

前言

本书目标

本书新的 Java 版论述数据结构——组织大量数据的方法，以及算法分析——算法运行时间的估计。随着计算机的速度越来越快，对于能够处理大量输入数据的程序的需求变得日益迫切。可是，由于在输入量很大的时候程序的低效率变得非常明显，因此这又要求对效率问题给予更仔细的关注。通过在实际编程之前对算法的分析，学生可以确定一个特定的解法是否可行。例如，在本书中学生可查阅一些特定的问题并看到巧妙的实现是如何能够把处理大量数据的时间限制从 16 年减至不到 1 秒的。因此，若无运行时间的阐释，就不会有算法和数据结构的提出。在某些情况下，对于影响实现的运行时间的一些微小细节都需要认真探究。

一旦确定了解法，接着就要编写程序。随着计算机功能的日益强大，它们必须解决的问题也变得更加庞大和复杂，这就要求我们开发更加复杂的程序。本书的目的是在教授学生良好的程序设计技巧和算法分析能力的同时，使得他们也能够开发出这种极为有效的程序。

本书适用于高级数据结构(CS7)课程或是第一年研究生的算法分析课程。学生应该具有中等程度的程序设计知识，包括面向对象程序设计和递归这样一些内容，此外，还要具有离散数学的一些知识。

处理方法

虽然本书的内容大部分都与语言无关，但是，程序设计还是需要使用某种特定的语言。正如书名指出的，我们为本书选择了 Java。

Java 是相对较新的语言，它常常用来和 C++ 进行比较。Java 具有许多的优点，编程人员常常把 Java 看成是一种比 C++ 更安全、更具有可移植性并且更容易使用的语言。因此，这使得它成为讨论和实现基础数据结构的一种优秀的核心语言。Java 的其他方面，诸如线程和 GUI(图形用户界面)，虽然很重要，但是本书并不需要，因此也就不再讨论。

使用 Java 和 C++ 对数据结构进行的完善描述均在互联网上提供了现成的材料。我们采用类似的编码约定以使得这两种语言之间的对等性更加明显。

本版中最显著的变化

本版(第 2 版)消除了一些程序中的错误，并对书中的许多部分进行了修订，以使阐述更加清晰。此外：

- 为了体现 Java 5.0 现代化的特色，我们对书中的程序代码进行了必要的更新。
- 对第 3 章进行了全面修改，包括对标准 ArrayList 类和 LinkedList 类(以及它们的迭代器)用法的讨论，以及对标准 ArrayList 类和 LinkedList 类的实现的讨论。
- 第 4 章也得到了修订，包括对 TreeSet 类和 TreeMap 类的讨论，同时用一个宽泛的实例阐述了它们在有效算法设计中的使用。第 9 章还包括一个例子，利用标准 TreeMap 类来实现最短路径算法。