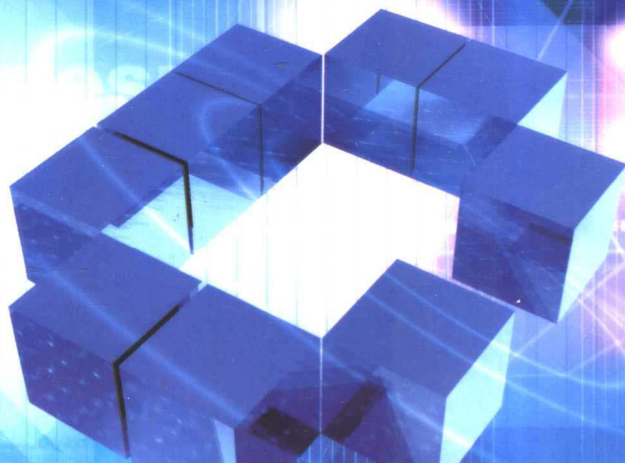


● 高等学校21世纪计算机教材

# 数据结构

## 习题解析与实验指导 (C语言描述)

刘怀亮 编著



冶金工业出版社

## 内 容 简 介

数据结构是计算机学科的核心专业课程之一，它是软件开发的重要基础。本书是 2004 年出版的《数据结构（C 语言描述）》一书的配套用书，本书分为三个部分：第一部分为习题解析，具体包括数据结构核心内容的重点阐述、数据结构经典算法讲解及习题；第二部分为实验指导，给出了相关的实验内容及其指导，包括实验目的和实验内容；第三部分为全真模拟试卷，包括 10 套全真模拟试卷及其答案。

本书内容新颖、文字清晰、逻辑严密、实用性较强，既可以作为计算机专业本科、大专和高职类院校数据结构课程的教材，也可以作为研究生入学考试、计算机技术与软件专业技术资格（水平）考试、计算机等级考试（三级或四级）、计算机专业自学考试的参考用书和各类培训班的教材，同时也可供从事计算机科学与技术相关工作的科技工作者参考。

### 图书在版编目（CIP）数据

数据结构习题解析与实验指导 / 刘怀亮编著. —北京：  
冶金工业出版社，2005.2  
ISBN 7-5024-3705-3

I. 数... II. 刘... III. 数据结构—自学参考资料  
IV. TP311.12

中国版本图书馆 CIP 数据核字（2004）第 142922 号

出版人 曹胜利（北京沙滩嵩祝院北巷 39 号，邮编 100009）

责任编辑 程志宏

湛江蓝星南华印务公司印刷；冶金工业出版社发行；各地新华书店经销

2005 年 2 月第 1 版，2005 年 2 月第 1 次印刷

787mm × 1092mm 1/16； 20.5 印张； 471 千字； 318 页

30.00 元

冶金工业出版社发行部 电话：（010）64044283 传真：（010）64027893

冶金书店 地址：北京东四西大街 46 号（100711） 电话：（010）65289081

（本社图书如有印装质量问题，本社发行部负责退换）

# 前 言

## 一、关于本书

数据结构是计算机学科的核心专业课程之一，它是软件开发的重要基础。数据结构不仅为程序设计提供了方法论性质的指导，还在更高层次上总结了现实生活中多种数据的计算机处理方法。作为计算机科学的主干课程，数据结构是一门实用性很强的课程，学好它不但要在理论上打好基础，更要在实践上多下功夫。

本书是作者结合在计算机教学和实践中的经验，依据计算机专业的相关教学大纲编写的。本书是2004年出版的《数据结构（C语言描述）》一书的配套用书，作者在教学和实践中积累、收集、验证和编写的有关数据结构的重要算法及相关题解都在本书中得以体现，且涵盖了数据结构在研究生入学考试、计算机技术与软件专业技术资格（水平）考试、计算机等级考试和自学考试等各类考试中涉及到的大部分经典算法、常见考题和疑难问题的分析与解答。

## 二、本书结构

本书分为三个部分：第一部分为习题解析，包括第1~9章，主要内容包括数据结构重点内容讲解、典型例题解析以及习题；第二部分为实验指导，包括第10~17章，主要包括实验目的和实验内容；第三部分为全真模拟试卷，包括10套模拟试卷。具体内容安排如下：

