

高等学校计算机基础教育规划教材

数据结构 习题解答与实验指导

樊艳芬 邵 斌 朱绍军 编著

清华大学出版社



高等学校计算机基础教育规划教材

数据结构

习题解答与实验指导

樊艳芬 邵 斌 朱绍军 编著

清华大学出版社
北京

内 容 简 介

本书是由编著者积累多年讲授“数据结构”课程及指导学生实验的教学实践经验编写而成的。编著者力图通过指导学生的实验实践和大量典型例题的解析,帮助学生深入学习、掌握并灵活运用数据结构知识。

本书作为高等学校计算机基础教育规划教材《数据结构》的配套教材,可以配合主教材使用,起到衔接课堂教学与实验教学、课下辅导的作用,可作为高等院校学生学习“数据结构”课程的参考书,对于从事计算机应用及开发的技术人员以及相关专业的自学者,也具有一定的参考价值。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

数据结构习题解答与实验指导/樊艳芬,邵斌,朱绍军编著. —北京:清华大学出版社,2018

(高等学校计算机基础教育规划教材)

ISBN 978-7-302-49331-0

I. ①数… II. ①樊… ②邵… ③朱… III. ①数据结构—高等学校—教学参考资料 IV. ①TP311.12

中国版本图书馆 CIP 数据核字(2018)第 004239 号

责任编辑:颜廷芳

封面设计:常雪影

责任校对:袁芳

责任印制:丛怀宇

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62770175-4278

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:16 字 数:367千字

版 次:2018年9月第1版 印 次:2018年9月第1次印刷

定 价:45.00元

产品编号:077656-01

在计算机教育中,“数据结构”的核心地位与重要作用是普遍公认的,而学习“数据结构”的困难也为广大师生所共识。

根据编著者多年的教学经验,学习数据结构的主要困难在于解题。解题中经常会出现错误的原因在于:一是教材中没有重点解释;二是教师授课时无法做到面面俱到。因此,要学好数据结构,仅仅通过课堂教学或自学来掌握理论知识是远远不够的,还必须加强实践。除了完成数据结构的习题以外,还需要上机完成数据结构实验的若干任务。为此,我们编写了这本《数据结构习题解答与实验指导》。

本书是高等院校计算机基础教育规划教材《数据结构》的配套实验用书。全书共分为3篇:第一篇为学习指导与习题解答,主要帮助读者理解数据结构的各种基本知识点和要点,并且提供了原教材的习题参考解答;第二篇为数据结构实验,本篇要求读者在实验前做好充分准备,然后利用课内学时和课外时间进行上机实践,实验后认真书写实验报告;第三篇为数据结构课程设计,此部分含大型作业题,主要帮助读者在完成数据结构的实验之外,再进一步去完成数据结构课程设计的若干实践任务,以帮助读者上机调试、运行各种典型的算法和自己编制的算法,从实践中得到锻炼和提高,从而学会运用理论知识去解决软件开发中的实际问题,达到学以致用目的。

本书使用C语言来描述算法和数据结构,各实验中的程序都在C-free或Visual C++ 6.0中调试通过,以方便读者在计算机上进行实践,有助于理解算法的实质和基本思想。

本书内容既配合原教材,又有相对的独立性,内容安排由浅入深,可以循序渐进地培养学生的实践技能。本书既可以作为高校计算机及相关专业本科生的配套教材,也可以作为专科和成人教育的辅助教材。另外,本书还可供从事计算机应用等工作的工程技术人员参考,读者只需掌握C语言编程的基本技术就可以学习本书。

