

<<数据结构>>

图书基本信息

书名：<<数据结构>>

13位ISBN编号：9787115185778

10位ISBN编号：7115185778

出版时间：2008-8

出版时间：人民邮电出版社

作者：王学军 编

页数：243

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

“数据结构”是计算机及相关专业的重要专业基础课程。随着计算机技术的不断发展，数据结构的内容也需要做适当调整，重点要提高学生的实践能力。本书就是以计算机及相关专业学生必须掌握的知识为核心，以高等职业教育所培养的学生应该具备的能力为依据，以突出实践性和实用性为目的，进行设计编写的。

本书编写人员都是长期从事高等职业教育计算机教学的一线教师，具有丰富的教学 and 实践经验，同时编写组中还有在企业从事过Java软件开发的工程人员。

本书的主要特色如下。

(1)内容选取合理，组织得当。

本书根据高等职业教育计算机人才培养目标组织内容，理论部分以够用为度，重点突出实践性和实用性。

每章都由实例引入，并且配备一定数量的扩展实例(其中包括一部分工程实例)，有助于对理论知识的消化、理解。

(2)算法实现方式先进，适合学生学习。

本书采用Java语言作为数据结构算法的实现语言，体现面向对象语言的特色，使数据结构的相关算法实现更加方便，教学效果更加突出。

(3)定位准确、恰当，适合高职高专院校学生使用。

本书主要面向高职高专院校学生，目的是把学生培养成高等技术应用型生产一线人才。

(4)突出适合高等职业教育的任务驱动方法。

本书所有内容都以学习任务的方式给出，帮助教师把握在教学中应该注意的重点、难点。

本书建议学时为70~90学时，其中理论学时为50~60学时，实践环节学时为20~30学时。授课时应以任务驱动为主，理论结合实际，突出实例在本课程教学中的作用。

<<数据结构>>

内容概要

本书共分10章，重点介绍3种基本数据结构及其应用，主要内容包括绪论、Java语言基础知识、线性表、栈和队列、数组和广义表、串、树与二叉树、图、查找和排序等。

本书采用Java语言描述数据结构中的算法，每章配有一定数量的具有完整程序的实例，并在最后提供难易适中、与所讲理论知识相配套的习题，帮助读者学习和理解理论知识。

本书面向高等职业院校学生，语言通俗易懂，每章都由实例引入，理论和实践紧密结合。

全书重点突出基本理论和基本算法的实现过程，强调实践性和实用性。

另外本书配有电子教案和习题解答，可从人民邮电出版社的网站（www.ptpress.com.cn）下载。

本书可作为高职高专院校计算机及相关专业“数据结构”课程的教材，也可作为各类计算机培训班的教材。

<<数据结构>>

书籍目录

第1章 绪论	1.1 数据结构的3种基本结构	1.1.1 线性结构	1.1.2 层次结构	1.1.3 网状结构
1.2 数据结构研究的主要问题	1.3 算法及描述	1.3.1 算法与算法特性	1.3.2 算法表示	1.4 算法效率分析
习题	第2章 Java语言基础知识	2.1 实例引入	2.2 Java语言概述	2.3 面向对象程序设计简述
2.3 面向对象程序设计的基本概念	2.3.1 面向对象程序设计的基本概念	2.3.2 面向对象程序设计的	2.4 Java语言基础知识	2.4.1 数据类型
2.4.2 运算符	2.4.3 流程控制	2.4.4 数组	2.4.5 类与对象	2.4.6 类的封装性
2.4.7 类的继承性	2.4.8 类的多态性	2.4.9 抽象类和内部类	2.4.10 接口	2.4.11 包
2.4.12 异常处理	2.4.13 Java标准数	据流	2.5 Java语言中的“指针”实现	2.6 JDK1.5新增特性
2.6.1 泛型	2.6.2 增强的集合	遍历结构	2.6.3 自动装箱/拆箱	2.6.4 枚举类型
2.6.5 静态import	2.6.6 从终端读取	数据	2.6.7 格式化输出	2.6.8 可变参数
习题	第3章 线性表	3.1 实例引入	3.2 线性表的概述	3.2.1 线性表的概念
3.2.2 线性表的存储结构及操作	3.3 顺序表的基本操作及实现	3.3.1 顺序表的概述	3.3.2 顺序表的基本操作及实现	3.4 链表的基本操作及实现
3.4.1 链表的分类	3.4.2 链表的分类	3.4.3 单链表的基本运算及实现	3.4.4 其他形式的链表的相关运算	3.4.5 算法实例
3.5 线性表的应用	3.5.1 顺序表的连接	3.5.2 字符串的逆转算法	习题	第4章 栈和队列
4.1 实例引入	4.2 栈的相关概述	4.2.1 栈的定义	4.2.2 栈的相关概念	4.2.3 栈的操作过程
4.2.4 栈的存储结构	4.3 用数组实现顺序栈及操作	4.4 用类实现链式栈及相应操作	4.5 队列的相关概述	4.5.1 队列的定义
4.5.2 队列的相关概念	4.5.3 队列的存储结构	4.6 用数组实现顺序队列及相应操作	4.7 用类实现链队列及相应操作	4.8 栈和队列的实例应用
习题	第5章 数组和广义表	5.1 实例引入	5.2 数组	5.2.1 数组的基本概念
