

数据结构

习题解答与考试指导

梁作娟 胡伟 唐瑞春 编著

S



清华大学出版社

与严蔚敏、吴伟民编著的《数据结构》（C语言版）配套使用

数据结构习题解答与考试指导

梁作娟 胡 伟 唐瑞春 编著

清华大学出版社

北 京

内 容 简 介

本书可作为“数据结构”课程的复习参考书,其内容、要点、题目都是根据该课程的范围和难度来组织的。

全书共分13章。前12章每章的内容编排为基本知识结构图、知识点、习题及参考答案、考研真题分析、自测题5个部分。最后一章精选了近几年的各大名校硕士研究生入学考试题,并附有参考答案。该书内容充实完整,层次分明,概念着眼点细微,知识点清晰,表述方式易于接受,有利于加深学生对“数据结构”课程的理解,帮助学生从广度和深度上把握知识体系,拓宽解题思路,提高解题速度。

本书针对性强,习题覆盖面广,可作为计算机及相关专业“数据结构”课程的学习指导书,也适用于报考硕士研究生的人员作为应试指导书。

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

数据结构习题解答与考试指导/梁作娟,胡伟,唐瑞春编著.

—北京:清华大学出版社,2004.11

ISBN 7-302-09820-4

I. 数… II. ①梁… ②胡… ③唐… III. 数据结构—研究生—入学考试—解题
IV. TP311.12-44

中国版本图书馆CIP数据核字(2004)第111839号

出 版 者:清华大学出版社

地 址:北京清华大学学研大厦

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

客户服务:010-62776969

组稿编辑:科海

文稿编辑:陈洁

封面设计:付剑飞

版式设计:科海

印 刷 者:北京市艺辉印刷有限公司

发 行 者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:27.25 字数:663千字

版 次:2004年11月第1版 2004年11月第1次印刷

书 号:ISBN 7-302-09820-4/TP·6774

印 数:1~5000

定 价:35.00元

本书如存在文字不清、漏印以及缺页、倒角、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:82896445

前 言

“数据结构”课程是计算机及相关专业最重要的专业基础课程之一，是进行程序设计的理论和技术基础，同时对于计算机专业其他课程的学习也是十分重要的。对于广大学生来说，“数据结构”是一门非常有用而难学的课程。本书是由长期在一线从事教学工作、有着丰富授课经验的教师编写而成。

本书的部分内容是与经典教材《数据结构（C语言版）》（清华大学出版社出版，严蔚敏、吴伟民编著）和《数据结构题集（C语言版）》（清华大学出版社出版，严蔚敏、吴伟民、米宁编著）配套使用的。

全书共分13章，分别为绪论、线性表、栈和队列、串、数组和广义表、树和二叉树、图、动态存储管理、查找、内部排序、外部排序、文件、名校试题。

前12章每章的内容编排为基本知识结构图、知识点、习题及参考答案、考研真题分析、自测题5个部分。“基本知识结构图”用框图的形式表示，可帮助学生清晰地归纳本章内容；“知识点”给出了本章应该掌握的主要知识要点；“习题及参考答案”部分给出了《数据结构题集（C语言版）》（清华大学出版社出版，严蔚敏、吴伟民、米宁编著）中的部分习题及答案，并对较难解的题目进行了分析指导，通过这些习题，可以掌握知识点中的概念，并对基础概念能有所提升；“考研真题分析”部分精心选择了近年来的考研真题并给出分析解答；“自测题”部分安排了难度适宜的题目供学生自测，以检验自己的水平。

最后一章“名校试题”中提供了几所名牌大学2002年和2003年的硕士研究生入学考试试题，读者可以进行实战演练。

该书内容充实完整，层次分明，概念着眼点细微，知识点清晰，表述方式易于接受，有利于加深学生对“数据结构”课程的理解，帮助学生从广度和深度上把握知识体系，拓宽解题思路，提高解题速度。