第一部分（第1~9章）：

第1章：绪论。主要介绍了数据结构的基本概念和重要术语、算法分析的相关内容、典型例题解析以及习题。

第2章：线性表。主要介绍了线性表和顺序表的相关知识、线性表的链式存储结构、典型例题解析以及习题。

第3章：栈和队列。主要介绍了栈和队列的相关内容、典型例题解析以及习题。

第4章：串。主要介绍了串的存储结构和串的模式匹配、典型例题解析以及习题。

第5章：数组和广义表。主要介绍了数组的顺序存储结构及其实现、特殊矩阵和稀疏矩阵的压缩存储与广义表、典型例题解析以及习题。

第6章：树和二叉树。主要介绍了树的基本概念和术语、二叉树、线索二叉树、树和森林以及哈夫曼树、典型例题解析以及习题。

第7章：图。主要介绍了图的基本概念和存储结构、图的遍历、最小生成树、最短路径和拓扑排序、典型例题解析以及习题。

第8章：查找。主要介绍了查找的基本概念、顺序查找、折半查找、分块查找、动态查找表、哈希表查找、典型例题解析以及习题。

第9章：排序。主要介绍了排序的基本概念、插入排序、交换排序、选择排序、归并排序、基数排序、典型例题解析以及习题。

第二部分(第10~17章):主要介绍了8个实验,包括线性表、栈和队列、串、数组和广义表、树和二叉树、图、查找及排序,且各个实验均分为实验目的和实验内容两部分进行阐述。

第三部分:主要包括10套全真模拟试卷,以选择题、填空题等多种题型对数据结构主要知识点从不同方面和不同难易程度进行了全面考查。

### 三、本书特点

本书中的所有算法、例题、习题、实验和模拟题具有较大普遍性和较强的实用性,但不是学习数据结构所必须了解的内容,也是参加各类考试必须掌握的重点内容,同时也是在程序设计和软件开发中经常遇到的实际问题。

本书内容新颖、文字清晰、逻辑严密、语言通俗易懂。同时本书习题覆盖面广,既收集了较容易的题目,也收集了难度适中和较高难度的题目,如一些高校计算机专业硕士研究生入学考试和计算机技术与软件专业技术资格(水平)考试中的数据结构试题。

### 四、适用对象

本书可以作为计算机专业本科、大专和高职类院校数据结构课程的教材,也可以作为研究生入学考试、计算机技术与软件专业技术资格(水平)考试、计算机等级考试(三级或四级)、计算机专业自学考试的参考用书和各类培训班的教材,同时也可供从事计算机科学与技术相关工作的科技工作者参考。

本书在编写过程中得到了计算机工程研发中心的多位老师、研究生和工程研发人员的热情帮助,在此表示衷心感谢。

由于编写时间仓促,作者水平有限,书中错误和疏漏之处在所难免,恳请读者批评指正。

虽然经过严格的审核、精细的编辑,本书在质量上有了一定的保障,但我们的目标是力求尽善尽美,欢迎广大读者和专家对我们的工作提出宝贵建议,联系方法如下:

电子邮件: [service@cnbook.net](mailto:service@cnbook.net)

网址: [www.cnbook.net](http://www.cnbook.net)

此外,本书习题参考答案也可从该网站免费下载,该网站还有一些其他相关书籍的介绍,可以方便读者选购参考。

编者

2004年12月

# 目 录

## 第一部分 习题解析

<b>第 1 章 绪论</b> .....	<b>2</b>
1.1 重点内容讲解 .....	2
1.1.1 数据结构课程的发展及其地位 .....	2
1.1.2 数据结构的基本概念和术语 .....	2
1.1.3 数据的逻辑结构和存储结构 .....	2
1.1.4 算法及其描述 .....	4
1.1.5 算法分析 .....	4
1.2 典型例题解析 .....	5
小结 .....	7
习题一 .....	7
一、选择题 .....	7
二、填空题 .....	7
三、判断题 .....	9
四、综合题 .....	9
<b>第 2 章 线性表</b> .....	<b>11</b>
2.1 重点内容讲解 .....	11
2.1.1 线性表的定义与特征 .....	11
2.1.2 顺序表的定义和表示 .....	11
2.1.3 线性表的链式存储结构 .....	12
2.2 典型例题解析 .....	14
小结 .....	22
习题二 .....	22
一、选择题 .....	22
二、填空题 .....	23
三、判断题 .....	24
四、综合题 .....	25
<b>第 3 章 栈和队列</b> .....	<b>26</b>
3.1 重点内容讲解 .....	26
3.1.1 栈 .....	26
3.1.2 队列 .....	27
3.2 典型例题解析 .....	28
小结 .....	35

