（www.csbiji.com）与作者进行在线交流，最大化地提高复习效率。

本书可作为参加计算机专业研究生入学考试的复习指导用书，也可作为全国各高校计算机专业或非计算机专业的学生学习相关课程的辅导用书。

图书在版编目（CIP）数据

2013年计算机专业基础综合考试习题详解Ⅱ. 数据结构+操作系统 / 周伟等编. —北京：机械工业出版社，2012.7
（天勤计算机考研系列）
ISBN 978-7-111-39246-0

Ⅰ. ①2… Ⅱ. ①周… Ⅲ. ①数据结构—研究生—入学考试—题解②操作系统—研究生—入学考试—题解 Ⅳ. ①TP3-44

中国版本图书馆CIP数据核字（2012）第170868号

机械工业出版社（北京市百万庄大街22号 邮政编码 100037）
策划编辑：吉 玲 责任编辑：吉 玲 李 宁 范成欣 徐 凡
封面设计：鞠 杨 责任印制：杨 曦
保定市中画美凯印刷有限公司印刷
2012年8月第1版第1次印刷
184mm×260mm • 17.25印张 • 435千字
标准书号：ISBN 978-7-111-39246-0
定价：38.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务	网络服务
社服务中心：（010）88361066	教材网：http://www.cmpedu.com
销售一部：（010）68326294	机工官网：http://www.cmpbook.com
销售二部：（010）88379649	机工官博：http://weibo.com/cmp1952
读者购书热线：（010）88379203	**封面无防伪标均为盗版**

序

欣看《2013 版数据结构高分笔记》、《2013 版计算机组成原理高分笔记》、《2013 版操作系统高分笔记》、《2013 版计算机网络高分笔记》、《2013 年计算机专业基础综合考试习题详解Ⅰ（计算机组成原理+计算机网络）》、《2013 年计算机专业基础综合考试习题详解Ⅱ（数据结构+操作系统）》等 6 本辅导教材问世了，这对于有志考研的同学是一大幸事。“它山之石，可以攻玉”，参考一下亲身经历过考研，并取得优秀成绩的师兄们的经验，必定有益于对考研知识点的复习和掌握。

能够考上研究生，这是无数考生的追求，能够以优异的成绩考上名牌大学的全国数一数二的计算机或软件工程学科的研究生，更是许多考生的梦想。如何学习或复习相关课程，如何打好扎实的理论基础、练好过硬的实践本领，如何抓住要害，掌握主要的知识点并获得考试的经验，先行者已经给考生们带路了。“高分笔记”的作者们在认真总结了考研体会，整理了考研的备战经验，参考了多种考研专业教材后，精心编写了系列辅导书。

“天勤计算机考研高分笔记系列”辅导教材的特点是：

✧ 贴近考生。作者们都亲身经历了考研，他们的视角与以往辅导教材不同，是从复习考研的学生的立场理解教材的知识点——哪些地方理解有困难，哪些地方需要整理思路，叙述处处替考生着想，有很好的引导作用。

✧ 重点突出。作者们在复习过程中做了大量习题，并经历了考研的严峻场面，对重要的知识点，考试出现频率高的题型都了如指掌。因此，在复习内容的取舍上进行了精细的考虑，使得读者可以抓住重点，有效地复习。

✧ 分析透彻。作者们在复习过程中对主要辅导教材的许多习题都深入分析并实践过，对重要知识点做过相关实验并有总结。因此，解题思路明确，叙述条理清晰，对问题求解的步骤和结果的分析透彻，不但可以扩展考生思路，还有助于考生举一反三。

计算机专业综合基础考试已经考过 4 年，今后考试的走向如何，这可能是考生最关心的问题了。我想，这要从考试命题的规则入手来讨论。

以清华大学为例，学校把研究生入学考试定性为选拔性考试。研究生入学考试试题主要测试考生对本学科的专业基础知识、基本理论和基本技能掌握的程度。因此，出题范围不应超出本科教学大纲和硕士生培养目标，并尽可能覆盖一级学科的知识面，一般会使本学科、本专业本科毕业的优秀考生能取得及格以上的成绩。

实际上，全国计算机专业研究生入学联考的命题原则也是如此，各学科的重点知识点都是命题的重点。一般知识要考，比较难的知识（较深难度的知识）也要考。从 2009 年以来几年的考试分析可知，考试的出题范围基本符合考试大纲，都覆盖到各大知识点，但题量有所侧重。因此，考试一开始不要抱侥幸的心理去押题，应踏踏实实读好书，认认真真做好复习题，仔仔细细归纳问题解决的思路，夯实基础，增长本事；然后再考虑重点复习，有几条规律可供参考：