由于编著者水平有限,因此书中难免有不足之处,殷切希望广大读者批评、指正。

编著者

2018年4月

第一篇 学习指导与习题解答

第 1 章 概论	3
1.1 学习指导	3
1.1.1 基本知识点	3
1.1.2 要点分析	4
1.2 习题参考解答	5
第 2 章 线性表	8
2.1 学习指导	8
2.1.1 基本知识点	8
2.1.2 要点分析	9
2.2 习题参考解答	10
第 3 章 栈和队列	16
3.1 学习指导	16
3.1.1 基本知识点	16
3.1.2 要点分析	17
3.2 习题参考解答	18
第 4 章 串	26
4.1 学习指导	26
4.1.1 基本知识点	26
4.1.2 要点分析	27
4.2 习题参考解答	27
第 5 章 数组和广义表	32
5.1 学习指导	32
5.1.1 基本知识点	32

5.1.2	要点分析	33
5.2	习题参考解答	33
第6章	树和二叉树	38
6.1	学习指导	38
6.1.1	基本知识点	38
6.1.2	要点分析	41
6.2	习题参考解答	43
第7章	图	49
7.1	学习指导	49
7.1.1	基本知识点	49
7.1.2	要点分析	52
7.2	习题参考解答	52
第8章	查找	65
8.1	学习指导	65
8.1.1	基本知识点	65
8.1.2	要点分析	66
8.2	习题参考解答	67
第9章	排序	73
9.1	学习指导	73
9.1.1	基本知识点	73
9.1.2	要点分析	73
9.2	习题参考解答	74

第二篇 数据结构实验

第10章	数据结构实验概述	85
10.1	实验教学的目的	85
10.2	实验教学的主要内容	85
10.3	实验步骤	86
10.4	实验报告示例	88
第11章	数据结构实验安排	91
11.1	线性表	91

11.1.1	顺序表操作(验证实验)	91
11.1.2	单链表操作(验证实验)	94
11.1.3	约瑟夫环问题(设计实验)	97
11.1.4	一元多项式加(减)法计算器(设计实验)	98
11.2	栈和队列	98
11.2.1	顺序栈操作(验证实验)	98
11.2.2	链栈操作(验证实验)	101
11.2.3	顺序队列操作(验证实验)	105
11.2.4	链队列操作(验证实验)	111
11.2.5	括号匹配的检验(设计实验)	114
11.2.6	栈逆置队列(设计实验)	115
11.3	树和二叉树	116
11.3.1	二叉树操作(验证实验)	116
11.3.2	哈夫曼编码问题(设计实验)	122
11.4	图	123
11.4.1	无向图的邻接矩阵操作(验证实验)	123
11.4.2	无向图的邻接表操作(验证实验)	124
11.4.3	单源点最短路径问题(设计实验)	128
11.4.4	多源点最短路径问题(设计实验)	128
11.5	查找	129
11.5.1	查找算法(验证实验)	129
11.5.2	哈希表设计(设计实验)	139
11.6	排序	140
11.6.1	直接插入排序(设计实验)	140
11.6.2	希尔排序(设计实验)	142
11.6.3	冒泡排序(设计实验)	143
11.6.4	直接选择排序(设计实验)	144
11.6.5	快速排序(设计实验)	144
11.6.6	归并排序(设计实验)	145

第三篇 数据结构课程设计

第 12 章	数据结构课程设计概述	149
12.1	课程设计的目的	149
12.2	课程设计的时间安排	150
12.3	课程设计的指导教师	150
12.4	课程设计的选题内容和要求	150