习题三 .....	35
一、选择题 .....	35
二、填空题 .....	36
三、判断题 .....	38
四、综合题 .....	38
<b>第4章 串 .....</b>	<b>40</b>
4.1 重点内容讲解 .....	40
4.1.1 串的基本概念 .....	40
4.1.2 串的存储结构 .....	40
4.1.3 串的模式匹配 .....	41
4.2 典型例题解析 .....	43
小结 .....	49
习题四 .....	49
一、选择题 .....	49
二、填空题 .....	49
三、判断题 .....	50
四、综合题 .....	50
<b>第5章 数组和广义表 .....</b>	<b>52</b>
5.1 重点内容讲解 .....	52
5.1.1 数组的定义 .....	52
5.1.2 数组的顺序存储结构和实现 .....	52
5.1.3 特殊矩阵和稀疏矩阵的压缩存储 .....	54
5.1.4 广义表 .....	56
5.2 典型例题解析 .....	58
小结 .....	62
习题五 .....	63
一、选择题 .....	63
二、填空题 .....	63
三、判断题 .....	64
四、综合题 .....	65
<b>第6章 树和二叉树 .....</b>	<b>66</b>
6.1 重点内容讲解 .....	66
6.1.1 树的基本概念和术语 .....	66
6.1.2 二叉树 .....	68
6.1.3 遍历二叉树 .....	69
6.1.4 线索二叉树 .....	69
6.1.5 树和森林 .....	70

6.1.6 哈夫曼树 .....	73
6.2 典型例题解析 .....	73
小结 .....	78
习题六 .....	79
一、选择题 .....	79
二、填空题 .....	80
三、判断题 .....	83
四、综合题 .....	84
<b>第7章 图 .....</b>	<b>86</b>
7.1 重点内容讲解 .....	86
7.1.1 图的基本概念 .....	86
7.1.2 图的存储结构 .....	88
7.1.3 图的遍历 .....	89
7.1.4 最小生成树 .....	89
7.1.5 最短路径 .....	90
7.1.6 拓扑排序 .....	91
7.2 典型例题解析 .....	91
小结 .....	95
习题七 .....	96
一、选择题 .....	96
二、填空题 .....	97
三、判断题 .....	100
四、综合题 .....	100
<b>第8章 查找 .....</b>	<b>102</b>
8.1 重点内容讲解 .....	102
8.1.1 查找的基本概念 .....	102
8.1.2 顺序查找 .....	103
8.1.3 折半查找 .....	103
8.1.4 分块查找 .....	104
8.1.5 动态查找表 .....	105
8.1.6 哈希表查找 .....	106
8.2 典型例题解析 .....	108
小结 .....	111
习题八 .....	111
一、选择题 .....	111
二、填空题 .....	113
三、判断题 .....	115

四、综合题 .....	116
<b>第9章 排序 .....</b>	<b>117</b>
9.1 重点内容讲解 .....	117
9.1.1 排序的基本概念 .....	117
9.1.2 插入排序 .....	118
9.1.3 交换排序 .....	119
9.1.4 选择排序 .....	120
9.1.5 归并排序 .....	121
9.1.6 基数排序 .....	121
9.1.7 各种内部排序方法的比较 .....	122
9.2 典型例题解析 .....	122
小结 .....	127
习题九 .....	128
一、选择题 .....	128
二、填空题 .....	129
三、判断题 .....	130
四、综合题 .....	131

## 第二部分 实验指导

<b>第10章 实验指导一——线性表 .....</b>	<b>134</b>
10.1 实验目的 .....	134
10.2 实验内容 .....	134
10.2.1 顺序表的基本运算 .....	134
10.2.2 单链表的基本运算 .....	136
10.2.3 一元多项式计算 .....	140
<b>第11章 实验指导二——栈和队列 .....</b>	<b>143</b>
11.1 实验目的 .....	143
11.2 实验内容 .....	143
11.2.1 算术表达式求值 .....	143
11.2.2 迷宫问题求解 .....	147
11.2.3 键盘缓冲区问题 .....	151
<b>第12章 实验指导三——串 .....</b>	<b>154</b>
12.1 实验目的 .....	154
12.2 实验内容 .....	154
12.2.1 串 KMP 算法的 C 语言基本实现 .....	154
12.2.2 求两个串的最长公共子串 .....	156
12.2.3 串的匹配与逆置 .....	158