5.2.2 一维数组	5.2.3 二维数组	5.3 特殊矩阵	5.3.1 对称矩阵	5.3.2 三角矩阵
5.3.3 对角矩阵	5.4 稀疏矩阵	5.5 广义表	5.5.1 广义表的概念	5.5.2 广义表的存储结构
习题	第6章 串	6.1 实例引入	6.2 串的概念	6.3 串的顺序存储结构
6.3.1 通过String类处理串	6.3.2 通过StringBuffer类处理串	6.4 串的链式存储结构	6.4.1 链串的实现	6.4.2 链串基本算法
习题	第7章 树与二叉树	7.1 实例引入	7.2 树	7.2.1 树的定义
7.2.2 树的表示方法	7.2.3 树的抽象数据类型	7.2.4 树的存储结构	7.3 二叉树	7.3.1 二叉树的定义
7.3.2 二叉树的性质	7.3.3 二叉树的抽象数据类型	7.3.4 二叉树的存储结构	7.4 二叉树的节点类及二叉树类	7.4.1 二叉树节点类
7.4.2 二叉树类	7.5 二叉树的遍历	7.5.1 二叉树遍历算法	7.5.2 二叉树遍历算法的实现	7.5.3 非递归的二叉树遍历算法
7.5.4 二叉树遍历的应用	7.6 线索二叉树	7.6.1 线索二叉树的定义	7.6.2 线索二叉树的存储结构	7.6.3 遍历线索二叉树
7.6.4 构造中序线索二叉树	7.7 树和森林	7.7.1 树、森林与二叉树的转换	7.7.2 树和森林的遍历	7.8 树的应用
7.8.1 二叉排序树	7.8.2 哈夫曼树和哈夫曼编码	7.8.3 判定树	习题	第8章 图
8.1 实例引入	8.2 图的基本概念	8.2.1 图的定义	8.2.2 图的相关概念	8.3 图的存储结构
8.3.1 邻接矩阵	8.3.2 邻接表	8.4 图的遍历	8.4.1 深度优先搜索遍历	8.4.2 广度优先搜索遍历
8.5 生成树和最小生成树	8.5.1 生成树	8.5.2 Kruskal算法	8.5.3 Prim算法	8.6 最短路径问题
8.7 拓扑排序	8.7.1 有向无环图	8.7.2 拓扑排序	8.8 AOE网与关键路径	8.8.1 AOE网
8.8.2 关键路径	8.9 综合示例	习题	第9章 查找	9.1 实例引入
9.2 基本概念与术语	9.2.1 查找的概念	9.2.2 查找方法	9.3 顺序查找法	9.4 折半查找法
9.5 二叉排序树法	9.6 哈希查找法	9.6.1 哈希查找概念	9.6.2 哈希函数	9.6.3 冲突解决方法
9.7 应用实例	习题	第10章 排序	10.1 实例引入	10.2 排序的概念
10.3 排序的分类	10.3.1 按照存储交换分类	10.3.2 按照内部排序的过程	10.3.3 按照排序的稳定性分类	10.4 插入排序
10.4.1 直接插入排序	10.4.2 希尔排序	10.5 交换排序	10.5.1 冒泡排序	10.5.2 快速排序
10.6 选择排序	10.6.1 直接选择排序	10.6.2 堆排序	10.7 其他排序	10.7.1 归并排序
10.7.2 基数排序	10.8 排序的工程应用举例	习题	参考文献	

章节摘录

第1章 绪论 1.4 算法效率分析 【学习任务】了解算法的实效性分析方法，重点了解时间复杂度的计算方法以及近似表示方法，并掌握通过时间复杂度判断算法优劣的方法。

通过前面的分析，每个算法都可以用多种方式实现，但是实现算法的效率是不一定相同的。每个程序执行所花费的时间，称为该程序的时间复杂度，如果忽略语句间的执行时间差别，一般用该程序每条语句的执行次数作为该程序的时间复杂度进行判断。

对时间复杂度的判断，是断定某程序（或算法）效率是否高的标准之一。

一个算法的时间复杂度（Time Complexity）是指在计算机上运行该算法（或程序）所需要的时间。实际上，每个程序员都知道，算法执行的时间和很多因素都有关系，例如，机器的性能、算法语言的选取、编译程序的效率、算法的选择，以及问题本身的因素（如问题的复杂程度、问题本身的规模等）。

在针对实际问题时，尤其是考虑规模比较大的问题时，一般使用渐进式表示法来判别算法的时间复杂度。

为了使程序员更好地掌握算法本身的特性，通常的做法是：在不考虑不确定情况的前提下，以算法中简单操作重复执行的次数作为算法的时间复杂度的衡量标准，即主要考虑问题的规模，而不考虑某些单个步骤之间的时间差异。

因此，一个特定算法的运行时间长短更多地依赖于问题的规模 n ，或者说它是问题规模 n 的函数 $f(n)$ ，因此，引入渐进时间复杂度在数量上估计一个算法的执行时间，也能够达到分析算法的目的。

算法时间的度量记做 $T(n) = O(f(n))$

<<数据结构>>

编辑推荐

教材编写思路：本书重点培养高职高专院校计算机及相关专业学生应具备的数据结构应用能力。理论知识部分以够用为主，强调理论和实践紧密结合，重点突出实践和实用性。每章均由实例引入，体现了“以就业为导向”的职业教育理念；采用“以应用实例巩固理论知识”的内容编排结构，突出了“以技能培养为目标”的职业教育思想。

适用教学对象：适合作为高职高专院校计算机及相关专业“数据结构”课程的教材。

辅助教学资源：教学课件，模拟试卷，习题答案。

<<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>