由于作者水平有限，时间仓促，书中出现的一些不足和差错之处在所难免，敬请广大读者批评指正。

作者
2004年10月

目 录

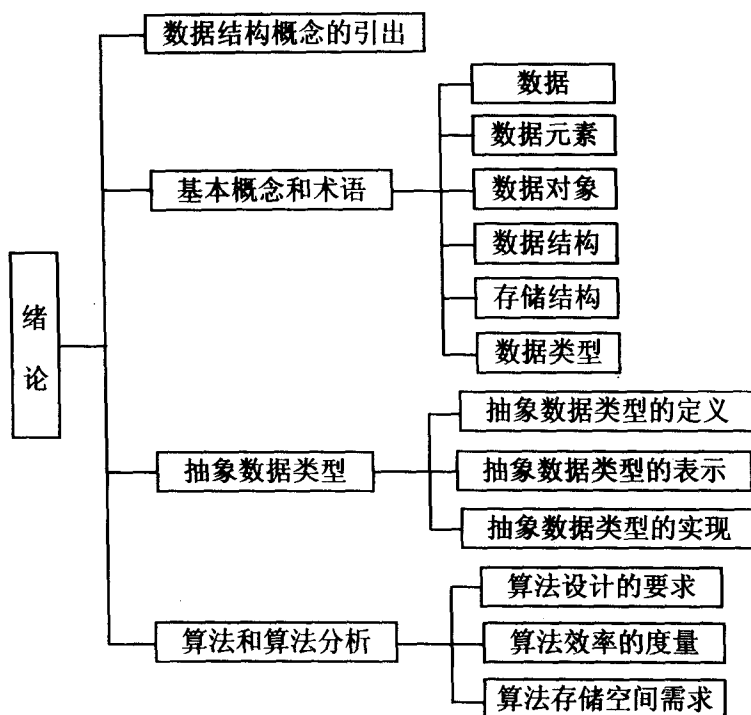
第1章 绪论	1
1.1 基本知识结构图.....	1
1.2 知识点	1
1.3 习题及参考答案.....	2
1.4 考研真题分析	8
1.5 自测题	13
1.6 自测题答案	14
第2章 线性表	16
2.1 基本知识结构图.....	16
2.2 知识点	16
2.3 习题及参考答案.....	17
2.4 考研真题分析	37
2.5 自测题	45
2.6 自测题答案	47
第3章 栈和队列	52
3.1 基本知识结构图.....	52
3.2 知识点	52
3.3 习题及参考答案.....	53
3.4 考研真题分析	70
3.5 自测题	78
3.6 自测题答案	80
第4章 串	82
4.1 基本知识结构图.....	82
4.2 知识点	82
4.3 习题及参考答案.....	83
4.4 考研真题分析	96
4.5 自测题	99
4.6 自测题答案	101
第5章 数组和广义表	103
5.1 基本知识结构图.....	103
5.2 知识点	103
5.3 习题及参考答案.....	104

5.4 考研真题分析	121
5.5 自测题	127
5.6 自测题答案	128
第6章 树和二叉树	133
6.1 基本知识结构图	133
6.2 知识点	133
6.3 习题及参考答案	135
6.4 考研真题分析	164
6.5 自测题	190
6.6 自测题答案	194
第7章 图	204
7.1 基本知识结构图	204
7.2 知识点	204
7.3 习题及参考答案	206
7.4 考研真题分析	236
7.5 自测题	252
7.6 自测题答案	257
第8章 动态存储管理	264
8.1 基本知识结构图	264
8.2 知识点	264
8.3 习题及参考答案	265
8.4 考研真题分析	271
8.5 自测题	274
8.6 自测题答案	274
第9章 查找	276
9.1 基本知识结构图	276
9.2 知识点	276
9.3 习题及参考答案	278
9.4 考研真题分析	296
9.5 自测题	312
9.6 自测题答案	316
第10章 内部排序	324
10.1 基本知识结构图	324
10.2 知识点	324
10.3 习题及参考答案	325
10.4 考研真题分析	344
10.5 自测题	353

