

<<数据结构>>

图书基本信息

书名：<<数据结构>>

13位ISBN编号：9787302235989

10位ISBN编号：7302235988

出版时间：2011-1

出版时间：清华大学出版社

作者：邓文华 编

页数：225

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>>

前言

作为独立学院三本的学生有着其自身的特点，如果直接拿原来一本、二本的教材给独立学院的三本学生使用是很不恰当的，本书是在保留原第2版精华基础上，专门针对独立学院三本的学生特点，即理论够用，着重培养学生设计能力与动手能力而编写的。

“数据结构”是计算机程序设计的重要理论基础，是计算机及其应用专业的一门重要基础课程和核心课程。

它不仅是计算机软件专业课程的先导，而且也逐渐为其他工科类专业所重视。

本书共分9章，第1章叙述数据、数据结构和算法等基本概念。

第2~6章分别讨论了线性表、栈和队列、串和数组、树与二叉树，以及图等基本数据结构的实现及其应用。

第7、8章分别讨论了查找和排序操作的各种实现方法及其特点，第9章给出了6个综合应用的实例程序，以便学生上机实习，进一步加强对学生的实际应用能力及动手能力的培养。

本书具有以下特点：（1）基础理论知识的阐述由浅入深、通俗易懂。

内容组织和编排以应用为主线，略去了一些理论推导和数学证明的过程，淡化算法的设计分析和复杂的时空分析。

（2）各章都配有相应的典型例题或应用，列举分析了许多实用的例子，大多数算法都直接给出了其相应的C语言程序，以便学生上机练习、实践。

（3）为了便于学生复习及掌握每章的重点、要点，本书在每章的结束处对该章进行小结，并列本章的重点。

（4）本书配有电子教案及习题解答，方便教师的备课与教学。

本书讲课时数为60~90学时左右，上机时数在20~36学时左右。

教师可根据学时数、专业和学生的实际情况选讲应用与典型例题中的例子。

本书由邓文华任主编，邹华胜任副主编。

其中第1、2章由邹华胜编写，第3、4章由胡智文编写，第5、6章由毕保祥编写，第7~9章由邓文华编写，谢胜利也编写了第7、8章的部分内容。

全书由邓文华统稿。

编写独立学院三本的教材是一项新的尝试，书中难免存在疏漏，恳请读者批评指正。

<<数据结构>>

内容概要

本书对常用的数据结构作了系统的介绍，力求概念清晰，注重实际应用。

全书共分9章，依次介绍了数据结构的基本概念、线性表、栈和队列、串与数组、树与二叉树、图结构，以及查找和排序等基本运算，第9章给出了6个综合应用的实例程序。

全书用C语言作为算法描述语言，每一章均附有典型例题与小结，便于总结与提高。

本书叙述简洁、深入浅出，主要面向独立学院三本计算机专业的学生，也可以作为大学非计算机专业学生的选修课教材和计算机应用技术人员自学参考用书。

<<数据结构>>

书籍目录

第1章 绪论 1.1 从问题到程序 1.2 有关概念和术语 1.3 算法及算法分析 1.3.1 算法特性
1.3.2 算法描述 1.3.3 算法分析 1.4 关于数据结构的学习 1.5 关于本书的编写说明 本章
小结 习题1第2章 线性表 2.1 线性表的逻辑结构 2.1.1 线性表的定义 2.1.2 线性表的
基本操作 2.2 线性表的顺序存储及其操作的实现 2.2.1 顺序表 2.2.2 顺序表基本操作的
实现 2.2.3 顺序表的其他操作举例 2.3 线性表的链式存储及其操作的实现 2.3.1 单链表
2.3.2 单链表基本操作的实现 2.3.3 循环链表 2.3.4 双向链表 2.3.5 单链表的其他操
作举例 2.4 典型应用 2.4.1 一元多项式的存储表示 2.4.2 一元多项式的相加运算 本章小
结 习题2第3章 栈和队列 3.1 栈 3.1.1 栈的定义及其基本运算 3.1.2 栈的存储结构和基
本运算的实现 3.1.3 栈的应用举例 3.1.4 栈与递归的实现 3.2 队列 3.2.1 队列的定义
及其基本运算 3.2.2 队列的存储结构和基本运算的实现 3.2.3 队列的应用举例 3.3 典型例
题 本章小结 习题3第4章 串和数组第5章 树与二叉树第6章 图第7章 查找第8章 排序第9章
综合应用实例参考文献

<<数据结构>>

章节摘录

插图：(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。

算法的含义与程序十分相似，但又有区别。

一个程序不一定满足有穷性。

例如，操作系统，只要整个系统不遭破坏，它将永远不会停止，即使没有作业需要处理，它仍处于动态等待中。

因此，操作系统不是一个算法。

另一方面，程序中的指令必须是机器可执行的，而算法中的指令则无此限制。

算法代表了对问题的求解方法，而程序则是算法在计算机上的特定的实现。

一个算法若用程序设计语言来描述，则它就是一个程序。

算法与数据结构是相辅相成的。

解决某一类特定问题的算法可以选择不同的数据结构，而且选择恰当与否直接影响算法的效率；反之，一种数据结构的优劣由各种算法的执行效果来体现。

在设计算法时通常需要考虑以下几个方面的要求。

(1) 正确性：算法的执行结果应当满足预先规定的功能和性能要求。

正确性要求表明算法必须满足实际需求，达到解决实际问题的目标。

(2) 可读性：一个算法应当思路清晰、层次分明、简单明了、易读易懂。

可读性要求表明算法主要是人与人之间交流解题思路和进行软件设计的工具，因此可读性必须要强。

同时一个可读性强的算法，其程序的可维护性、可扩展性都要好许多，因此，许多时候人们往往在一定程度上以牺牲效率来提高可读性。

(3) 健壮性：当输入不合法数据时，应能作适当处理，不至于引起严重后果。

健壮性要求表明算法要全面细致地考虑所有可能的边界情况，并对这些边界条件作出完备的处理，尽可能使算法没有意外的情况。

(4) 高效性：有效使用存储空间和有较好的时间效率。

高效性主要是指时间效率，即解决相同规模的问题时间尽可能短。

一般来说，数据结构上的基本操作主要有以下几种。

(1) 查找：寻找满足特定条件的数据元素所在的位置。

(2) 读取：读出指定位置上数据元素的内容。

(3) 插入：在指定位置上添加新的数据元素。

(4) 删除：删去指定位置上对应的数据元素。

(5) 更新：修改某个数据元素的值。

<<数据结构>>

编辑推荐

《数据结构(C语言版)(第3版)》：案例驱动的教学模式，一线优秀教师担纲编写，立体化教学资源解决方案。

<<数据结构>>

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>