✧ 出过题的知识点还会有题，出题频率高的知识点，今后出题的可能性也大。

✧ 选择题大部分题目涉及基本概念，主要考查各个知识点的定义、特点的理解，个别选择题会涉及相应延伸的概念。

✧ 综合应用题分为两部分：简作题和设计题。简作题的重点在设计和计算；设计题的重点在算法、实验或综合应用。

常言道："学习不怕根基浅，只要迈步总不迟"，只要大家努力了，收获总会有的。

清华大学　殷人昆

2012年6月

前　言

众所周知，在考研界流传着一句话，“只有考计算机专业的研究生，才叫真正经历过考研”。忽略其他因素，单从知识层面来考虑，我对这句话的解读是这样的，其蕴义并不是说计算机专业考研的知识点本身理解有多难，而是知识点太过于广泛，看了这科忘记那科，如此进入一个恶性循环。所以说在短短几个月的复习时间里，要想全面掌握大纲所要求的知识点是非常困难的，何况还要将其灵活运用去应对综合性越来越高的考研真题，就更是难上加难了。

作为过来人，我虽然不是什么高分学长，但是我一直希望能将自己备考专业课的方法与全国的考生们分享，至少这种方法已经被身边的研友认可，即用生活来解释考研。于是，复试之后就抱着一种尝试的态度，写出了第一本高分笔记系列考研辅导书——《计算机网络高分笔记》。当时，拿着成稿心里还真是挺忐忑，这种完全突破传统书籍的写作风格考生会不会喜欢？或许恰好就适合我身边的同学。幸运的是，几乎所有给《计算机网络高分笔记》第 1 版勘误的同学都给予它很高的评价，这也给予了我极大的信心坚持把这件事情做下去。

今年正式出版的是高分笔记系列书籍的第 3 版，在前两版中，考生在知识点讲解方面给予了我们很多启发，使得知识点讲解越来越完善。但是，作为单科辅导书，其不足也是显而易见的。每章的章节习题无法覆盖到所有考点、难点、易混淆点，使得一些知识点仍然停留在理论层面上，无法与实际进行结合。针对以上单科书的不足，天勤论坛编写团队精心组编了《计算机专业基础综合考试习题详解》。该习题集分为Ⅰ、Ⅱ两册，第Ⅰ册包括“计算机组成原理”与“计算机网络”两科，第Ⅱ册包括“数据结构”与“操作系统”两科。本书特点如下。

（1）**量身定做**：本书所有习题完全按照大纲所要求的知识点进行选题、编题，特别是针对Ⅰ、Ⅱ、Ⅲ这种变相多选题形式，可谓是历年真题必考题型。本书编写团队仿造这种出题思路，编写了很多高质量的习题，让考生时时刻刻保持一种做真题的感觉。

（2）**梯度分类**：将各章习题进行梯度分类，包括基础题与拔高题。基础题的命题思路主要偏于对基础概念的理解，而拔高题更注重综合知识的应用。

（3）**论坛精品**：本书融入了天勤论坛众多高分学子的精华交流内容。并且对于基础薄弱的考生常问的各种难点、易混淆点进行了总结，并将其融入到相应习题的讲解中。

（4）**实时答疑**：考生对于书中的任何疑问，都可通过天勤论坛（www.csbiji.com）与本书作者进行在线交流，大大提高考生复习的效率，达到事半功倍的复习效果。

由于《计算机专业基础综合考试习题详解》是第一次出版，其中肯定存在很多的不足之处，希望考生能够多给予我们批评及建议，只有这样，高分笔记系列书籍才能成长。

参加本书编写的人员有：王勇，王征兴，王征勇，霍宇驰，董明昊，王辉，郑华斌，王长仁，刘泱，刘桐，章露捷，刘建萍，刘炳瑞，刘菁，孙琪，施伟，金苍宏，蔡明婉，吴雪霞，周政强，孙建兴，周政斌，叶萍，周伟，孔蓓，率四杰，张继建，胡素素，邱纪虎，率方杰，李玉兰，率秀颂。

最后，感谢为本书勘误的天勤会员，他们是：刘晓通、曾杰、黄亮、王馨妍、胡迪、李芳、陈人良、安小强、盛守波、周丽存、王俊、丁奥林、吴军、黄欢、李天华、段庆龙、陈

元为、高亚运、洪树鑫、翁蔚涛、黄天祺、孙景润、费凡、马原龙、梁智鹏、王跃、张小牙、刘安柱、杨帆、王旭、刘温格、臧凯、章旭东、刘增明、杨昕、朱瑞茂、袁浩亮。是你们提出的批评性意见让本书的质量上升了一个台阶，再次表示衷心的感谢，祝你们金榜题名！