<b>第 13 章 实验指导四——数组和广义表</b> .....	<b>160</b>
13.1 实验目的.....	160
13.2 实验内容.....	160
13.2.1 n 阶魔方阵的求解.....	160
13.2.2 鞍点问题的求解.....	162
13.2.3 十字链表基本操作的实现.....	163
13.2.4 广义表基本操作的实现.....	168
<b>第 14 章 实验指导五——树和二叉树</b> .....	<b>173</b>
14.1 实验目的.....	173
14.2 实验内容.....	173
14.2.1 二叉树各种遍历操作的实现.....	173
14.2.2 二叉树基本操作的实现.....	178
14.2.3 哈夫曼树的构造与显示.....	181
<b>第 15 章 实验指导六——图</b> .....	<b>185</b>
15.1 实验目的.....	185
15.2 实验内容.....	185
15.2.1 通信工程造价问题求解.....	185
15.2.2 最短交通时间问题.....	187
15.2.3 工程拓扑排序问题.....	189
<b>第 16 章 实验指导七——查找</b> .....	<b>194</b>
16.1 实验目的.....	194
16.2 实验内容.....	194
16.2.1 二叉排序树查找的基本操作.....	194
16.2.2 哈希表查找的设计.....	198
<b>第 17 章 实验指导八——排序</b> .....	<b>202</b>
17.1 实验目的.....	202
17.2 实验内容.....	202
17.2.1 多种排序算法的比较.....	202
17.2.2 归并排序算法的实现.....	209
17.2.3 基数排序算法的实现.....	212
<b>第三部分 全真模拟试卷</b>	
<b>全真模拟试卷</b> .....	<b>216</b>
全真模拟试卷一.....	216
全真模拟试卷二.....	221
全真模拟试卷三.....	227

---

全真模拟试卷四 .....	234
全真模拟试卷五 .....	240
全真模拟试卷六 .....	246
全真模拟试卷七 .....	253
全真模拟试卷八 .....	259
全真模拟试卷九 .....	265
全真模拟试卷十 .....	272
全真模拟试卷参考答案 .....	279
<b>参考文献 .....</b>	<b>317</b>

## 第一部分 习题解析

本部分主要是对各重要知识点的回顾，同时穿插重点、难点知识的典型例题解析及相应练习。具体内容安排如下：

- 第1章 绪论
- 第2章 线性表
- 第3章 栈和队列
- 第4章 串
- 第5章 数组和广义表
- 第6章 树和二叉树
- 第7章 图
- 第8章 查找
- 第9章 排序

# 第 1 章 绪 论

本章主要介绍数据结构的基本概念、基本术语、算法描述和分析的相关内容及相关典型例题分析。通过本章的学习，读者要掌握数据结构的一些基本概念、术语和分析算法时间复杂度的方法。

