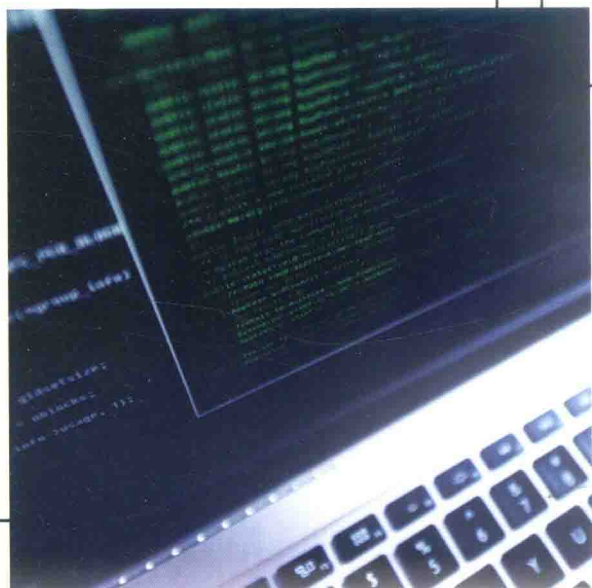




高等院校计算机类规划教材
全国高等院校计算机基础教育研究会立项项目成果



数据结构分析与解答 (C语言版)

牛为华 袁和金 刘 军 编著
王 妤 王翠茹



北京邮电大学出版社
www.buptpress.com



高等院校计算机类规划教材
全国高等院校计算机基础教育研究会立项项目成果

数据结构分析与解答 (C语言版)

牛为华 袁和金 刘 军 编著
王 好 王翠茹



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书与北京邮电大学出版社2019年8月出版的《数据结构(C语言版)》一书配套使用。本书不仅对配套教材的课后习题进行了详细的分析和解答,还对数据结构各章的知识结构、知识要点和难点进行了分析和总结。本书不仅可以作为教材的配套辅导资料,还可单独使用。书中精选了一些综合性较强、难度较大的题目作为例题进行讲解,可以满足学生的自学需求,有助于培养他们综合运用所学知识解决实际问题的能力。本书附录中包含近几年计算机专业全国硕士研究生招生考试数据结构部分的真题及模拟试卷,帮助学生了解考研的题型、难度和知识要点,提高他们的实际应用能力。本书既可作为各大中专院校“数据结构”课程的复习资料,又可作为相关专业自学考试与硕士入学考试“数据结构”科目的复习资料。

图书在版编目(CIP)数据

数据结构分析与解答: C语言版 / 牛为华等编著. -- 北京: 北京邮电大学出版社, 2022. 3
ISBN 978-7-5635-6581-8

I. ①数… II. ①牛… III. ①数据结构—教学参考资料②C语言—程序设计—教学参考资料
IV. ①TP311.12②TP312.8

中国版本图书馆CIP数据核字(2021)第255941号

策划编辑: 马晓芹 责任编辑: 王晓丹 左佳灵 封面设计: 七星博纳

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路10号

邮政编码: 100876

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 保定市中画美凯印刷有限公司

开 本: 787 mm×1 092 mm 1/16

印 张: 18.75

字 数: 441千字

版 次: 2022年3月第1版

印 次: 2022年3月第1次印刷

ISBN 978-7-5635-6581-8

定价: 48.00元

· 如有印装质量问题, 请与北京邮电大学出版社发行部联系 ·

前 言

数据结构的原理和算法比较抽象,对于只有计算机程序设计基础知识的本科低年级学生来说,深入理解和熟练掌握其中的原理和算法是比较困难的,需要通过大量的习题来复习和巩固所学的知识,为此,我们编写了本书。本书遵循全国高等院校计算机相关专业本科“数据结构”课程教学大纲的要求,简要分析了各章的知识结构、难点,并归纳和总结了各章的知识要点;精选了一些难度稍大、综合性较强的题目作为例题进行详细的分析和解答;给出了与教材配套的习题及其分析和解答,这些习题内容丰富,题型多样,较好地覆盖了数据结构的相关知识;提供了近几年计算机专业全国硕士研究生招生考试中一些数据结构相关的试题及其解析,并汇编了几套模拟试题,便于学生自学和测验。