周伟

2012年6月

目　录

下篇 操作系统

上篇　数据结构

第 1 章　算法复杂度相关问题专练

说明：对于算法复杂度的评估问题，已经是每年必考的重点，因此本书将其独立作为一章，通过一些具体的例题进行有针对性的讲解，来帮助考生在解决此类问题时形成系统的套路，以应对多变的考研题目。

如果考生已经读过《数据结构高分笔记》绪论部分，则可以直接做算法复杂度综合题目专练。

对于这部分，要牢记一句话：**将算法中基本操作的执行次数作为算法时间复杂度的度量。**这里所讨论的时间复杂度，不是执行完一段程序的总时间，而是其中基本操作的总次数。因此对于一个算法进行时间复杂度分析的要点，无非是明确算法中哪些操作是基本操作，然后计算出基本操作所重复执行的次数。在考试中算法题目里总能找到一个 n，可以称为问题的规模，如要处理的数组元素的个数为 n，而基本操作所执行的次数是 n 的一个函数 f(n)（这里的函数是数学中的函数的概念，不是 C 或 C++语言中的函数的概念）。对于求其基本操作执行的次数，就是求函数 f(n)。求出以后就可以取出 f(n)中随 n 增大而增长最快的项，然后将其系数变为 1，作为时间复杂度的度量，记为 **T(n)=O（f(n)中增长最快的项/此项的系数）**，如 $f(n)=2n^3+4n^2+100$，则其时间复杂度为 $T(n)=O(2n^3/2)=O(n^3)$。其实计算算法的时间复杂度，就是给出相应的数量级，当 f(n)与 n 无关时，时间复杂度 T(n)=O(1)；当 f(n)与 n 是线性关系时，T(n)=O(n)；当 f(n)与 n 是平方关系时，$T(n)=O(n^2)$；以此类推。

说明：考研中常常要比较各种时间复杂度的大小，常用的比较关系如下。

$$O(1)\leqslant O(\log_2(n))\leqslant O(n)\leqslant O(n\log_2(n))\leqslant O(n^2)\leqslant O(n^3)\leqslant\cdots\leqslant O(n^k)\leqslant O(2^n)$$

通过以上分析总结出计算一个算法时间复杂度的步骤如下：

1）确定算法中的基本操作以及问题的规模。

2）根据基本操作执行情况计算出规模 n 的函数 f(n)，并确定时间复杂度为 T(n)=O(f(n)中增长最快的项/此项的系数)。

注意：有的算法中基本操作执行次数不仅仅跟初始输入的数据规模有关，还和数据本身有关，如一些排序算法，同样 n 个待处理数据，数据初始有序性不同，基本操作执行次数也会不同。如果题目不做特殊要求，一般依照使得基本操作执行次数最多的输入来计算时间复杂度，即将最坏的情况作为算法时间复杂度的度量。

例题 1：求出以下算法的时间复杂度。

```
void fun(int n)
{
    int i=1,j=100;
    while(i<n)
```

```
    {
        ++j;
        i+=2;
    }
}
```

分析：

1）找出基本操作，确定规模n。

① 找基本操作（所谓基本操作，即其重复执行次数和算法的执行时间成正比的操作。通俗点说，这种操作组成了算法，当它们都执行完的时候算法也结束了，**多数情况下取最深层循环内的语句所描述的操作作为基本操作**），显然题目中++j;语句与 i+=2;语句都可以作为基本操作。

② 确定规模，由循环条件i<n可以知道，循环执行的次数即基本操作执行的次数和参数n有关，因此参数n就是规模n。

2）计算出n的函数f(n)。

显然，n确定以后，循环的结束与否与i有关。i的初值为1，每次自增2，假设i自增m次后循环结束，则i最后的值为1+2×m，因此有1+2×m+K=n（其中K为一个常数，在循环结束时i的值稍大于n，为了方便表述和进一步计算，用K将1+2×m修正成n。因为K为常数，所以这样做不会影响最终时间复杂度的计算），解得 m=(n−1−K)/2，即 f(n)=(n−1−K)/2。可以发现其中增长最快的项为n/2，因此时间复杂度T(n)=O(n)。

例题2：分析以下算法的时间复杂度。

```
void fun(int n)
{
    int i,j,x=0;
    for(i=1;i<n;++i)
        for(j=i+1;j<=n;++j)
            ++x;
}
```

