

<<数据结构教程>>

图书基本信息

书名：<<数据结构教程>>

13位ISBN编号：9787302305170

10位ISBN编号：730230517X

出版时间：2013-2

出版时间：清华大学出版社

作者：李春葆 编

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>

<<数据结构教程>>

内容概要

《高等学校数据结构课程系列教材:数据结构教程(C#语言描述)》系统地介绍了常用的数据结构以及排序、查找的各种算法,阐述了各种数据结构的逻辑关系、存储表示及运算操作,并采用C#语言描述数据组织和算法实现。

《高等学校数据结构课程系列教材:数据结构教程(C#语言描述)》既注重原理,又注重实践,配有大量图表、实践教学项目和习题,内容丰富,概念讲解清楚,表达严谨,逻辑性强,语言精练,可读性好。

作者简介

李春葆，武汉大学计算机学院教授，主要研究方向为数据挖掘和算法设计，先后主持和参加多个大型研究项目。

主要为本科生讲授数据结构（15年以上）和软件工程等课程，为研究生讲授软件开发新技术、数据仓库与数据挖掘等课程，并出版十多部精品著作。

<<数据结构教程>>

书籍目录

第1章绪论 1.1什么是数据结构 1.1.1数据结构的定义 1.1.2数据的逻辑结构 1.1.3数据的存储结构 1.1.4数据的运算 1.1.5数据结构和数据类型 1.2算法及其描述 1.2.1什么是算法 1.2.2算法描述 1.3算法分析 1.3.1算法的特性和算法设计的目标 1.3.2算法时间效率分析 1.3.3算法存储空间分析 1.4数据结构的目標 本章小结 练习题1 第2章线性表 2.1线性表的定义 2.1.1什么是线性表 2.1.2线性表的抽象数据类型描述 2.2线性表的顺序存储结构 2.2.1线性表的顺序存储结构——顺序表 2.2.2顺序表基本运算的实现 2.3线性表的链式存储结构 2.3.1线性表的链式存储结构——链表 2.3.2单链表 2.3.3双链表 2.3.4循环链表 2.4线性表的应用 本章小结 练习题2 第3章栈和队列 3.1栈 3.1.1栈的定义 3.1.2栈的顺序存储结构及其基本运算的实现 3.1.3栈的链式存储结构及其基本运算的实现 3.1.4栈的应用 3.2队列 3.2.1队列的定义 3.2.2队列的顺序存储结构及其基本运算的实现 3.2.3队列的链式存储结构及其基本运算的实现 3.2.4队列的应用 本章小结 练习题3 第4章串 4.1串的基本概念 4.1.1什么是串 4.1.2串的抽象数据类型 4.2串的存储结构 4.2.1串的顺序存储结构——顺序串 4.2.2串的链式存储结构——链串 4.3串的模式匹配 4.3.1 Brute—Force算法 4.3.2 KMP算法 本章小结 练习题4 第5章数组和广义表 5.1数组 5.1.1数组的定义 5.1.2数组的存储结构 5.1.3特殊矩阵的压缩存储 5.2稀疏矩阵 5.2.1稀疏矩阵的三元组表示 5.2.2稀疏矩阵的十字链表表示 5.3递归 5.3.1递归的定义 5.3.2何时使用递归 5.3.3递归模型 5.3.4递归算法设计的步骤 5.3.5递归算法转换为非递归算法 5.4广义表 5.4.1广义表的定义 5.4.2广义表的存储结构 5.4.3广义表的运算 本章小结 练习题5 第6章树和二叉树 6.1树 6.1.1树的定义 6.1.2树的逻辑结构表示方法 6.1.3树的基本术语 6.1.4树的性质 6.1.5树的基本运算 6.1.6树的存储结构— 6.2二叉树 6.2.1二叉树的定义 6.2.2二叉树的性质 6.2.3二叉树与树、森林之间的转换 6.2.4二叉树的存储结构 6.2.5二叉树的基本运算及其实现 6.2.6二叉树的遍历 6.2.7二叉树的构造 6.2.8线索二叉树 6.3哈夫曼树 6.3.1哈夫曼树的定义 6.3.2哈夫曼树的构造算法 6.3.3哈夫曼编码 本章小结 练习题6 第7章图 7.1图的基本概念 7.1.1图的定义 7.1.2图的基本术语 7.2图的存储结构和基本运算的实现 7.2.1邻接矩阵存储方法 7.2.2邻接表存储方法 7.3图的遍历 7.3.1图的遍历的概念 7.3.2深度优先遍历 7.3.3广度优先遍历 7.3.4非连通图的遍历 7.3.5图遍历算法的应用 7.4图的应用 7.4.1生成树和最小生成树 7.4.2最短路径 7.4.3拓扑排序 7.4.4 AOE网与关键路径 本章小结 练习题7 第8章查找 8.1查找的基本概念 8.2线性表的查找 8.2.1顺序查找 8.2.2折半查找 8.2.3索引存储结构和分块查找 8.3树表的查找 8.3.1二叉排序树 8.3.2平衡二叉树 8.3.3 B—树 8.3.4 B+树 8.4哈希表查找 8.4.1哈希表的基本概念 8.4.2哈希函数构造方法 8.4.3哈希冲突的解决方法 8.4.4哈希表查找及性能分析 本章小结 第9章内排序 第10章外排序 附录A部分练习题参考答案 参考文献