全书共分绪论、顺序表、链表、数组和广义表、字符串、树、图、查找表、内排序等内容。本书可以作为《数据结构(C语言版)》一书的配套辅导资料,适合作为计算机科学与技术、软件工程、网络工程,以及电子、信息相关专业数据结构知识的教辅资料,也可供从事相关工作的科技与工程人员参考。

本书由牛为华组织编写并统稿。其中,第1~4章由牛为华编写,第5~7章由袁和金编写,第8~9章由刘军编写,附录部分由王好编写,王翠茹教授认真审阅了全书并提出了许多宝贵意见。

由于编者水平有限,本书难免存在不妥之处,恳请读者批评指正。

目 录

第 1 章 绪论	1
1.1 知识要点	1
1.1.1 相关术语	1
1.1.2 算法和算法分析	3
1.2 典型例题分析	3
教材习题 1	6
习题 1 答案及解析	8
第 2 章 顺序表	12
2.1 知识要点	12
2.1.1 线性表	12
2.1.2 栈	15
2.1.3 队列	16
2.2 典型例题分析	17
教材习题 2	25
习题 2 答案及解析	28
第 3 章 链表	39
3.1 知识要点	39
3.1.1 单链表	39
3.1.2 链栈和链队	43
3.1.3 循环链表和多重链表	45
3.2 典型例题分析	46
教材习题 3	52
习题 3 答案及解析	55

第 4 章 数组和广义表	71
4.1 知识要点	71
4.1.1 数组	71
4.1.2 广义表	74
4.2 典型例题分析	74
教材习题 4	82
习题 4 答案及解析	83
第 5 章 字符串	100
5.1 知识要点	100
5.1.1 字符串	100
5.1.2 字符串的存储表示	100
5.1.3 字符串的基本运算	101
5.2 典型例题分析	105
教材习题 5	113
习题 5 答案及解析	114
第 6 章 树	127
6.1 知识要点	128
6.1.1 树的基本知识	128
6.1.2 二叉树	128
6.1.3 线索二叉树	135
6.1.4 哈夫曼树及哈夫曼编码	137
6.2 典型例题分析	138
教材习题 6	142
习题 6 答案及解析	144
第 7 章 图	157
7.1 知识要点	158
7.1.1 图的定义、术语及性质	158
7.1.2 图的存储结构	159
7.1.3 图的遍历	161
7.1.4 图的应用	165

7.2 典型例题分析	167
教材习题 7	173
习题 7 答案及解析	174
第 8 章 查找表	184
8.1 知识要点	185
8.1.1 查找表的基本概念	185
8.1.2 静态查找表	185
8.1.3 树形查找表	188
8.1.4 散列表查找	192
8.2 典型例题分析	193
教材习题 8	200
习题 8 答案及解析	201
第 9 章 内排序	209
9.1 知识要点	210
9.1.1 排序的基本概念	210
9.1.2 插入类排序	210
9.1.3 交换类排序	213
9.1.4 选择类排序	215
9.1.5 归并排序和基数排序	217
9.1.6 总结	221
9.2 典型例题分析	221
教材习题 9	225
习题 9 答案及解析	226
参考文献	233
附录 1 计算机专业全国硕士研究生招生考试数据结构部分真题解析	234
2020 年真题解析	234
2019 年真题解析	241
2018 年真题解析	249
2017 年真题解析	255
2016 年真题解析	260

附录 2 模拟试题及参考答案	268
模拟试题 1	268
模拟试题 2	270
模拟试题 3	272
模拟试题 4	275
模拟试题 1 参考答案	278
模拟试题 2 参考答案	281
模拟试题 3 参考答案	283
模拟试题 4 参考答案	287

