

高等学校教材·计算机科学与技术

可赠送课件素材

jsjc@tup.tsinghua.edu.cn

数据结构(C语言描述)

学习指导与习题解答

徐孝凯 贺桂英 编著



清华大学出版社

高等学校教材·计算机科学与技术

数据结构（C 语言描述） 学习指导与习题解答

徐孝凯 贺桂英 编著

清华大学出版社
北 京

内 容 简 介

本书是与清华大学出版社出版的、徐孝凯和贺桂英编著的《数据结构(C语言描述)》一书相配套的辅助教材,它给出了主教材中每一章的知识要点与绝大部分习题的参考解答,并在附录A中给出了每章的单选题和填空题,附录B中给出了相应的参考解答。本书既可与《数据结构(C语言描述)》一书配套使用,也可与其他用C语言描述的数据结构教材配套使用,还可作为考研的参考用书。

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(C语言描述)学习指导与习题解答/徐孝凯,贺桂英编著. —北京:清华大学出版社,2004.10
(高等学校教材·计算机科学与技术)

ISBN 7-302-09359-8

I. 数… II. ①徐… ②贺… III. ①数据结构—高等学校—教学参考资料 ②C语言—程序设计—高等学校—教学参考资料 IV. ①TP311.12②TP312

中国版本图书馆CIP数据核字(2004)第088321号

出 版 者:清华大学出版社

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

社总机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

责任编辑:郑寅堃

印 刷 者:清华园胶印厂

装 订 者:北京鑫海金澳胶印有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:7.5 字数:178千字

版 次:2004年10月第1版 2004年10月第1次印刷

书 号:ISBN 7-302-09359-8/TP·6538

印 数:1~3000

定 价:12.00元

本书如存在文字不清、漏印以及缺页、倒页、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:(010) 62770175-3103 或 (010) 62795704

清华大学出版社计算机教材

编审委员会成员

(按地区排序)

清华大学	周立柱 教授	北京大学	杨冬青 教授
	章 征 教授		陈 钟 教授
	王建民 教授		陈立军 副教授
	刘 强 副教授	中国人民大学	王 珊 教授
	冯建华 副教授		孟小峰 教授
北京航空航天大学	马殿富 教授		陈 红 教授
	吴超英 副教授	北京信息工程学院	孟庆昌 教授
	姚淑珍 教授	石油大学(北京)	陈 明 教授
北京交通大学	阮秋琦 教授	天津大学	艾德才 教授
北京科技大学	杨炳儒 教授		
南开大学	吴功宜 教授		
复旦大学	吴立德 教授	上海交通大学	傅育熙 教授
	吴百锋 教授		蒋建伟 副教授
	杨卫东 副教授	华东师范大学	杨宗源 教授
华东理工大学	邵志清 教授		应吉康 教授
东华大学	乐嘉锦 教授	上海第二工业大学	蒋川群 教授
浙江大学	吴朝晖 教授	南京大学	骆 斌 教授
	李善平 教授	南京理工大学	张功萱 教授
南京航空航天大学	秦小麟 教授	苏州大学	龚声蓉 教授
南京邮电学院	朱秀昌 教授		
江苏大学	宋余庆 教授		
武汉大学	何炎祥 教授	华中科技大学	刘乐善 教授
中南财经政法大学	刘腾红 教授		朱定华 教授
武汉理工大学	李中年 教授	华中师范大学	魏开平 教授
			王林平 副教授

国防科技大学	赵克佳 教授	中南大学	陈松乔 教授
	肖 侬 副教授		
湖南大学	林亚平 教授		
	邹北骥 教授		
西安交通大学	沈钧毅 教授	西北大学	周明全 教授
	齐 勇 教授	西安石油学院	方 明 教授
长安大学	巨永峰 教授		
西安邮电学院	陈莉君 副教授		
哈尔滨工业大学	郭茂祖 教授	吉林大学	何 桥 教授
长春工程学院	沙胜贤 教授		徐一平 教授
			毕 强 教授
山东大学	孟祥旭 教授	山东科技大学	郑永果 教授
	郝兴伟 教授		
中山大学	潘小轰 教授	厦门大学	冯少荣 副教授
福州大学	林世平 副教授		
云南大学	刘惟一 教授	重庆邮电学院	王国胤 教授
西南交通大学	杨 燕 副教授		