12.5	课程设计的实施步骤	151
12.6	课程设计总结报告的撰写规范	151
12.7	课程设计的上交材料	152
12.8	课程设计的成绩评定	152
第 13 章	数据结构课程设计安排	153
13.1	线性表(课程设计 1)	153
13.1.1	顺序表的就地逆置	153
13.1.2	单链表的就地逆置	157
13.2	栈(课程设计 2)	160
13.2.1	用栈逆置一个单链表	163
13.2.2	共享栈的设计	166
13.3	队列(课程设计 3)	168
13.4	树和二叉树(课程设计 4)	175
13.4.1	二叉树的生成	175
13.4.2	最优二叉树(哈夫曼树)的建立	178
13.5	图(课程设计 5)	181
13.5.1	图的存储结构	181
13.5.2	图的遍历	184
13.5.3	图的应用	186
13.6	查找(课程设计 6)	188
13.7	排序(课程设计 7)	194
13.8	大型作业题(课程设计 8)	207
13.9	数据结构课程设计补充题目	210
第 14 章	数据结构课程设计案例——图书管理信息系统的设计与实现	224
14.1	设计要求	224
14.2	设计分析	225
14.2.1	数据文件类型设计	225
14.2.2	系统功能算法描述	227
14.3	设计的实现	231
14.4	测试运行实例	241
14.4.1	主控菜单的设计	241
14.4.2	测试运行实例	242
参考文献		248

第一篇

学习指导与习题解答

概 论

本章要点

- 数据结构的概念
- 逻辑结构、存储结构和运算集合
- 算法时间、空间复杂度的分析

本章学习目标

- 了解数据、数据元素等概念
- 掌握数据结构的概念
- 掌握逻辑结构、存储结构和运算集合的关系
- 掌握算法时间复杂度的分析方法
- 了解算法空间复杂度的分析方法

1.1 学习指导

1.1.1 基本知识点

1. 数据结构的基本概念

(1) 数据(data): 是信息的载体, 是对客观事物的符号表示, 它能够被计算机识别、存储和加工处理。可以说, 数据是计算机程序加工的“原料”。目前, 图像、声音、视频等都可以通过编码而由计算机处理, 因此它们也属于数据的范畴。

(2) 数据元素(data element): 是数据的基本单位, 通常在计算机程序中作为一个整体进行考虑和处理。数据元素也称为元素、节点或记录。有时, 一个数据元素可以有若干个数据项(也称字段、域), 数据项是数据不可分割的最小单位。

(3) 数据对象(data object): 是性质相同的数据元素的集合, 它是数据的一个子集。例如, 整数数据对象是集合 $N = \{0, \pm 1, \pm 2, \pm 3, \dots\}$; 大写字母字符数据对象是集合 $C = \{ 'A', 'B', \dots, 'Z' \}$ 。需要注意的是, 计算机中的整数数据对象集合 N_1 应该是上述集合 N 的一个子集, $N_1 = \{0, \pm 1, \pm 2, \dots, \pm \text{maxint}\}$, 其中, maxint 是依赖于所使用的计算机和语言的最大整数。

(4) 数据类型(data type): 是计算机程序中的数据对象以及定义在这个数据对象集合上的一组操作的总称。例如, C语言中的整数类型是区间 $[-\text{maxint}, \text{maxint}]$ 上的整数, 在这个集合上可以进行加、减、乘、整除、求余等操作。

(5) 数据结构(data structure): 是指数据对象以及该数据对象集合中的数据元素之间的相互关系(数据元素的组织形式)。

(6) 数据的逻辑结构: 是指数据元素之间的逻辑关系。数据的逻辑结构通常有下列 4 类。

- ① 集合: 其中的数据元素之间除了“属于同一个集合”的关系以外, 别无其他关系。
- ② 线性结构: 其中的数据元素之间存在一对一的关系。
- ③ 树形结构: 其中的数据元素之间存在一对多的关系。
- ④ 图状结构(或称网状结构): 其中的数据元素之间存在多对多的关系。