第 1 章 绪 论

“数据结构”是计算机学科的一门专业基础课程,它为后续的多门专业课程提供了必要的知识和技能准备。例如,程序设计语言及其编译技术会用到栈、散列表及语法树,操作系统会用到队列、存储管理表及目录树,数据库系统会用到线性表、多重链表以及索引树等基本数据结构及其相关算法。

本章首先介绍了数据结构的一些基本概念、数据的逻辑结构和存储结构,以及数据类型的定义,然后阐述了算法设计的要求和方法,最后讨论了分析算法时间复杂度的方法。本章的重点在于了解数据结构的各个名词,以及时间复杂度、空间复杂度的估算。其中,算法时间复杂度的分析是本章的难点所在。

知识结构图

本章的知识结构如图 1.1 所示。



图 1.1 绪论的知识结构

1.1 知识要点

1.1.1 相关术语

1. 数据、数据元素、数据项、数据对象、数据结构

数据:数据是对客观事物的符号表示,在计算机科学中是指所有能输入计算机并被计算机程序处理的符号的总称。

数据元素:数据元素是数据的基本单位,在计算机程序中通常作为一个整体进行考虑

和处理。一个数据元素可以由若干个数据项组成。

数据项:数据项是数据的不可分割的最小单位。

数据对象: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: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。数据结构包括三方面的内容,即数据的逻辑结构、数据的存储结构和数据的运算。

2. 数据的逻辑结构

逻辑结构是数据元素之间的逻辑关系。数据的逻辑结构可以看作是从具体问题上抽象出来的数学模型,通常我们把数据元素之间的关联方式(邻接关系)叫作数据元素间的逻辑关系。数据元素之间逻辑关系的整体称为逻辑结构。表示方法通常有四种:

(1) 集合:结构中的数据元素之间除了“同属于一个集合”的关系外,别无其他关系。

(2) 线性结构:结构中的数据元素之间存在一对一的关系。若结构是非空集,则有且仅有一个开始结点,有且仅有一个终端结点,并且除开始结点无直接前驱和终端结点无直接后继外,其他所有结点有且仅有一个直接前驱和一个直接后继。

(3) 树形结构:结构中的数据元素之间存在一对多的关系。若结构是非空集,则除第一个结点外,其他所有结点都只有一个直接前驱,除叶子结点外,其他所有结点可能有多个直接后继。

(4) 图状结构:结构中的数据元素之间存在多对多的关系。若结构是非空集,则所有结点都可能有多于一个直接前驱和多个直接后继。

3. 数据的存储结构

数据的存储结构是指数据元素及其关系在计算机存储器中的表示。元素间的关系在计算机内的表示方法通常有四种:

(1) 顺序存储:每个存储结点只含有一个数据元素,所有结点相继存放在一个连续的存储区里,用存储结点间的位置关系表示数据元素之间的逻辑关系。

(2) 链式存储:每个存储结点不仅含有一个数据元素,还包含一组指针,每个指针指向一个与本结点有逻辑关系的结点。

(3) 索引存储:在存储结点信息的同时,还建立索引表,索引表的每一项称为索引项,索引项的一般形式是“(关键字,地址)”。关键字是能唯一标识一个结点的那些数据项,地址表示该关键字所在结点的起始存储位置。

(4) 散列存储:根据结点的关键字,采用某种方法,直接计算出结点的存储地址。

4. 数据类型

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。通常包括以下三种类型:

(1) 原子类型:其值不可再分的数据类型。

(2) 结构类型:结构类型的值由若干成分按某种结构组成,而且它的成分可以是非结构的,也可以是结构的。

(3) 抽象数据类型:抽象数据类型和数据类型实质上是一个概念,是指一个数学模型以及定义在该模型上的一组操作。

1.1.2 算法和算法分析

1. 算法的定义和特性

算法是指解决某一特定类型问题的有限运算序列,其中每一条指令表示一个或多个操作。一个算法应该具有下列特性:

- (1) 有穷性:一个算法必须在有穷步之后结束,即必须在有限时间内完成。
- (2) 确定性:算法的每一步必须有确切的定义,无二义性,算法的执行对应着相同的输入仅有唯一的一条执行路径。
- (3) 可行性:算法中的每一步都可以通过已经实现的基本运算的有限次执行得以实现。
- (4) 输入:一个算法具有零个或多个输入,这些输入取自特定的数据对象集合。
- (5) 输出:一个算法具有一个或多个输出,这些输出同输入之间存在某种特定的关系。

2. 算法的设计要求

- (1) 正确:算法的执行结果应当满足预先规定的功能和性能要求。
- (2) 可读:一个算法应当思路清晰、层次分明、简单明了、易读易懂。
- (3) 健壮:当输入的不是合法数据时,应能进行适当处理,不至于引起严重后果。
- (4) 高效:有效地使用存储空间和具有较高的时间效率。

3. 算法的性能分析与度量

算法的时间复杂度是执行该算法所耗费的时间,通常是问题规模为 n 的函数 $T(n)$ 。在实际分析时,可将语句执行次数的多少作为算法的时间量度。通常采用大 O 记号法来表示算法的时间复杂度。一般情况下有:

$$O(1) < O(\log_2 n) < O(n) < O(n \log_2 n) < O(n^2) < O(n^3) < O(2^n)$$

算法的空间复杂度是指该算法所消耗的存储空间,它也是问题规模 n 的函数 $S(n)$ 。

1.2 典型例题分析

例 1 简述数据的逻辑结构和存储结构的区别和联系。

解答 数据的逻辑结构反映的是数据之间的固有关系,而数据的存储结构是数据在计算机中的存储表示。可以采用不同的存储结构来表示具有逻辑结构特性的数据,如线性表可以用顺序表表示,也可以用链表来表示。当然采用的存储结构不同,对数据的操作在灵活性、算法复杂度等方面的差别也较大。

例 2 下列是几种用二元组表示的数据结构,画出它们分别对应的逻辑图形,并指出它们分别属于何种结构。

(1) $A = (K, R)$, 其中:

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

$$R = \{r\}$$

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

(2) $B=(K, R)$, 其中:

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

$$R=\{r\}$$

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

(3) $C=(K, R)$, 其中:

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

$$R=\{r\}$$

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

这里的圆括号对表示两结点是双向的。

(4) $D=(K, R)$, 其中:

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

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

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

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

解答 (1) A 对应逻辑图形如图 1.2 所示, 它是一种线性结构。

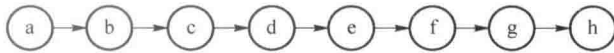


图 1.2 对应 A 的逻辑结构图示

(2) B 对应逻辑图形如图 1.3 所示, 它是一种树形结构。

(3) C 对应逻辑图形如图 1.4 所示, 它是一种图状结构。

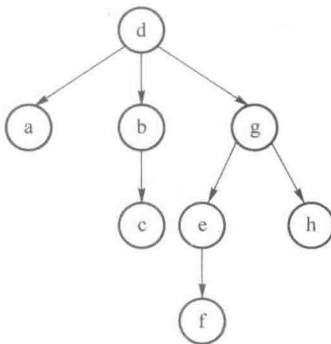


图 1.3 对应 B 的逻辑结构图示

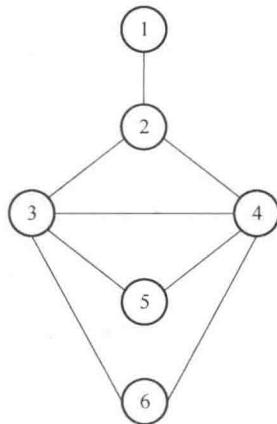


图 1.4 对应 C 的逻辑结构图示

(4) D 对应逻辑图形如图 1.5 所示, 它是一种图状结构, r_1 (对应图中虚线部分) 为线性结构, r_2 (对应图中实线部分) 则为树形结构。

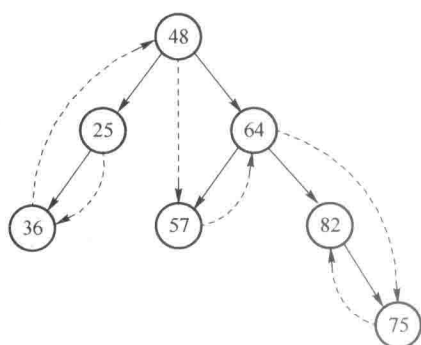


图 1.5 对应 D 的逻辑结构图示

例 3 举一个数据结构的例子,叙述其逻辑结构、存储结构和运算三方面的内容。

解答 例如有一份通讯录,如表 1.1 所示,记录了相关人员的电话号码,将其按姓名一人占一行构成表,这个表是一个数据结构,每一行为一个记录,每个记录(包括姓名、工作单位、职务、电话号码)即为一个结点,对于整个表来说,只有一个开始结点(前面无记录)和一个终端结点(后面无记录),其他结点则各有一个也只有一个直接前驱和直接后继(它的前面和后面均有且仅有一个记录)。这几个关系就确定了这个表的逻辑结构。

表 1.1 通讯录

姓名	工作单位	职务	电话号码
吴浩	市交警大队	副大队长	88234567
汪涌	市自来水公司	职工	87342567
...

那么怎样把这个表中的数据存储在计算机里呢?用高级语言如何表示各结点之间的关系呢?是用一段连续的内存单元来存放这些记录(如用数组表示),还是随机存放各结点数据再用指针进行链接呢?这就是存储结构的问题。我们都是从高级语言的层次来讨论这个问题的。例如,若用链式存储方式,结点的数据类型定义如下:

```
struct node{
    char name[8];           //存放姓名的数据域
    char dept[20];         //存放工作单位的数据域
    char duty[10];         //存放职务的数据域
    char num[11];          //存放电话号码的数据域
    struct node * next;    //指向下一个结点的指针
}
```

例 4 求两个 n 阶矩形的乘法 $C=A \times B$, 其算法如下:

```
#define m 100
void maxtrixmult(int n, float a[m][m], float b[m][m], float &c[m][m])
{
    int i, j, k;
    float x;
    for (i = 1; i <= n; i++) ①
    {
        for (j = 1; j <= n; j++) ②
        {
            x = 0; ③
            for (k = 1; k <= n; k++) ④
                x += a[i][k] * b[k][j]; ⑤
            c[i][j] = x; ⑥
        }
    }
}
```

说明该算法的时间复杂度。

解答 该算法中主要语句的频度分别是:

① $n+1$; ② $n(n+1)$; ③ n^2 ; ④ $n^2(n+1)$; ⑤ n^3 ; ⑥ n^2

时间复杂度为所有语句的频度之和, 即 $T(n) = 2n^3 + 3n^2 + 2n + 1 = O(n^3)$ 。

教材习题 1

1. 什么是数据结构? 有关数据结构的讨论涉及哪三个方面?
2. 设有数据逻辑结构: $\text{line} = (D, R)$ 。其中:

$$D = \{01, 02, 03, 04, 05, 06, 07, 08, 09, 10\}$$

$$R = \{r\}$$

$$r = \{\langle 05, 01 \rangle, \langle 01, 03 \rangle, \langle 03, 08 \rangle, \langle 08, 02 \rangle, \langle 02, 07 \rangle, \langle 07, 04 \rangle, \langle 04, 06 \rangle, \langle 06, 09 \rangle, \langle 09, 10 \rangle\}$$