分析：

++x;语句处于最内层循环，因此取++x;语句作为基本操作。显然n为规模，可以算出++x;语句的执行次数为f(n)=n(n−1)/2，变化最快的项为n^2，因此时间复杂度$T(n)=O(n^2)$。

例题3：分析以下算法的时间复杂度。

```
void fun(int n)
{
    int i=0;s=0;
    while(s<n)
    {
        ++i;
        s=s+i;
    }
}
```

分析：

显然 n 为规模，基本操作为++i;语句和 s=s+i;语句。i 与 s 都从 0 开始，假设循环执行 m 次结束，则有 $s_1=1$，$s_2=1+2=3$，$s_3=1+2+3=6$，…，$s_m=m(m+1)/2$（其中 s_m 为执行到第 m 次时 s 的值），则有 $m(m+1)/2+K=n$（K 为起修正作用的常数），由求根公式得

$$m=\frac{-1+\sqrt{8n+1-8k}}{2}$$

即

$$f(n)=\frac{-1+\sqrt{8n+1-8k}}{2}$$

由此可知时间复杂度为

$$T(n)=O(\sqrt{n})$$

说明：在计算时间复杂度的时候有可能会出现这种情况，即对于相同的规模，因输入序列不同会出现不同的时间复杂度，这时一般取最坏的情况下的输入序列（即使得基本操作执行次数最多的序列）来计算时间复杂度。

算法复杂度综合题目专练

1．设 n 为如下程序段处理的数据个数，求其时间复杂度。

```
for(i=1;i<=n;i=2*i)
    cout<<"i="<<i<<endl;
```

2．计算 n!的递归函数 fac(n)如下，试分析它的时间复杂度。

```
int fac(int n)
{
    if(n<=1)
        return 1;                    //①
    else
        return(n*f(n-1));            //②
}
```

3．设计一个算法用不多于 3n/2 的平均比较次数，在数组 A[1..n]中找出最大值和最小值的元素。

4．设 A 是一个线性表（a_1，a_2，…，a_n），采用顺序存储结构，则在等概率的前提下，平均插入一个元素需要移动的元素个数是多少？若元素插入在 a_i 与 a_{i+1} 之间（$0\leqslant i\leqslant n-1$）的概率为 $\frac{n-i}{n(n+1)/2}$，则平均每插入一个元素所要移动的元素个数是多少？

5．分析以下各程序段的时间复杂度。

```
(1) void main()
    {
        int i=1,k=0,n=10;
        while(i<=n-1)
        {
```

```
            k+=10*i;
            i++;
        }
    }
(2) void main()
    {
        int i=1,k=0,n=100;
        do
        {
            k+=10*i;i++;
        } while(i==n);
    }
(3) void main()
    {
        int i=1,j=0,n=10;
        while(i+j<=n)
            if(i>j)
                j++;
            else
                i++;
    }
(4) void main()
    {
        int n=10,x=n,y=0;
        while(x>=(y+1)*(y+1))
            y++;
    }
(5) void main()
    {
        int n=9,i=1;
        while(i<=n)
            i=i*3;
    }
```

6. 分析以下程序段的时间复杂度。

```
……
i=1;
while(i<=n)
    i=i*2;
……
```

7. 假设n为2的乘幂，如n=2，4，8，16，…，试求下列程序的时间复杂度及变量count

的值（以 n 的函数形式表示）。

```
void counter()
{
    int n,x,count;
    cout<<"n: ";
    cin>>n;
    count=0;
    x=2;
    while(x<n/2)
    {
        x=2*x;
        count++;
    }
    cout<<count<<endl;
}
```

8．某算法所需时间由下述方程表示，试求出该算法的时间复杂度（以大 O 形式表示）。注意，n 为求解问题的规模，为简单起见，设 n 是 2 的正整数幂。

$$T(n)=\begin{cases}1 & \text{当} n=1\\ 2T\left(\dfrac{n}{2}\right)+n & \text{当} n>1\end{cases}$$

9．分析 order 函数的时间复杂度。

```
order(int j,int m)
{
    int i,temp;
    if(j<m)
    {
        for(i=j;i<=n;i++)
            if(a[i]<a[j])
            {
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        j++;
        order(j,m);                    //递归调用
    }
}
```

10．斐波那契数列 F_n 定义如下：$F_0=0$，$F_1=1$，$F_n=F_{n-1}+F_{n-2}$，n=2，3，…，就此斐波那契数列，回答下列问题。