出版说明

改革开放以来,特别是党的十五大以来,我国教育事业取得了举世瞩目的辉煌成就,高等教育实现了历史性的跨越,已由精英教育阶段进入国际公认的大众化教育阶段。在质量不断提高的基础上,高等教育规模取得如此快速的发展,创造了世界教育发展史上的奇迹。当前,教育工作既面临着千载难逢的良好机遇,同时也面临着前所未有的严峻挑战。社会不断增长的高等教育需求同教育供给特别是优质教育供给不足的矛盾,是现阶段教育发展面临的基本矛盾。

教育部一直十分重视高等教育质量工作。2001年8月,教育部下发了《关于加强高等学校本科教学工作,提高教学质量的若干意见》,提出了十二条加强本科教学工作提高教学质量的措施和意见。2003年6月和2004年2月,教育部分别下发了《关于启动高等学校教学质量与教学改革工程精品课程建设工作的通知》和《教育部实施精品课程建设提高高校教学质量和人才培养质量》文件,指出“高等学校教学质量和教学改革工程”,是教育部正在制订的《2003—2007年教育振兴行动计划》的重要组成部分,精品课程建设是“质量工程”的重要内容之一,教育部计划用五年时间(2003—2007年)建设1500门国家级精品课程,利用现代化的教育信息技术手段将精品课程的相关内容上网并免费开放,以实现优质教学资源共享,提高高等学校教学质量和人才培养质量。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合新世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。首批推出的特色精品教材包括以下三个系列:

(1) 高等学校教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算

机应用类教材。

(2) 高等学校教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 高等学校教材·电子信息——高等学校电子信息相关专业的教材。

清华大学出版社经过近二十年的努力，在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌，为我国的高等教育事业做出了重要贡献。清华版教材经过二十多年的精雕细刻，形成了技术准确、内容严谨的独特风格，这种风格将延续并反映在特色精品教材的建设中。

总策划 李家强

策 划 卢先和 丁 岭

清华大学出版社教材编审委员会

E-mail: dingl@tup.tsinghua.edu.cn

luxh@tup.tsinghua.edu.cn

前 言

本书是与清华大学出版社出版的、徐孝凯和贺桂英编著的《数据结构(C语言描述)》一书相配套的辅助教材,它给出了主教材中每一章的知识要点与绝大部分习题的参考解答,并在附录A中给出了每章的单选题和填空题,附录B中给出了相应的参考解答。希望同学们按照每个题目的要求,独立地分析问题和解决问题,给出相应的解答,不要事先翻阅答案。书中所给的参考解答只有当你解题后或确实不会做时进行参考才是有益的。

数据结构是一门实践性很强的课程。对于自己编写的每一个算法,不仅要尽量符合算法评价的各项指标,更重要的是上机验证,在反复调试的过程中,通过典型的数据输入使得算法中的每条语句都被执行通过,或者说不存在没有被执行通过的语句或语句块。若调试过程发现语法或逻辑错误,则要及时修改。所谓逻辑错误是指算法设计上隐含的错误,虽然算法能够被正确地编辑和连接,但运行后得不到正确的结果。通过上机运行程序能够加深对所学知识的理解和掌握,进而获得书本上学不到的知识。

解决一个算法问题通常要经过以下几步:①根据题目要求分析出设计思路或建立起数学模型;②根据设计思路或数学模型画出相应的流程图;③根据流程图用一种计算机语言(如C语言)编写出详细算法;④编写出能够调用该算法的完整程序;⑤上机调试和运行该程序。若发现错误则回到上述某一步开始向下修改,通过反复调试和修改,直到获得满意的结果为止。当然对于一些简单问题,上述步骤有的可以省略,有的则可以合并。

对于要解决的同一个问题,由于所采用的数据结构可能不同,所选择的计算方法(即算法)可能不同,则编写出的程序就可能不同,但只要你的程序正确并且有效(即具有较好的时间和空间复杂度)即可。例如要对 n 个数据进行排序,既可选择数组结构,也可选择二叉搜索树结构;对于数组结构,可以从快速、堆、合并、直接插入、直接选择等多种排序运算方法中任选一种。当采用不同的数据结构和排序方法时编写出的排序算法也不会相同,但它们都是正确的。通过进行时间和空间复杂度的分析可以比较出哪一种或几种更为有效。因此,每个人按照习题编写出的算法程序不要求与本书所给的解答完全一致,也许你编写出的算法具有更好的性能。

书中所有算法和程序都在C语言或VC++6.0语言环境下调试通过,但由于编写时间仓促,错误和不足之处在所难免,敬请广大读者和专家批评指正。我们的电子邮箱地址为xuxk@crtvu.edu.cn,联系电话为010-64910302。