(7) 数据的存储结构: 数据元素以及它们之间的相互关系在计算机存储器内的表示(又称映像), 也称为数据的物理结构。

数据的存储结构可采用以下 4 种基本的存储方法得到。

- ① 顺序存储方法。
- ② 链接存储方法。
- ③ 索引存储方法。
- ④ 散列存储方法。

上述 4 种基本的存储方法, 既可以单独使用, 也可以组合起来对数据结构进行存储映像。同一种逻辑结构, 若采用不同的存储方法, 则可以得到不同的存储结构。

2. 算法和算法分析

(1) 算法(algorithm): 是对特定问题求解步骤的一种描述, 它是指令的有限序列, 其中每一条指令表示一个或多个操作。

算法具有 5 个主要的特性: 有穷性、确定性、可行性、输入和输出。

(2) 算法时间复杂度: 算法在运行时所要花费的时间。

(3) 算法空间复杂度: 算法在运行时所要花费的空间。

(4) 算法分析则主要考察算法的时间复杂度与空间复杂度。在空间不受限制的情况下, 时间复杂度决定了一个算法的好坏。

1.1.2 要点分析

1. 数据结构研究内容

数据结构的内容可归纳为 3 个部分: 逻辑结构、存储结构和运算集合。按某种逻辑关系组织起来的一批数据, 按一定的映像方式把它存放在计算机的存储器中, 并在这些数据上定义了一个运算的集合, 这就是一个数据结构。

一种数据结构究竟能有哪些运算, 是我们给它定义的。例如, 在线性表中我们定义了以下 9 种基本操作。

InitList(L): 初始化操作, 构造一个空线性表 L 。

ClearList(L): 清除线性表 L 的内容, 将 L 置为空表。

ListLength(L): 求表长(表中元素个数)。

Ins(L, i, Item): 插入数据。

Del(L, i): 删除数据。

GetNode(L, i): 获取表 L 中位置 i 的节点值。

Loc($L, Item$): 定位(按值查找)。

GetPrior($L, Item, p$): 获取值为 $Item$ 的节点的前趋节点。

GetNext($L, Item, p$): 获取值为 $Item$ 的节点的后继节点。

但在一个具体实际应用中,我们可能并不需要上述全部 9 种运算,而只应用其中的几种。在编写程序时,只对需要的运算编写代码。

2. 算法时间复杂度的分析

一个算法所耗费的时间是算法中所有语句执行时间之和,而每条语句的执行时间是该语句的执行次数(频度)与该语句执行一次所需时间(略,因机器不同而不同)的乘积。

为了消除机器硬件给算法时间分析带来的影响,我们并不真正计算算法运行实际所耗费的时间,而是以语句执行的次数代替语句执行的时间。通常,一个算法是由控制结构(顺序、选择和循环)和“原操作”(指固有数据类型的操作,一条基本语句)构成的,而算法时间复杂度取决于两者的综合效果。算法时间复杂度就是所有“原操作”执行次数之和,它一般与输入数据量 n 相关。

一般的,一个算法所耗费的时间将随输入数据量 n 的增大而增大。所以算法时间复杂度是输入数据量 n 的函数,这时就称该算法的时间代价为 $T(n)$ 。

评价算法时间复杂度,就是设法找出 $T(n)$ 和 n 的关系,即求出 $T(n)$ 。如 $T(n) = 3n^2 + 4n + 10$ 。在进行算法分析的时候,一般只考虑当 n 充分大时的情况,所以常常用渐进时间复杂度来表示。当 n 充分大时, $3n^2$ 是时间耗费的主要方面, $4n$ 及常数 10 都是次要方面。 $3n^2$ 中的 n^2 又是主要方面,常数 3 又是次要方面。用渐进符号 O 来表示,则 $T(n) = O(n^2)$ 。

1.2 习题参考解答

1. 填空题

(1) 数据的逻辑结构是数据元素之间的逻辑关系,通常有下列 4 类: _____、_____、_____和_____。

【答】集合、线性结构、树形结构、图状结构。

(2) 数据的存储结构是数据在计算机存储器中的表示,主要有 4 种基本存储方法: _____、_____、_____和_____。

【答】顺序存储方法、链式存储方法、索引存储方法、散列存储方法。

2. 选择题

(1) 一个算法必须在执行有穷步之后结束。这是算法的()。

- A. 正确性 B. 有穷性 C. 确定性 D. 可行性

【答】B。

(2) 算法的每一步,必须有确切的定义。也就是说,对于每步需要执行的动作必须严格、清楚地给出规定。这是算法的()。

- A. 正确性 B. 有穷性 C. 确定性 D. 可行性

【答】C。

(3) 算法原则上都是能够由机器或人完成的。整个算法好像是一个解决问题的“工作序列”，其中的每一步都是我们力所能及的一个动作。这是算法的()。

- A. 正确性 B. 有穷性 C. 确定性 D. 可行性

【答】D。

3. 简答题

(1) 算法与程序有何异同?