10.6 自测题答案	358
第11章 外部排序	361
11.1 基本知识结构图	361
11.2 知识点	361
11.3 习题及参考答案	362
11.4 考研真题分析	362
11.5 自测题	364
11.6 自测题答案	365
第12章 文件	367
12.1 基本知识结构图	367
12.2 知识点	367
12.3 考研真题分析	367
12.4 自测题	368
12.5 自测题答案	369
第13章 名校试题	370
哈尔滨工业大学2002年硕士研究生入学考试试题	370
参考答案	372
华北计算技术研究所2002年硕士研究生入学考试试题	378
参考答案	381
华中科技大学2002年硕士研究生入学考试试题	384
参考答案	387
南开大学2002年硕士研究生入学考试试题	389
参考答案	391
南京大学2003年硕士研究生入学考试试题	394
参考答案	396
武汉理工大学2003年硕士研究生入学考试试题	399
参考答案	402
复旦大学2003年硕士研究生入学考试试题	406
参考答案	409
华东师范大学2003年硕士研究生入学考试试题	411
参考答案	413
北京邮电大学2003年硕士研究生入学考试试题	415
参考答案	418
北京科技大学2003年硕士研究生入学考试试题	420
参考答案	423

第1章 绪 论

1.1 基本知识结构图



1.2 知 识 点

1. 基本概念和术语

数据是对客观事物的符号表示，是计算机程序加工的“原料”。

数据元素是数据的基本单位，可由若干个数据项组成，数据项是数据的不可分割的最小单位。

数据对象是性质相同的数据元素的集合，是数据的一个子集。

数据结构是相互之间存在一种或多种特定关系的数据元素的集合。基本结构有4类：集

合、线性结构、树形结构、图状结构或网状结构。

存储结构即数据的物理结构，是数据结构在计算机中的表示，包括数据元素的表示和关系的表示。主要有顺序存储结构、链式存储结构、索引存储方法和散列存储方法等。

数据类型是一个值的集合和定义在这个值集上的一组操作的总称。

抽象数据类型指一个数学模型以及定义在该模型上的一组操作，仅取决于它的一组逻辑特性，而与其在计算机内部如何表示和实现无关。可通过固有数据类型来表示和实现，即利用处理器中已存在的数据类型来说明新的结构，用已实现的操作来组合新的操作。

2. 算法的定义和算法的时间、空间复杂度

算法是对特定问题求解步骤的一种描述，是指令的有限序列，其中每一条指令表示一个或多个操作。算法具有有穷性、确定性、可行性、输入和输出5个特性。“好”的算法应具有正确性、可读性、健壮性、高效率与低存储量需求。

一般情况下，算法中基本操作重复执行的次数是问题规模 n 的某个函数 $f(n)$ ，算法的时间量度记作 $T(n)=O(f(n))$ ， $T(n)$ 就为该算法的时间复杂度。有时要考虑平均时间复杂度和最坏时间复杂度。

空间复杂度是算法所需存储空间的量度，记作 $S(n)=O(f(n))$ ，其中 n 为问题的规模。

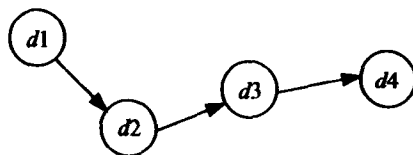
1.3 习题及参考答案

1. 试描述数据结构和抽象数据类型的概念与程序设计语言中数据类型概念的区别。

【解答】简单地说，数据结构定义了一组按某些关系结合在一起的数组元素。数据类型不仅定义了一组带结构的数据元素，而且还在其上定义了一组操作。程序设计语言中的数据类型是一个值的集合和定义在这个值集上的一组操作的总称。而抽象数据类型是指一个数学模型以及定义在该模型上的一组操作。

2. 设有数据结构 (D, R) ，其中 $D=\{d1, d2, d3, d4\}$ ， $R=\{r\}$ ， $r=\{(d1, d2), (d2, d3), (d3, d4)\}$ 。试按图论中图的画法惯例画出其逻辑结构图。