徐孝凯 贺桂英
2004年3月

目 录

第 1 章 绪论	1
内容提要	1
习题一	2
参考解答	4
第 2 章 线性表	8
内容提要	8
习题二	9
参考解答	11
第 3 章 稀疏矩阵和广义表	21
内容提要	21
习题三	23
参考解答	23
第 4 章 栈和队列	29
内容提要	29
习题四	30
参考解答	33
第 5 章 树和二叉树	45
内容提要	45
习题五	46
参考解答	50
第 6 章 二叉树的应用	56
内容提要	56
习题六	57
参考解答	57
第 7 章 图	65
内容提要	65
习题七	66
参考解答	67
第 8 章 查找	74
内容提要	74

习题八	75
参考解答	75
第9章 排序	80
内容提要	80
习题九	81
参考解答	81
附录 A 补充题	87
附录 B 补充题参考解答	104

第1章 绪 论

内容提要

1. 数据由数据元素按照一定的逻辑结构组织而成, 而每个数据元素通常又由一个或多个数据项顺序排列而成。同一数据元素的各数据项之间无任何次序关系, 则说它们之间为集合结构。

2. 数据的逻辑结构可概括为四种类型: 集合结构、线性结构、树形结构和图状结构。由它们的组合可以构成任何更为复杂的逻辑结构。

3. 在数据的集合结构中, 不考虑数据之间的任何次序, 它们处于无序的、各自独立的状态; 在线性结构中, 数据之间是 1 对 1 的关系; 在树形结构中, 数据之间是 1 对多的关系; 在图状结构中, 数据之间是多对多的关系。

4. 数据的存储结构可概括为四种类型: 顺序结构、链接结构、索引结构和散列结构, 由它们的组合可以构成任何更为复杂的存储结构。

5. 一个数组占有一块连续的存储空间, 每个元素的物理存储单元是按下标位置从 0 开始连续编号的, 相邻元素其存储位置也相邻。对于任一种数据的逻辑结构, 若能够把元素之间的逻辑关系对应地转换为数组下标位置之间的物理关系, 则就能够利用数组来实现其顺序存储结构。

6. 数据的逻辑结构需要采用图形来表示, 数据的存储结构既需要采用图形来表示, 又需要采用一种计算机语言的数据类型来定义。

7. 数据的一种逻辑结构可以通过不同的存储结构来实现, 反过来, 一种存储结构可以用来实现不同的逻辑结构。具体采用何种存储结构来存储数据 (包括存储所有数据元素及相互之间的逻辑结构), 由对数据的运算要求而定。

8. 实现一个运算的方法和步骤称为此运算的算法。描述一个算法有许多形式, 如采用流程图、自然语言、计算机程序设计语言等。在数据结构课程中主要使用某一种计算机程序设计语言来具体描述。本书采用通用性强的 C 语言来描述。

9. 一个算法的性能主要从四个方面来评价: 正确性、可读性、健壮性、有效性 (即程序运行时的时间复杂度和空间复杂度)。

10. 一个算法的时间或空间复杂度可分为最好、最差 (坏) 和平均这三种情况。通常只关心最差或平均的情况, 并且以讨论时间复杂度为主, 因为节省时间比空间更重要。

11. 算法的时间或空间复杂度的数量级采用大写的 O 表示, 通常有常量级、对数级、线性级、线性与对数乘积级、平方级、立方级、指数级等级别, 对应量级表示依次为 $O(1)$, $O(\log_2 n)$, $O(n)$, $O(n \log_2 n)$, $O(n^2)$, $O(n^3)$, $O(2^n)$ 等。当 n 较大时, 量级越靠前的算法, 其运行时间越短或占用空间越少, 也就是说算法的时间或空间复杂度越好。

习题一

1.1 有下列几种用二元组表示的数据结构, 试画出它们分别对应的图形表示 (当出现多个关系时, 对每个关系画出相应的结构图), 并指出它们分别属于何种结构。

1. $A=(K,R)$ 其中

$$K=\{a_1, a_2, a_3, \dots, a_n\}$$

$$R=\{\}$$

2. $B=(K,R)$, 其中

$$K=\{a, b, c, d, e, f, g, h\}$$

$$R=\{\langle a, b \rangle, \langle b, c \rangle, \langle c, d \rangle, \langle d, e \rangle, \langle e, f \rangle, \langle f, g \rangle, \langle g, h \rangle\}$$