## 1.1 重点内容讲解

### 1.1.1 数据结构课程的发展及其地位

数据结构是介于数学、计算机硬件和计算机软件之间的一门核心课程。

20 世纪 60 年代，美国计算机界首先使用了数据结构这一名称。1968 年美国著名的计算机专家唐·欧·克努特教授创立了数据结构的最初体系。

从 60 年代末期到 70 年代初期，结构化程序设计成为程序设计方法学和软件设计的主要内容，人们越来越重视数据结构。

从 70 年代中期到 80 年代初期，人们对数据结构的研究逐渐产生了浓厚的兴趣，各种版本的数据结构著作相继出现。

在计算机科学中，数据结构不仅是一般程序设计的基础，而且是设计和实现程序编译系统、计算机操作系统、数据库系统和其他系统程序与大型应用程序的基础。

### 1.1.2 数据结构的基本概念和术语

#### 1. 数据

数据是信息的载体，是对客观事物的符号表示，它能够被计算机识别、存储、加工和处理，数据是能够有效的输入到计算机中并且能够被计算机处理的符号的总称。

#### 2. 数据元素

数据元素也称为元素、结点、顶点、记录等，它是数据的基本单位。

#### 3. 数据项

数据项是数据不可分割的最小单元，它是具有独立含义的最小标识单位。

#### 4. 数据结构

数据结构是数据之间的相互关系，即数据的组织形式。

数据结构包含逻辑结构、存储结构和数据的运算三方面的内容。

### 1.1.3 数据的逻辑结构和存储结构

#### 1. 数据的逻辑结构

数据的逻辑结构是指数据之间的逻辑关系。逻辑结构是从逻辑关系上描述数据，它与具体存储无关，是独立于计算机的。因此逻辑结构可以看作是从具体问题抽象出来的数学模型。

设  $D$  表示数据元素（或称为结点、记录等）的集合， $R$  表示  $D$  上的关系集合，即  $R$

反映了 D 中各元素的前驱、后继关系。则一个数据结构 B 可以表示为:

$$B = (D, R)$$

的二元组的形式。

数据的逻辑结构可以分为集合、线性结构、树型结构和图型结构四种基本类型。

#### 1) 集合

集合结构中的元素除了仅仅只是同属于一个集合, 不存在什么关系, 即  $R = \{\}$ 。

#### 2) 线性结构

线性结构中的元素是一对一的关系, 这种结构的特征是: 如果结构是非空集, 则有且只有一个开始结点和一个终端结点, 并且所有结点最多只能有一个直接前驱和一个直接后继。

#### 3) 树型结构

树型结构中的元素是一对多的关系。在这种逻辑结构中, 除了根结点外, 其他所有结点有且只有一个前驱, 但是所有结点都可以有多个后继。

#### 4) 图型结构

图型结构或图状结构中的元素是多对多的关系。在这种逻辑结构中, 所有结点均可以有多个前驱和多个后继。

也可将树型结构、集合和图型结构称为非线性结构, 这样数据的逻辑结构就可以分为线性结构和非线性结构这两大类。

在不易混淆的情况下, 常常将数据的逻辑结构简称为数据结构。

### 2. 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内的表示。存储结构是逻辑结构在计算机里的实现, 也称之为映像。

数据的存储结构可以用顺序存储方式、链接存储方式、索引存储方式和散列存储方式四种基本的存储方法得到。

#### 1) 顺序存储方式

顺序存储方式是指逻辑上相邻的结点被存储到物理上也相邻的存储单元中。这种存储结构只存储结点的数值, 不存储结点之间的关系, 结点之间的关系是通过存储单元的相邻关系隐含地表示出来的。

#### 2) 链接存储方式

链接存储方式数据元素可以存储在任意的位置上, 不要求逻辑上相邻的结点在物理上也相邻。在这种存储方式下, 为了存储数据元素之间的逻辑关系, 就需要附加指针字段来存储各结点后继结点的地址。

#### 3) 索引存储方式

索引存储方式是在存储结点信息的同时, 还建立附加的索引表, 索引表中的每一项称为索引项, 索引项的一般形式为: [关键字, 地址], 其中关键字可惟一标识一个结点的数据项。