【答】尽管算法的含义与程序非常相似,但两者还是有所区别的。首先,一个程序不一定满足有穷性,因此它不一定是算法。例如,系统程序中的操作系统,只要整个系统不被破坏,它就永远不会停止,即使没有作业要处理,它仍处于等待循环中,以待一个新作业的进入。因此操作系统就不是一个算法。其次,程序中的指令必须是计算机可以执行的,而算法中的指令却无此限制。如果一个算法采用机器可执行的语言来书写,那么它就是一个程序。

(2) 什么是数据结构? 试举一个简单的例子说明。

【答】数据结构是指数据对象以及该数据对象集合中的数据元素之间的相互关系(数据元素的组织形式)。例如,队列的逻辑结构是线性表(先进先出);队列在计算机中既可以采用顺序存储也可以采用链式存储;对队列可进行删除、插入数据元素以及判断是否为空队列、将队列置空等操作。

(3) 什么是数据的逻辑结构? 什么是数据的存储结构?

【答】数据元素之间的逻辑关系,也称为数据的逻辑结构。数据元素以及它们之间的相互关系在计算机存储器内的表示(又称映像),称为数据的存储结构,也称数据的物理结构。

(4) 什么是算法? 算法有哪些特性?

【答】算法是对特定问题求解步骤的一种描述,它是指令的有限序列,其中每一条指令表示一个或多个操作;此外,一个算法还具有下列5个特性。

① 有穷性: 一个算法必须在执行有穷步之后结束,即算法必须在有限时间内完成。

② 确定性: 算法中每一步必须有确切的含义,不会产生二义性。并且,在任何条件下,算法只有唯一的一条执行路径,即对于相同的输入只能得出相同的输出。

③ 可行性: 一个算法是能被执行的,即算法中的每一步都可以通过已经实现的基本运算执行有限次得以实现。

④ 输入: 一个算法有零个或多个输入,它们是算法开始时对算法给出的初始量。

⑤ 输出: 一个算法有一个或多个输出,它们是与输入有特定关系的量。

4. 算法分析题

(1) 将下列复杂度由小到大重新排序: 2^n 、 $n!$ 、 n^2 、10000、 $\log_2 n$ 、 $n \times \log_2 n$ 。

【答】 $10000 < \log_2 n < n \times \log_2 n < n^2 < 2^n < n!$ 。

(2) 用大O表示法描述下列复杂度。

① $5n^{5/2} + n^{2/5}$ 。

【答】 $O(n^{5/2})$ 。

② $6 \times \log_2 n + 9n$ 。

【答】 $O(n)$ 。

③ $3n^4 + n \times \log_2 n$ 。

【答】 $O(n^4)$ 。

④ $n \times \log_2 n + n \times \log_3 n$ 。

【答】 $O(n \times \log_2 n)$ 。(3) 设 n 为正整数, 请用大 O 表示法描述下列程序段的时间复杂度。

```
① i=1; k=0;
   while(i<n)
   {k=k+10 * i; i++;
   }
```

【答】 $O(n)$ 。

```
② i=0; k=0;
   do{k=k+10 * i;
      i++;
   }while(i<n);
```

【答】 $O(n)$ 。

```
③ i=1; j=0;
   while(i+j<=n)
   {if(i>j) j++;
     else i++;
   }
```