试分析该数据结构属于哪种逻辑结构。

3. 什么是算法? 试根据算法特性解释算法与程序的区别。
4. 设有三个值大小不同的整数 a, b 和 c , 试求:
 - (1) 其中值最大的整数;

(2) 其中值最小的整数;

(3) 其中位于中间值的整数。

5. 为字符串定义一个 ADT (Abstract Data Type, 抽象数据类型), 该 ADT 要包含字符串的常用操作, 每个操作定义一个函数, 每个函数由它的输入输出来定义。

6. 设 n 为整数, 指出下列各算法的时间复杂度。

(1) void prime (int n) //n 为一个正整数

```
{
    int i = 2;
    while ( (n % i) != 0 && i * 1.0 < sqrt(n) )
        i++;
    if ( i * 1.0 > sqrt(n) )
        printf ("%d 是一个素数\n", n);
    else
        printf ("%d 不是一个素数\n", n);
}
```

(2) int sum1 (int n) //n 为一个正整数

```
{
    int p = 1, sum = 0, i;
    for (i = 1; i <= n; i++)
    {
        p *= i;
        sum += p;
    }
    return sum;
}
```

(3) int sum2 (int n) //n 为一个正整数

```
{
    int sum = 0, i, j;
    for (i = 1; i <= n; i++)
    {
        p = 1;
        for (j = 1; j <= i; j++)
            p *= j;
        sum += p;
    }
    return sum;
}
```

7. 考查下列两段描述。它们是否满足算法的特征? 若不满足, 说明违反了哪些特征。

```
(1) void exam1( )
    {
        n = 2;
        while ( n % 2 == 0 )
            n = n + 2;
        printf ( " % d\n" );
    }

(2) void exam2( )
    {
        y = 0;
        x = 5/y;
        printf ( " % d, % d\n", x, y );
    }
```

习题 1 答案及解析

1. 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。数据结构包括三方面的内容: 数据的逻辑结构、数据的存储结构和数据的运算。数据结构课程的内容包括三个层次的五个“要素”, 如表 1.2 所示。

表 1.2 数据结构课程的内容构成

层次	方面	
		数据表示
抽象	逻辑结构	基本运算
实现	存储结构	算法
评价	不同结构的比较及算法分析	

2. 题中的逻辑关系图形如图 1.6 所示。

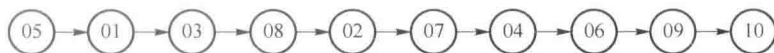


图 1.6 数据的线性结构示意图

在 line 中, 每个数据元素有且仅有一个直接前驱元素(除结构中第一个元素 05 外), 有且仅有一个直接后继元素(除结构中最后一个元素 10 外)。这种数据结构的特点是数据元素之间存在一对一的关系, 即线性关系, 因此本题所给定的数据结构为线性结构。

3. 算法是解决问题的步骤; 程序是算法的代码实现。按照算法的特性来比较算法与

程序的区别如下:

(1) 算法是求解问题的有穷操作序列,算法可以用程序来描述或实现,算法必须终止而程序则不一定;

(2) 算法对于特定的输入有特定的输出,程序提供了确定算法结果的平台;

(3) 算法有输入,算法的输入依靠程序的平台提供;

(4) 算法有输出,算法的输出也靠代码的支持。

4. 按照题目要求,分别设计算法如下:

(1) 求值最大的整数。

```
void max( )
{
    int max, a, b, c;
    max = a;
    if(b > max)
        max = b;
    if(c > max)
        max = c;
    printf("max = %d", max);
}
```

(2) 求值最小的整数。

```
void min( )
{
    int min, a, b, c;
    min = a;
    if(b < min)
        min = b;
    if(c < min)
        min = c;
    printf("min = %d", min);
}
```

(3) 求值位于中间的整数。

```
void minddle( )
{
    int temp, a, b, c;
    if(a > b)
    {
        temp = a;
```