#### 4) 散列存储方式

散列存储方式是利用结点关键字的值直接计算出该结点存储单元地址, 然后将结点按某种方式存入该地址的一种方法。

### 1.1.4 算法及其描述

#### 1. 算法的概念

算法是对特定问题求解方法和步骤的一种描述，它是指令的一组有限序列，其中每个指令表示一个或多个操作。

算法具有下列五个重要特性：

##### 1) 有穷性

算法的有穷性是指算法必须能够在执行有限个步骤之后结束，并且每个步骤都必须有穷的时间内完成。

##### 2) 确定性

算法的确定性是指算法中的每一个步骤必须是有明确定义的，不允许有模棱两可的解释，也不允许有多义性。

##### 3) 可行性

算法的可行性又称为算法的能行性，它是指算法中描述的操作是可以通过已经实现的基本运算执行有限次来实现，即算法的具体实现应该能够被计算机执行。

##### 4) 输入

一个算法可以有零个或多个输入，这些输入取自某个特定的对象集合。

##### 5) 输出

一个算法必须有一个或多个输出，这些输出是同输入有着某些特定关系的量。

判断一个算法的好坏主要有正确性、可读性、健壮性和效率这几个标准。

#### 2. 算法的描述

算法的描述形式有多种，例如可以采用自然语言、数据流程图、伪代码、数学语言或约定的符号来描述，也可以采用高级程序设计语言来描述。

本书选用高级程序设计语言 C 语言作为算法描述的工具。为了描述的方便，使算法清晰简明，本书又不完全拘泥于 C 语言的细节，增加了抽象数据类型 (ADT)。

### 1.1.5 算法分析

算法分析是对一种算法所消耗的计算机资源的估算，其中包括计算机运行时间的长短和所占据空间的大小。

#### 1. 算法效率及其度量

使用绝对运行时间来衡量算法的优劣是不科学的，因为这样的统计依赖于计算机的软、硬件环境等多种综合因素，不单纯反映算法的优劣，容易掩盖算法自身的特性。

通常是在不实际运行该算法的前提下，估算当问题的规模扩大时，该算法的时间代价和空间代价。

下面是用渐进式法表达时间复杂度的定义：

设  $T(n)$  是问题规模  $n$  的时间函数，采用渐进式的“大 O 表示法”表示算法代价增长率的上界，设  $f(n)$  和  $T(n)$  是定义在正数集合上，当且仅当存在正数  $c$  和非负数  $N$ ，使得对所有的  $n \geq N$  时，满足：

$$0 \leq T(n) \leq cf(n)$$

则称  $O(f(n))=T(n)$ ，即  $f(n)$  是  $T(n)$  的渐近紧密上限。

上面的定义可以理解为：当  $n$  充分大时，随着  $n$  的增加，如果函数  $T(n)$  存在一个增长的上界  $cf(n)$ ，则算法的渐近时间复杂度为  $O(f(n))$ ，简称为时间复杂度。

算法的时间复杂度有多种表现形式，常见的有以下几种：

- (1) 常量时间  $O(1)$ 。
- (2) 线性时间  $O(n)$ 。
- (3) 平方时间  $O(n^2)$ 。
- (4) 立方时间  $O(n^3)$ 。
- (5) 对数时间  $O(\log n)$ 。
- (6) 指数时间  $O(a^n)$ 。

当  $n$  充分大时，它们的关系如下：

常量时间 < 对数时间 < 线性时间 < 平方时间 < 立方时间 < 指数时间

即： $O(1) < O(\log n) < O(n) < O(n^2) < O(n^3) < O(2^n)$ 。

## 2. 算法时间复杂度的计算

常常通过分析实现算法的高级语言程序来实现对算法时间复杂度的计算。结果经常以渐近式的大  $O$  形式来表示算法的时间复杂度。

渐近式的大  $O$  形式表示时间复杂度的主要运算规则如下：

- (1)  $O(f(n))+O(g(n))=O(\max(f(n),g(n)))$ 。
- (2)  $O(f(n))*O(g(n))=O(f(n)*g(n))$ 。
- (3)  $O(cf(n))=O(f(n))$ ，其中， $c$  是任一正的常数。

## 3. 空间复杂度

空间复杂度 (Space Complexity) 也是度量一个算法好坏的标准，它所描述的是算法在运行过程中所占存储空间的大小。设  $S(n)$  是问题规模  $n$  的函数，可定义算法的渐近空间复杂度为  $O(f(n))$ ，记为：

$$s(n)=O(f(n))$$

类似于时间复杂度，算法的空间复杂度也以数量级的形式给出，如  $O(1)$ 、 $O(n)$  等。

## 1.2 典型例题解析

**【例 1-1】** 根据下面求解  $n!$  的递归算法  $\text{fact}(n)$ ，求出该程序段的时间复杂度。

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

**解析：**

设  $\text{fact}(n)$  的运行时间函数为  $T(n)$ ，从以上算法可以看出，当  $n \leq 1$  时，只执行语句 (1)，它的时间复杂度为  $O(1)$ ；当  $n > 1$  时，执行语句 (2)，它的时间复杂度为  $O(1)+T(n-1)$ ，其中  $O(1)$  为执行判断所消耗的常量阶时间。由此建立的该递归算法的时间函数如下：

$$T(n)=\begin{cases} O(1) & n \leq 1 \\ T(n-1)+O(1) & n > 1 \end{cases}$$

这是一个递推公式，通过重复代入来递推计算  $T(n)$ 。

其递推计算过程如下：

$$\begin{aligned}
 T(n) &= O(1) + T(n-1) \\
 &= 2 * O(1) + T(n-2) \\
 &= 3 * O(1) + T(n-3) \\
 &\dots\dots \\
 &= (n-1) * O(1) + T(1) \\
 &= (n-1) * O(1) + O(1) \\
 &= n * O(1) \\
 &= O(n)
 \end{aligned}$$

