

<<数据结构导论>>

图书基本信息

书名：<<数据结构导论>>

13位ISBN编号：9787300124308

10位ISBN编号：7300124305

出版时间：2010-8

出版时间：中国人民大学出版社

作者：蔡厚新，肖守柏 著

页数：192

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<数据结构导论>>

前言

计算机科学技术以惊人的速度迅猛发展，它的应用范围已渗入到社会和生活的各个领域。相应地，数据处理的对象也从简单的数值发展到字符、表格和图形等带有结构的数据。在这里要解决的关键问题是：针对每一种新的应用领域的处理对象，如何选择合适的表示（结构），如何有效地组织数据、处理数据。

数据结构就是研究数据以及数据之间关系的一门学科，主要研究数据之间的逻辑结构及其基本操作在计算机中的表示和实现。

数据结构课程不仅是计算机专业重要的专业基础课，也是从事计算机软件开发所必备的专业知识。

本教材主要面向高职高专院校或应用性本科的计算机类专业的学生，培养技术应用性人才。

内容的构造力求体现“以应用为主体”，强调理论知识的理解和运用，实现教学以实践体系为主及以技术应用能力培养为主的培养目标。

案例教学是计算机语言教学最有效的方法之一，好的案例对学生理解知识、掌握如何应用知识都十分重要。

本书围绕教学内容组织案例，对学生的知识和能力训练具有很强的针对性。

全书共十二章，大体上可看成为由四个部分组成，基本的线性结构及有关的典型应用是第一部分（第二章到第六章）；具有广泛应用价值的树形结构在第七、八章讲述，这两部分占据了本书的主要篇幅；第九章及第十章介绍复杂数据结构，如图、稀疏矩阵及广义表等；有关外存储器中的数据结构和文件组织放在第四部分。

<<数据结构导论>>

内容概要

《数据结构导论》不仅是计算机专业重要的专业基础课，也是从事计算机软件开发必备的专业知识。

全书共十二章分为四部分，依次介绍了数据结构的基本概念，线性表、栈、串、队列和数组、树结构和图结构，以及查找和排序等基本运算。

每章节从实例入手，系统地介绍了各种常用的数据结构，注重实用性，由浅入深，图文并茂，易教易学。

《数据结构导论》内容丰富，概念讲解清楚，叙述严谨流畅，逻辑性强。

每章均配有小结和思考与练习。

《数据结构导论》可作为高等院校高职高专计算机专业教材和相关培训教材，也可作为从事计算机软件工作人员的参考用书。

<<数据结构导论>>

书籍目录

第1章 绪论1.1 数据结构1.2 实例：编写HELLO, WORLD!
程序1.3 实例：数组元素排序第2章 线性表2.1 实例：“银行排队”顺序存储2.2 实例：“学生健康登记表”链式存储2.3 其他链表第3章 栈和队列3.1 实例：回文3.2 实例：杨辉三角第4章 串4.1 串的基本概念4.2 实例：文本加密第5章 内部排序5.1 排序的基本概念5.2 实例：学生成绩插入排序5.3 实例：学生成绩交换排序5.4 实例：学生成绩选择排序5.5 其他排序第6章 查找6.1 实例：学生成绩不及格的查找6.2 实例：学生成绩及格的查找6.3 实例：学生成绩优秀的查找第7章 二叉树7.1 实例：高校篮球比赛7.2 实例：高校篮球总决赛7.3 实例：学生成绩及格的查找7.4 实例：报文第8章 树8.1 实例：高校教师讲课比赛（一）8.2 实例：高校教师讲课比赛（二）第9章 图9.1 实例：城际铁路9.2 实例：游园路线第10章 数组、矩阵和广义表10.1 实例：学生出勤的天数10.2 实例：学生出勤的放假天数10.3 实例：学生出勤的请假天数第11章 文件11.1 文件的基本概念11.2 顺序文件11.3 散列文件第12章 外部排序12.1 外部排序的基本思想12.2 外部排序的方法参考文献附：《数据结构导论学习指导》

章节摘录

对于一个问题可以有多种算法，如将在第5章介绍的排序有多达8种算法。

那么如何来衡量哪种算法最有效？

或者优于目前已知的算法呢？

人们一般从两个方面来衡量。

一个是时间效率，即算法处理数据时所花费的时间，用时间复杂度来表示；一个是空间效率，即算法所需求的存储量的大小，用空间复杂度来表示。

但二者往往有冲突，不能同时兼顾，一般取时间效率，时间效率被认为更重要一些。

1.时间复杂度分析 对于解决同一个问题的算法，执行时间短的显然比执行时间长的时间效率高，即执行时间短的算法比执行时间长的算法时间复杂度要低。

那么算法执行时间的长短如何度量呢？

一种方法是编制一个程序实现这个算法，然后输入不同的数据运行这个程序，测定该程序运行的时间被称为事后统计法。

这种方法的缺陷非常明显：一是必须编制程序和运行程序，非常耗费时间，也比较麻烦；二是受到的约束条件比较多，比如运行程序的计算机软硬件条件、使用的编程语言等，这些有时会掩盖算法本身的优劣。

另一种方法是分析算法运行的时间，称为事前分析法。

它不上机运行依算法编制的程序，而是分析影响算法执行时间的各种因素，从而估算出算法执行的时间。

其中，一个最重要的因素是输入算法的数据量（称为问题规模）。

例如，一个查找单词的算法，在100个单词中查找某个单词与在工。

万个单词中查找某个单词所花费的时间肯定是不同的。因此，一个算法的执行时间 T 可被表示为问题规模 n 的一个函数 $T(n)$ 。

除了问题规模以外，实现算法的程序设计语言、源程序编译后产生的机器代码的质量、机器执行指令的速度等都会影响算法的执行时间。

因此，不可能将 $T(n)$ 表达为算法实际执行的时间。

一般用算法中语句被执行的次数来表示算法的时间效率（算法的时间复杂度）。

可用下面的例子来说明。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>