<<数据结构>>

图书基本信息

书名：<<数据结构>>

13位ISBN编号：9787040092035

10位ISBN编号：7040092034

出版时间：2001-1

出版时间：高等教育

作者：张乃孝//裘宗燕

页数：392

<<数据结构>>

前言

本书是一本系统介绍数据结构的教材。
在本书的论述过程中会反复联系到计算机领域的许多重要问题，包括问题求解，算法的分析与设计、抽象数据类型、程序设计方法、程序设计语言等内容，根据目前计算机科学发展的情况，本书采用面向对象的思想组织与课程有关的内容，运用C++语言作为数据结构和算法的描述语言，其目的是为了保证本书的先进性和适用性。

数据结构的一般讨论并不依赖于具体的语言，然而，在具体问题求解时，使用什么语言描述数据结构会在很大程度上影响程序设计者的思维方式和方法，选择一种实用的程序语言作为工作语言，有利于使讨论更面向实际应用，使参加课程学习的学生不仅可以掌握数据结构的抽象概念和性质，也有助于了解实现的思想和方法。

由于程序设计语言的发展，适合用来讨论数据结构的语言越来越多，今天国际上新的数据结构教材已经不再用伪语言作为工具了，随着面向对象方法的广泛应用，许多学校已经把C++语言定为学习计算机的第一种程序设计语言，越来越多的教材采用C++作为数据结构的教学语言。

在本书中，不仅讨论了抽象的数据结构，而且讨论了各种数据结构的不同实现方式并对它们进行了比较，从中可以看到不同数据结构的相互关系，包括结构上的（逻辑）关系和实现上的（表示）关系。
采用C++语言和面向对象的方法讲解数据结构，对这些关系的讨论提供了有力的工具。

由于C++语言的一些优点，特别是它的模板机制，使我们可以比较清晰地讨论具有类型参数的数据结构。
例如，我们实现的不是整数的堆栈，也不是字符串的堆栈，而是一般的具有类型参数的堆栈。
一个实现可以满足各种程序对于堆栈的需要。
这样学习的数据结构不仅更接近那种抽象概念上的数据结构，同时也可以直接用到不同类型对象的相同计算过程中去。

本书的讨论力图反映计算机科学的最新发展，用面向对象的方法组织数据结构的主要内容，以抽象数据类型的方式定义各种结构的用户界面，以时间和空间开销作为评价各种结构使用好坏的主要标准，从简到繁，系统地介绍各种常用的数据结构，强调不同结构之间的联系。
书中采用C++语言描述各种类的界面和实现。
把面向对象的程序设计方法与数据结构的主要原理紧密结合起来，讲解过程中还插入大量实例。
读者通过本书的学习不仅能掌握数据结构的基本原理和基本方法，同时可以把所学的知识用于实际问题求解的过程之中。

<<数据结构>>

内容概要

《数据结构：C++与面向对象的途径（修订版）》是1998年6月出版的《数据结构——C++与面向对象的途径》一书的修订版.它采用面向对象的思想组织数据结构的内容，运用C什语言作为讨论数据结构的工作语言。
在第一版的基础上，除对各章的顺序及内容安排进行了进一步的调整之外，还补充了各章的例子、习题，并增加了若干上机实习题，使读者可以更好地对数据结构进行学习、实践.在《数据结构：C++与面向对象的途径（修订版）》的最后还附加了一个上机实习报告的例子，使其具有较强的实用性。
《数据结构：C++与面向对象的途径（修订版）》除延续了第一版的风格外，内容更加充实、完整，，讲解更加清楚、透彻。
可作为本科计算机专业或相关专业数据结构课程教材，也可作为面向对象程序设计课程或C++程序设计实践课程的教材和参考书。

<<数据结构>>

书籍目录

章节摘录

另一种常用的算法设计方法称为“分治法”，其基本思想是：把一个规模较大的问题分成两个或多个较小的与原问题相似的子问题。
首先对子问题进行求解，然后设法把子问题的解合并起来，得出整个问题的解，即对问题分而治之。
如果一个子问题的规模仍然比较大，不能很容易地求得解，就可以对这个子问题重复地应用分治策略。

“二分法检索”就是用分治策略的典型例子。
如果要在一个整数的有序数组（可以假设元素按增序排列）中找一个数，要求查出这个数在这个数组里的位置，二分法检索采用的方法是先找到数组中居于中间位置的元素，将该元素与所找数字进行比较，如果所查数字比较小，那么它一定不会出现在中间元素的右边，下面只需在左半个数组中检索。
反过来，若所查数字较大，则不必在左半个数组中继续寻找，只需在右半个数组中检索。
这样，通过一次比较就能够排除掉一半元素。
再做下去，只要对剩下的半个数组重复使用同样方式，又可以再排除掉四分之一的元素，如此重复若干次就能很快确定整数的存在与否，并确定如果存在时它在什么位置。

还有一类常用算法被称为“回溯法”。
有一些问题，只能通过彻底搜索所有可能情况寻找一个满足某些预定条件的最优解。
由于彻底搜索的运算量通常非常大，往往大到使用计算机也不能在合理时间内得到解。
回溯算法是处理这类问题的一个方法，基本思想是一步一步向前试探，当有多种选择时可以任意选择一种，只要可行（或暂时没有失败）就继续向前，一旦发现问题或失败就后退，回到上一步重新选择，借助于回溯技巧和中间判断，常常可以大大地减少搜索时间。
常见的迷宫问题以及八皇后问题都可以用回溯方法来解决。

除上面提到的几种典型方法外，常用的算法设计方法还有“动态规划”、“局部搜索”和“分支限界”等。

本书在讨论数据结构的同时，自然涉及与数据结构相关的许多算法，掌握这些算法的设计方法和共性，对于提高程序设计的水平是十分重要的。

版权说明