从而求得算法的时间复杂度为  $O(n)$ 。

**【例 1-2】** 写出下面算法的时间复杂度。

```
(1) i=1;
(2) while ( i<=n )
    i=i*5;
```

**解析：**

语句 (1) 的频度为  $O(1)$ ，设语句 (2) 的频度为  $f(x)$ ，则有：

$$5^{f(x)} \leq n$$

即有  $f(x) \leq \log_5 n$ ，取  $f(x)$  的最大值得到该算法的时间复杂度为  $O(\log_5(n))$ 。

**【例 1-3】** 汉诺塔问题描述为：有 X、Y 和 Z 三个柱子， $n$  个大小不等且都能套进柱子的圆盘（编号为 1、2、 $\dots$ 、 $n$ ），这  $n$  个圆盘已经按照自上至下由小到大的顺序在其中的一根柱子 X 上，要求将这些圆盘，按照如下规则由 X 柱子移动到 Z 柱子。

- (1) 每次只能移动柱子最上面的一个圆盘。
- (2) 任何圆盘都不能放在比它小的圆盘之上。
- (3) 圆盘只能在 X、Y 和 Z 三根柱子之上放置。

写出相关算法并分析时间复杂度。

**解析：**

汉诺塔递归算法可以描述成如下形式：

```
void hanoi( n,x,y,z )
char x,y,z;
int n;
{
    if ( n==1 )
        printf("%c-->%c\n",x,z);
    else
    {
        hanoi( n-1,x,z,y );
        printf("%c-->%c\n",x,z);
        hanoi( n-1,y,x,z );
    }
}
/*hanoi*/
```

上述算法语句频度的关系式可以表达成如下形式：



$$F(n)=\begin{cases} O(1) & n=1 \\ F(n-1)+1 & n>1 \end{cases}$$

因此有如下递推过程:

$$\begin{aligned} F(n) &= 2F(n-1)+1 \\ &= 2(2(F(n-2)+1))+1 \\ &= 2^2 F(n-2)+3 \\ &= 2^2 (2F(n-3)+1)+2^2-1 \\ &= 2^3 F(n-3)+7 \\ &= 2^3 F(n-3)+2^3-1 \\ &= \dots \\ &= 2^{n-1} F(1)+2^{n-1}-1 \\ &= 2^n-1 \end{aligned}$$

因此汉诺塔问题算法的时间复杂度为  $O(2^n)$ 。

## 小结

本章主要介绍了数据结构的基本概念、基本术语、算法的描述和分析,包括:数据结构的发展概况、数据结构的基本概念和术语、数据的逻辑结构和存储结构的含义、算法的基本概念和重要特性、算法的分析方法、估算算法时间复杂度的方法和空间复杂度的含义等内容。

在典型例题解析中分析了几种较为复杂和典型的求算法时间复杂度的方法。读者要认真体会本章的一些基本概念、术语和分析算法时间复杂度的方法。

## 习题一

### 一、选择题

1. 数据结构是一门研究非数值计算的程序设计问题中的操作对象以及它们之间的( )和运算的学科。  
A. 结构            B. 关系            C. 运算            D. 算法
2. 在数据结构中,从逻辑上可以把数据结构分成( )。  
A. 动态结构和静态结构            B. 紧凑结构和非紧凑结构  
C. 线性结构和非线性结构            D. 逻辑结构和存储结构
3. 线性表的逻辑顺序和存储顺序总是一致的,这种说法( )。  
A. 正确            B. 不正确            C. 无法确定            D. 以上答案都不对
4. 算法分析的目的是( )。  
A. 找出算法的合理性            B. 研究算法的输入/输出关系  
C. 分析算法的有效性以求改进            D. 分析算法的易懂性

### 二、填空题

1. \_\_\_\_\_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