【解答】



3. 试画出与下列程序段等价的框图。

```

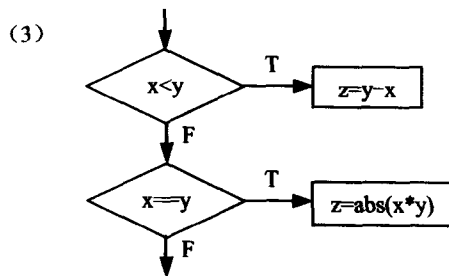
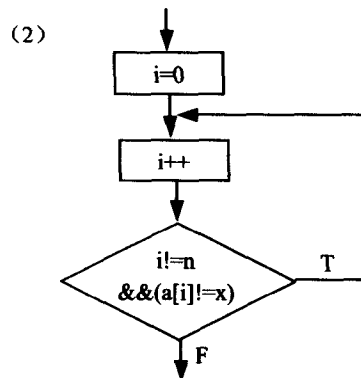
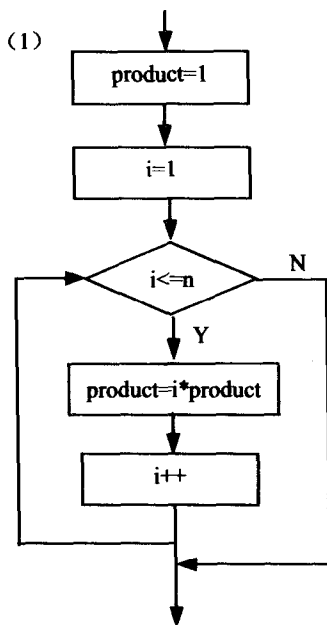
(1) product=1; i=1;
    while (i<=n) {
        product *=i;
        i++;
    }
  
```

```

    }
(2) i=0;
    do{
        i++;
    } while ((i!=n)&&(a[i]!=x));
(3) switch{
    case x<y:z=y-x;break;
    case x==y:z=abs(x*y);break; //abs()为取绝对值函数
    default:z=(x-y)/abs(x)*abs(y);
}

```

【解答】



4. 设 n 为正整数。试确定下列各程序段中前置以记号@的语句的频度。

- (1) $i=1; k=0;$
`while (i<=n-1){`
 @ $k+=10*i;$
 $i++;$
`}`
- (2) $i=1; k=0;$
`do {`
 @ $k+=10*i;$
 $i++;$
`} while(i<=n-1);`
- (3) $i=1; k=0;$
`while (i<=n-1) {`

```

        i++;
        @ k+=10*i;
    }
(4) k=0;
    for (i=1;i<=n;i++) {
        for (j=i;j<=n;j++)
            @ k++;
    }
(5) for (i=1;i<=n;i++) {
        for (j=1;j<=i;j++){
            for (k=1;k<=j;k++)
                @ x+=delta;
        }
    }
(6) i=1;j=0;
    while (i+j<=n){
        @ if (i>j) j++;
        else i++;
    }
(7) x=n; y=0; //n是不小于1的常数
    while (x>=(y+1)*(y+1)) {
        @ y++;
    }
(8) x=91; y=100;
    while (y>0){
        @ if (x>100) {x-=10;y--;}
        else x++;
    }

```

【解答】 (1)、(2) 和 (3) 三个程序段中任何两段都不等效；程序段 (8) 取自著名的 McCarthy 91 函数

$$M(x) = \begin{cases} x-10 & x > 100 \\ M(M(x+1)) & x \leq 100 \end{cases}$$

对任何 $x \leq 100$, $M(x)=91$ 。此程序实质上是一个双重循环，对每个 $y(>0)$ 值，@ 语句执行 11 次，其中 10 次是执行 $x++$ 。

5. 假设 n 为 2 的乘幂，并且 $n > 2$ ，试求下列算法的时间复杂度及变量 count 的值（以 n 的函数形式表示）。

```

int Time (int n){
    count=0;x=2;
    while(x<n/2){
        x *=2; count++;
    }
    return(count);
} //Time

```

【解答】时间复杂度为 $\log_2 n$ ，变量count的值为 $\log_2 n - 1$ 。

6. 按增长率由小至大的顺序排列下列各函数。

2^{100} , $(3/2)^n$, $(2/3)^n$, $(4/3)^n$, n^n , $n^{3/2}$, $n^{2/3}$, $\sqrt{n}$, $n!$, n , $\log_2 n$, $n/\log_2 n$, $(\log_2 n)^2$, $\log_2(\log_2 n)$, $n \log_2 n$, $n^{\log_2 n}$