3. $C=(K,R)$, 其中

$$K=\{a, b, c, d, e, f, g, h\}$$

$$R=\{\langle d, b \rangle, \langle d, g \rangle, \langle b, a \rangle, \langle b, c \rangle, \langle g, e \rangle, \langle g, h \rangle, \langle e, f \rangle\}$$

4. $D=(K,R)$, 其中

$$K=\{1, 2, 3, 4, 5, 6\}$$

$$R=\{(1, 2), (2, 3), (2, 4), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6)\}$$

5. $E=(K,R)$, 其中

$$K=\{48, 25, 64, 57, 82, 36, 75, 43\}$$

$$R=\{r_1, r_2, r_3\}$$

$$r_1=\{\langle 48, 25 \rangle, \langle 25, 64 \rangle, \langle 64, 57 \rangle, \langle 57, 82 \rangle, \langle 82, 36 \rangle, \langle 36, 75 \rangle, \langle 75, 43 \rangle\}$$

$$r_2=\{\langle 48, 25 \rangle, \langle 48, 64 \rangle, \langle 64, 57 \rangle, \langle 64, 82 \rangle, \langle 25, 36 \rangle, \langle 82, 75 \rangle, \langle 36, 43 \rangle\}$$

$$r_3=\{\langle 25, 36 \rangle, \langle 36, 43 \rangle, \langle 43, 48 \rangle, \langle 48, 57 \rangle, \langle 57, 64 \rangle, \langle 64, 75 \rangle, \langle 75, 82 \rangle\}$$

1.2 用 C 语言函数编写下列每一个算法, 并分别求出它们的时间复杂度。

1. 比较同一简单类型的两个数据 x_1 和 x_2 的大小, 对于 $x_1 > x_2$, $x_1 = x_2$ 和 $x_1 < x_2$ 这三种不同情况应分别返回 '>', '=' 和 '<' 字符。假定简单类型用 SimpleType 表示, 它可通过 typedef 语句定义为任一简单类型。

2. 将一个字符串中的所有字符按相反的次序重新放置。

3. 求一维 double 型数组 $a[n]$ 中的所有元素之乘积。

4. 计算 $\sum_{i=0}^n \frac{x^i}{i+1}$ 的值。

5. 假定一维整型数组 $a[n]$ 中的每个元素值均在 $[0, 200]$ 区间内, 分别统计出落在 $[0, 20]$, $[20, 50]$, $[50, 80]$, $[80, 130]$, $[130, 200]$ 等各区间内的元素个数。

6. 从二维整型数组 $a[m][n]$ 中查找出最大元素所在的行、列下标。

1.3 指出下列各算法的功能并求出其时间复杂度。

1. `int Prime(int n)`

```
{
    int i=1;
    int x=(int)sqrt(n);
    while(++i<=x)
        if(n%i==0) break;
    if(i>x) return 1;
    else return 0;
}

2. int sum1(int n)
{
    int i, p=1, s=0;
    for(i=1; i<=n; i++){
        p*=i;
        s+=p;
    }
    return s;
}

3. int sum2(int n)
{
    int i,j,s=0;
    for(i=1; i<=n; i++){
        int p=1;
        for(j=1; j<=i; j++)
            p*=j;
        s+=p;
    }
    return s;
}

4. int fun(int n)
{
    int i=1, s=1;
    while(s<n)
        s+=++i;
    return i;
}

5. void mtable(int n)
{
    int i,j;
    for(i=1; i<=n; i++){
        for(j=i; j<=n; j++)
            printf("%d*%d=%d\n",i,j,i*j);
        printf("\n");
    }
```

```

    }
}

6. void cmatrix(int a[M][N], int d)
    /*M 和 N 为全局整型常量*/
{
    int i, j;
    for(i=0; i<M; i++)
        for(j=0; j<N; j++)
            a[i][j]*=d;
}

7. void matmult(int a[M][N], int b[N][L], int c[M][L])
    /*M, N 和 L 均为全局整型常量*/
{
    int i, j, k;
    for(i=0; i<M; i++)
        for(j=0; j<L; j++)
            c[i][j]=0;
    for(i=0; i<M; i++)
        for(j=0; j<L; j++)
            for(k=0; k<N; k++)
                c[i][j]+=a[i][k]*b[k][j];
}

```

参考解答

1.1 解答 (略)。

1.2 用 C 语言函数编写下列每一个算法, 并分别求出它们的时间复杂度。
参考解答包含在下面程序中。

```

#include<stdio.h>
#include<string.h>
#define M 10
#define N 10
typedef int SimpleType;
char Compare(SimpleType x1, SimpleType x2)
{
    if(x1>x2) return '>';
    else if(x1==x2) return '=';
    else return '<';
}
/*时间复杂度为 O(1)*/