（1）在递归计算 F_n 时，需要对较小的 F_{n-1}，F_{n-2}，…，F_1，F_0 精确计算多少次？

（2）如果用大O表示法，递归计算F_n时递归函数的时间复杂度是多少？

11．设计求解下列问题的算法，并分析其最坏情况的时间复杂度。

（1）在数组A[1..n]中查找值为K的元素，若找到则输出其位置i；否则输出0作为标志。

（2）找出数组A[1..n]中元素的最大值和次最大值。

12．以下算法是在一个有n个数据元素的数组A中删除第i个位置的数组元素，要求当删除成功时数组元素个数减1，求平均删除一个数组元素需要移动的元素个数是多少？其中，数组下标从0至n-1。

```
int delete(int A[],int n,int i)
{
    int j;
    if(i<0||i>n)
        return 0;
    for(j=i+1;j<n;j++)
        A[j-1]=A[j];
    n--;
    return 1;
}
```

算法复杂度综合题目专练答案

1.【答案要点】

其中的基本语句是第2行的cout语句，设它的执行次数为f(n)，则有$2^{f(n)}\leqslant n$，即$f(n)\leqslant \log_2 n$。

本程序段时间复杂度$T(n)=f(n)\leqslant \log_2 n=O(\log_2 n)$。

所以时间复杂度为$O(\log_2 n)$。

2.【答案要点】

设T(n)为fac(n)的时间开销函数。显然T(1)=O(1)。①的时间开销为O(1)，递归调用fac(n-1)的时间开销为T(n-1)；②的时间开销为O(1)+T(n-1)。时间复杂度为

$$\begin{aligned}T(n)&=O(1)+T(n-1)\\&=O(1)+O(1)+T(n-2)\\&=2\times O(1)+T(n-2)\\&=\cdots\\&=(n-1)\times O(1)+T(1)\\&=n\times O(1)=O(n)\end{aligned}$$

3.【答案要点】

本题的算法思想是：如果在查找出最大和最小值的元素时各遍历一遍所有元素，则至少要比较2n次，为此，使用一遍遍历找出最大值和最小值的元素。实现本题功能的函数如下：

```
void maxmin(int A[],int n)
{
    int i;
```

```
    int max,min;
    max=A[1];min=A[1];
    for(i=2;i<=n;i++)
        if(A[i]>max)
            max=A[i];
        else if(A[i]<min)
            min=A[i];
    cout << "max=" << max << ",min=" << min << endl;
}
```

在这个函数中，最坏的情况是A中的元素按递减次序排列，这时（A[i]>max）条件均不成立，比较的次数为n-1。另外，每次都要比较（A[i]<min），同样比较的次数为n-1。因此，总的比较次数为2(n-1)。最好的情况是A中的元素按递增次序排列，这时（A[i]>max）条件均成立，不会再执行else的比较，所以总的比较次数为n-1。

平均比较次数为

$$\frac{2(n-1)+n-1}{2}=\frac{3n}{2}-\frac{3}{2}\leqslant\frac{3n}{2}$$

由此可知，本算法的平均比较次数不多于3n/2。

4.【答案要点】

A=（a_1，a_2，…，a_n），有n+1个位置可插入元素，即a_1之前，a_1与a_2之间，…，a_{n-1}与a_n之间和a_n之后，分别需要移动的元素个数为n，n-1，…，1，0，则平均插入一个元素需要移动的元素个数为

$$(n+(n-1)+\cdots+1+0)\times\frac{1}{n+1}=\frac{n}{2}$$

若元素插入在a_i与a_{i+1}之间（0≤i≤n-1）的概率为$\frac{n-i}{n(n+1)/2}$，其中移动的元素个数为

$$(n-i)\times\frac{n-i}{n(n+1)/2}=\frac{(n-i)^2}{n(n+1)/2}$$

则平均每插入一个元素所要移动的元素个数为

$$\sum_{i=1}^{n-1}\frac{(n-i)^2}{n(n+1)/2}=\frac{2n+1}{3}$$

5.【答案要点】

（1）O(n)；（2）O(1)；（3）O(n)；（4）O(n)；（5）$O(\log_3 n)$。

本题较为简单，这里只讲解（5）。

设执行k次，则有$3^k+C=n$（其中C是起到修正作用的常数），即$k=\log_3(n-C)$。

$\log_3(n-C)$与$\log_3 n$同数量级，因此复杂度为$O(\log_3 n)$。

6.【答案要点】

此处循环体里面是i=i*2，即每循环一次，i值增加一倍，所以执行次数与n之间是以2为底的对数关系，故时间复杂度为$O(\log_2 n)$。

7.【答案要点】

n与count的关系如下：