章节摘录

版权页：插图：2.链式存储结构 链式存储结构不要求逻辑上相邻的结点在物理位置上也相邻，结点间的逻辑关系是由附加的指针字段表示的。

由此得到的存储表示称为链式存储结构，通常要借助于计算机程序设计语言（如C、C++、C#等）的指针来描述。

链式存储方法的主要优点是便于修改，在进行插入、删除运算时，仅需修改相应结点的指针域，不必移动结点。

但与顺序存储方法相比，链式存储方法的主要缺点是存储空间的利用率较低，因为分配给数据的存储单元有一部分被用来存储结点之间的逻辑关系了。

另外，由于逻辑上相邻的结点在存储空间中不一定相邻，所以不能对结点进行随机存取。

3.索引存储结构 索引存储结构通常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。

索引项的一般形式是：（关键字，地址），关键字唯一标识一个结点，索引表按关键字有序排序，地址作为指向结点的指针。

这种带有索引表的存储结构可以大大提高数据查找的速度。

线性结构采用索引存储方法后，可以对结点进行随机访问。

在进行插入、删除运算时，只需移动存储在索引表中对应结点的存储地址，而不必移动存放在结点表中结点的数据，所以仍保持较高的数据修改运算效率。

索引存储方法的缺点是增加了索引表，降低了存储空间的利用率。

4.哈希（或散列）存储结构 哈希存储结构的基本思想是：根据结点的关键字，通过哈希（或散列）函数直接计算出一个值，并将这个值作为该结点的存储地址。

哈希存储方法的优点是：查找速度快，只要给出待查结点的关键字，就可立即计算出该结点的存储地址。

与前3种存储方法不同的是，哈希存储方法只存储结点的数据，不存储结点之间的逻辑关系。

哈希存储方法一般只适合要求对数据能够进行快速查找和插入的场合。

上述4种基本的存储结构既可以单独使用，也可以组合使用。

同一种逻辑结构采用不同的存储方法，可以得到不同的存储结构。

选择何种存储结构来表示相应的逻辑结构，视具体要求而定，主要考虑的是运算方便及算法的时空要求。

1.1.4数据的运算 将数据存放在计算机中的目的是为了实现在一种或多种运算。

运算包括功能描述和功能实现，前者是基于逻辑结构的，是用户定义的，是抽象的；后者是基于存储结构的，是程序员用计算机语言或伪码表示的，是详细的过程。

例如，对于高等数学成绩单这种数据结构，可以进行一系列的运算，如增加一个学生成绩记录、删除一个学生成绩记录、求所有学生的平均分，查找序号为*i*的学生姓名和分数等。

编辑推荐

《高等学校数据结构课程系列教材:数据结构教程(C#语言描述)》既便于教师课堂讲授,又便于自学者阅读,可作为高等院校计算机相关专业本科生、专科生的教材,也可供广大从事计算机应用的科技人员参考。

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>