```

```
void Reverse(char* p)
{
    int i;
    int n=strlen(p);
    for(i=0; i<n/2; i++) {
        char ch;
        ch=p[i];
        p[i]=p[n-i-1];
        p[n-i-1]=ch;
    }
}
/*时间复杂度为 O(n), n 表示字符串的长度*/

double Product(double a[], int n)
{
    int i;
    double p=1;
    for(i=0; i<n; i++)
        p*=a[i];
    return p;
}
/*时间复杂度为 O(n)*/

double Accumulate(double x, int n)
{
    double p=1, s=1, i;
    for(i=1; i<=n; i++){
        p*=x;
        s+=p/(i+1);
    }
    return s;
}
/*时间复杂度为 O(n)*/

int Count(int a[], int n, int c[5])
    /*用数组 c[5] 保存统计结果*/
{
    int d[5]={20,50,80,130,201};    /*用来保存各统计区间的上限*/
    int i,j;
    for(i=0; i<5; i++) c[i]=0;      /*给数组 c[5] 中的每个元素赋初值 0*/
    for(i=0; i<n; i++)
    {
        if(a[i]<0 || a[i]>200)
            return 0;    /*返回数值 0 表示数组中数据有错, 统计失败*/
    }
}
```

```

        for(j=0; j<5; j++) /*查找 a[i] 所在的区间*/
            if(a[i]<d[j]) break;
        c[j]++; /*使统计相应区间的元素增 1*/
    }
    return 1; /*返回数值 1 表示统计成功*/
}
/*时间复杂度为 O(n)*/

void find(int a[M][N], int m, int n, int* Lin, int* Col)
    /*M 和 N 为全局常量, 应满足 M>=n 和 N>=n 的条件, Lin 和 Col 为指针形参, 用其所
    指向的实参带回行、列值*/
{
    int i, j;
    *Lin=0; *Col=0;
    for(i=0; i<m; i++)
        for(j=0; j<n; j++)
            if(a[i][j]>a[*Lin][*Col]) {*Lin=i; *Col=j;}
}
/*时间复杂度为 O(m*n)*/

void main()
{
    int x=20, y=30;
    char p[12]="123456789";
    double a[5]={1,2,3,4,5};
    int b[8]={23,10,56,125,38,44,130,98};
    int c[3][N]={ {1,3,5,7}, {10,8,6,4}, {4,12,8,3} };
    int d[5];
    int i, c1, c2;
    printf("%c\n", Compare(x, y));
    Reverse(p);
    printf("%s\n", p);
    printf("%f\n", Product(a, 5));
    printf("%f\n", Accumulate(1.5, 3));
    Count(b, 8, d);
    for(i=0; i<5; i++) printf("%4d", d[i]); printf("\n");
    find(c, 3, 4, &c1, &c2);
    printf("%4d%4d%4d\n", c1, c2, c[c1][c2]);
}

```

1.3 指出下列各算法的功能并求出其时间复杂度。

参考解答如下。

1. 判断 n 是否是一个素数, 若是则返回数值 1, 否则返回 0。该算法的时间复杂度为 $O(\sqrt{n})$ 。

2. 计算 $\sum_{i=1}^n i!$ 的值。时间复杂度为 $O(n)$ 。

3. 计算 $\sum_{i=1}^n i!$ 的值。时间复杂度为 $O(n^2)$ 。

4. 求出满足不等式 $1+2+3+\cdots+i \geq n$ 的最小 i 值。时间复杂度为 $O(\sqrt{n})$ 。

提示：因为 $1+2+3+\cdots+i=(1+i)i/2$ ，即当 n 很大时 i 的平方与 n 成正比，所以 i 的值（即函数中 while 循环的次数）与 n 的平方根成正比。

5. 打印出一个具有 n 行的乘法表，第 i 行 ($1 \leq i \leq n$) 中有 $n-i+1$ 个乘法项，每个乘法项为 i 与 j ($i \leq j \leq n$) 的乘积。时间复杂度为 $O(n^2)$ 。

6. 使数组 $a[M][N]$ 中的每一个元素均乘以 d 的值。时间复杂度为 $O(M \times N)$ 。

7. 矩阵相乘，即 $a[M][N] \times b[N][L] \rightarrow c[M][L]$ 。时间复杂度为 $O(M \times N \times L)$ 。