【答】 $O(n)$ 。

```
④ x=n; /* n 是常数且 n>1 */
   while(x>= (y+1) * (y+1))
   y++;
```

【答】 $O(\sqrt{n})$ 。

```
⑤ for(i=1; i<=n; i++)
   for(j=1; j<=i; j++)
   for(k=1; k<=j; k++)
   x+=c; (c 为常数)
```

【答】 $O(n^3)$ 。

```
⑥ x=91; y=100;
   while(y>0)
   {if(x>100) {x-=10; y--;}
     else x++;
   }
```

【答】本题只能计算 if 语句的频度。上述程序实质上是一个双重循环, 对于每个 y 值 ($y > 0$), if 语句执行 11 次, 其中 10 次执行 $x++$ 。因此, if 语句的频度为 $11 \times 100 = 1100$ 次。

线性表

本章要点

- 线性表的概念
- 线性表的顺序存储结构
- 线性表的链式存储结构
- 顺序存储结构下线性表的各种操作的实现
- 链式存储结构下线性表的各种操作的实现

本章学习目标

- 理解顺序表、链表与线性表的区别
- 掌握顺序存储结构的数据类型定义方法
- 掌握顺序存储结构下线性表的各种操作实现
- 掌握链式存储结构的数据类型定义方法
- 掌握链式存储结构下线性表的各种操作实现
- 了解双向链表
- 了解循环链表

2.1 学习指导

2.1.1 基本知识点

1. 线性表的基本概念

(1) 线性结构的基本特征如下。

- ① 有且只有一个“第一元素”。
- ② 有且只有一个“最后元素”。
- ③ 除第一元素之外,其他元素都有唯一的直接前趋。
- ④ 除最后元素之外,其他元素都有唯一的直接后继。

(2) 线性表:是具有相同数据类型的 $n(n \geq 0)$ 个数据元素的有限序列。

2. 线性表的顺序存储

顺序表:采用顺序存储结构的线性表称为顺序表。

顺序表 L 的第 i 个元素的存储位置和第一个元素的存储位置的关系如下。

$$\text{LOC}(a_i) = \text{LOC}(a_1) + (i-1) \times m$$

式中, $LOC(a_1)$ 是线性表的第一个数据元素的存储位置, 通常称为线性表的起始位置或基地址。

顺序表的存储结构如下。

```
typedef int datatype;           /* 定义表元素的类型 */
#define maxsize 1024           /* 线性表的最大长度 */
typedef struct
{
    datatype elem[maxsize];     /* 存放表节点的数组 */
    int length;                 /* 表长 */
}sequenlist;
```

3. 线性表的链式存储

链表: 以链式结构存储的线性表称为链表。

链表包括单链表、双向链表和循环链表。

单链表的存储结构如下。

```
typedef struct LNode
{
    ElemType data;
    Struct LNode * next;
}LinkList;
LinkList * L, * head;
```

双向链表的存储结构如下。

```
typedef struct DNode
{
    struct DNode * prior;
    ElemType data;
    Struct DNode * next;
}DLinkList;
DLinkList * DL, * p;
```

单向循环链表的结构与单链表相同, 双向循环链表的结构与双向链表相同。不同之处在于, 单向循环链表最后一个节点的 `next` 指针指向第一个节点, 双向循环链表的第一个节点的 `prior` 指针指向最后一个节点, 在非循环链表中, 这两个特殊的指针均为 `NULL`。

线性表最主要操作有初始化操作、插入、删除。

2.1.2 要点分析

1. 线性表顺序存储与链式存储的比较

从时间的角度考虑, 在按位置查找数据, 或者在查找元素的前趋和后继等方面, 顺序存储有着较大的优势。在插入数据、删除数据时, 链式存储就有较大的优势, 这是由于在链表中只要修改指针即可做到; 而在顺序表中进行插入和删除数据, 平均要移动表中将近一半的数据元素。