【解答】各函数的排列次序如下：

$(2/3)^n$, 2^{100} , $\log_2(\log_2 n)$, $\log_2 n$, $(\log_2 n)^2$, $\sqrt{n}$, $n^{2/3}$, $n/\log_2 n$, n , $n \log_2 n$, $n^{3/2}$, $(4/3)^n$, $(3/2)^n$, $n^{\log_2 n}$, $n!$, n^n

7. 已知有实现同一功能的两个算法，其时间复杂度分别为 $O(2^n)$ 和 $O(n^{10})$ ，假设现实计算机可连续运算的时间为 10^7 秒（100多天），又每秒可执行基本操作（根据这些操作来估算算法时间复杂度） 10^5 次。试问在此条件下，这两个算法可解问题的规模（即 n 值的范围）各为多少？哪个算法更适宜？

【解答】这两个算法可解问题的规模各为 $n \leq \log_2 10^{12}$ 和 $n^{10} \leq 10^{12}$ 。结论是第一个算法较适宜。由此可见，虽然一般情况下多项式阶的算法优于指数阶的算法，但高次多项式的算法在 n 的很大范围内不如某些指数阶的算法。

8. 试设定若干 n 值，比较两函数 n^2 和 $50n \log_2 n$ 的增长趋势，并确定 n 在什么范围内，函数 n^2 的值大于 $50n \log_2 n$ 的值。

【解答】大约在 $n > 450$ 时，函数 n^2 的值才大于函数 $50n \log_2 n$ 的值。

9. 试写一算法，自大至小依次输出顺序读入的3个整数 X 、 Y 和 Z 的值。

【分析】假设我们仍依 X 、 Y 和 Z 的次序输入这3个整数，则此题的目标是做到 $X \geq Y \geq Z$ 。在算法中应考虑对这3个元素作尽可能少的比较和移动，如下述算法在最坏的情况下只需进行3次比较和7次移动。

【解答】算法如下：

```
void Descending() {
    scanf(x, y, z);
    if (x < y)
        {temp=x; x=y; y=temp;} //使x≥y
    if (y < z) {
        temp=z; z=y;          //使temp≥z
        if (x ≥ temp) y=temp;
        else {y=x; x=temp;}
    }
    printf(x, y, z);
} //Descending
```

10. 已知 k 阶斐波那契序列的定义为

$$f_0=0, f_1=0, \dots, f_{k-2}=0, f_{k-1}=1;$$

$$f_n=f_{n-1}+f_{n-2}+\dots+f_{n-k}, n=k, k+1, \dots$$

试编写求 k 阶斐波那契序列的第 m 项值的函数算法, k 和 m 均以值调用的形式在函数参数表中出现。

【分析】在编写此题目的函数算法的过程中, 首先应根据参量 m 和 k 区分下列4种情况: (1) $m < 0$; (2) $0 \leq m < k-1$; (3) $m = k-1$; (4) $m \geq k$ 。其次在计算 $m \geq k$ 的 f_m 值时, 可先对计算公式作数学处理, 将 f_{i+1} 表示为 f_i 和 f_{i-k} 的简单函数; 最后考虑计算 f_n 所需的辅助空间。

【解答】算法如下:

```
Status fib(int k, int m, int &f)//求k阶斐波那契序列的第m项的值f
{
    int tempd;
    if(k<2 || m<0) return ERROR;
    if(m<k-1) f=0;
    else if (m==k-1) f=1;
    else
    {
        for(i=0;i<=k-2;i++) temp[i]=0;
        temp[k-1]=1; //初始化
        for(i=k;i<=m;i++) //求出序列第k至第m个元素的值
        {
            sum=0;
            for(j=i-k;j<i;j++) sum+=temp[j];
            temp[i]=sum;
        }
        f=temp[m];
    }
    return OK;
} //fib
```

11. 假设有A, B, C, D, E 5个高等院校进行田径对抗赛, 各院校的单项成绩均已存入计算机, 并构成一张表, 表中每一行的形式为:

项目名称	性别	校名	成绩	得分
------	----	----	----	----

试编写算法, 处理上述表格, 以统计各院校的男、女总分和团体总分, 并输出。

【分析】设置此题目的目的在于复习结构型变量的使用方法, 在此题中可设

```
typedef struct{
    char *sport;
    enum{male, female} gender;
    char schoolname; //校名为'A', 'B', 'C', 'D'或'E'
    char *result;
    int score;
} resulttype;
typedef struct{
    int malescore;
```

```

    int femalescore;
    int totalscore;
} scoretype;

```

【解答】算法如下:

```

void summary(resulttype result[ ])//求各校的男女总分和团体总分, 假设结果已经储存在result[ ]数组中
{
    scoretype score ;
    i=0;
    while(result[i].sport!=NULL)
    {
        switch(result[i].schoolname)
        {
            case 'A':
                score[0].totalscore+=result[i].score;
                if(result[i].gender==0) score[0].malescore+=result[i].score;
                else score[0].femalescore+=result[i].score;
                break;
            case 'B':
                score [1].totalscore+=result[i].score;
                if(result[i].gender==0) score[1] .malescore+=result[i].score;
                else score[1] .femalescore+=result[i].score;
                break;
            .....
        }
        i++;
    }
    for(i=0;i<5;i++)
    {
        printf("School %d:\n", i);
        printf("Total score of male:%d\n", score[i].malescore);
        printf("Total score of female:%d\n", score[i].femalescore);
        printf("Total score of all:%d\n\n", score[i].totalscore);
    }
} //summary

```

12. 试编写算法, 计算 $i! \cdot 2^i (i=0, 1, \dots, n-1)$ 的值并分别存入数组 $a[\text{arrsize}]$ 的各个分量中。假设计算机中允许的整数最大值为MAXINT, 则当 $n > \text{arrsize}$ 或对某个 $k (0 \leq k \leq n-1)$ 使 $k! \cdot 2k > \text{MAXINT}$ 时, 应按出错处理。注意选择你认为较好的出错处理方法。

【分析】注意MAXINT为计算机中允许出现的最大值, 则在过程体中不能以计算所得结果大于MAXINT作为判断出错的依据。当某一项的结果超过了MAXINT时, 它除以前面一项的商会发生异常。

【解答】算法如下:

```

Status algo119(int a[arrsize])//求 $i! \cdot 2^i$ 序列的值且不超过MAXINT
{
    last=1;
    for(i=1;i<=arrsize;i++)

```

```

{
    a[i-1]=last*2*i;
    if((a[i-1]/last)!= (2*i)) return OVERFLOW;
    last=a[i-1];
    last=a[i-1];
    return OK;
}
} //algo119

```

13. 试编写算法求一元多项式 $P_n(x) = \sum_{i=0}^n a_i x^i$ 的值 $P_n(x_0)$, 并确定算法中每一语句的执行次数和整个算法的时间复杂度。注意选择你认为较好的输入和输出方法。本题的输入为 $a_i (i=0, 1, \dots, n)$ 、 x_0 和 n , 输出为 $P_n(x_0)$ 。

【分析】注意计算过程中, 不要对多项式中的每一项独立计算 x 的幂。

【解答】算法如下:

```

void polyvalue()
{
    float ad;
    float *p=a;
    printf("Input number of terms:");
    scanf("%d", &n);
    printf("Input the %d coefficients from a0 to a%d:\n", n, n);
    for(i=0;i<=n;i++) scanf("%f", p++);
    printf("Input value of x:");
    scanf("%f", &x);
    p=a;xp=1;sum=0; //xp用于存放x的i次方
    for(i=0;i<=n;i++)
    {
        sum+=xp*(p++);
        xp*=x;
    }
    printf("Value is:%f", sum);
} //polyvalue

```

1.4 考研真题分析

例题1-1 (中国科学院软件研究所1999年试题)

判断正误:

- ① 顺序存储方式只能用于存储线性结构。
- ② 顺序查找法适用于存储结构为顺序或链接存储的线性表。

【解答】① 错误。顺序存储方式也可以用来存储树型结构, 例如堆排序中的堆。

- ② 正确。

例題1-2 (中国科学院计算机网络信息中心2001年试题)

单项选择题:

计算机算法必须具备输入、输出、_____等5个特性。

- A. 可行性、可移植性和可扩展性 B. 可行性、确定性和有穷性
C. 确定性、有穷性和稳定性 D. 易读性、安全性和稳定性

【解答】B

例題1-3 (中国科学院计算机网络信息中心2001年试题)

单项选择题:

设三个函数 f , g , h 分别为

$$f(n)=100n^3+n^2+1000$$

$$g(n)=25n^3+4000n^2$$

$$h(n)=n^{1.01}+1000n\log_2 n$$

以下关系式中有错的是_____。

- A. $f(n)=O(g(n))$ B. $g(n)=O(f(n))$
C. $h(n)=O(n^{1.01})$ D. $h(n)=O(n\log_2 n)$

【解答】D

例題1-4 (中国科学院计算机网络信息中心2001年试题)

问答题:

已知在串匹配的KMP算法中求Next函数的算法为

```
void get_next(char *t, int *next){
    for(j=1, k=0, next[1]=0; j<strlen(t);)
        if((k==0) || (t[j]==t[k])){
            j++, k++;
            next[j]=k;
        }
        else k=next[k];
} // end get_next
```

试给出该函数的时间复杂度并证明该结论。

【解答】设 $\text{strlen}(t)=m$, 则时间复杂度为 $O(m)$ 。

【证明】考虑算法中 $j-k$ 的值, 在算法执行过程中 $j-k$ 值只增不减, 而 $j-k$ 增加的次数小于等于 j 的最大值 m , $j-k$ 得数不变的次数小于等于 j 增加的次数 m , 而算法中要么 j 增加, 要么 $j-k$ 增加, 故总执行次数小于等于 $2m$, 因此, 算法的时间复杂度为 $O(m)$ 。

例題1-5 (中国科学院计算机网络信息中心2001年试题)

算法题:

设 L 是长度为 n 的单链表的头指针, 不带头结点, 每个表结点中存放一个字符。请写一算法判断 L 中存放的字符串是否是中心对称的, 要求附加的空间是 $O(1)$, 时间复杂度为 $O(n)$ 。

【分析】① 先计算出 n 值;

② 再将前 $n/2$ 个结点作为一个链表, 后 $n/2$ 个结点也作为一个链表并逆置;

③ 然后依次判断两个单链表对应元素是否相同。

应注意边量条件处理, 即 n 为奇数的情况。

【解答】算法如下:

```
PROC checkstr(var L:linkisttp; n:integer);
```

```
VAR p, h, q:linkisttp;
```

```
    i:integer;
```

```
BEGIN
```

```
    P:=L;
```

```
    if (n mod 2 = 0) then
```

```
        for i:=1 to (n/2-1) do
```

```
            p:=p ↑ .next;
```

```
        else
```

```
            for i:=1 to n-1/2 do
```

```
                p:=p ↑ .next;
```

```
            h:=p ↑ .next;
```

```
            p ↑ .next:=nil;
```

```
            p:=h ↑ .next;
```

```
            h ↑ .next:=nil;
```

```
            q:=p;
```

```
            while q <> nil do
```

```
                begin
```

```
                    q:=p ↑ .next;
```

```
                    p ↑ .next:=h;
```

```
                    h:=p;
```

```
                    p:=q;
```

```
                end
```

```
            p:=L;
```

```
            q:=h;
```

```
            while q <> nil do
```

```
                if (p ↑ .data <> q ↑ .data) then
```

```
                    return (false)
```

```
                else begin
```

```
                    p:=p ↑ .next;
```

```
                    q:=q ↑ .next;
```

```
                end
```

```
            return (true);
```

```
ENDP
```

例题1-6 (中国科学院计算机网络信息中心2000年试题)

下述函数中渐近时间最小的是_____。

A. $T_1(n) = n \log_2 n + 1000 \log_2 n$

B. $T_2(n) = n^{\log_2 3} - 1000 \log_2 n$

C. $T_3(n) = n^2 - 1000 \log_2 n$

D. $T_4(n) = 2n \log_2 n - 1000 \log_2 n$