**算法导论**

第一部分 基础知识

引言

这一部分将引导读者开始思考算法的设计和分析问题，简单介绍算法的表达方法、将在本书中用到的一些设计策略，以及算法分析中用到的许多基本思想。本书后面的内容基本都是建立在这些基础知识上的。

第一章是对算法及其在现代计算系统中地位的一个综述。本章给出了算法的定义和一些算法的例子。它还说明了算法是一项技术，就像快速的硬件、图形用户界面、面向对象系统和网络一样。

在第二章中，我们给出了书中的，第一批算法，他们解决的是对n个数进行排序的问题。这些算法是用一种伪代码形式给出的，这种伪代码尽管不能直接翻译为任何常规的程序设计语言，但足够清晰地表达了算法的结构，以便任何一位能力比较强的程序员都能用自己选择的某种语言将算法实现出来。我们分析的排序算法是插入排序，它采用了一种增量式的做法；另外还分析了合并排序算法，它采用了一种递归技术，称为“分治法”。尽管这两种算法所需的运行时间都随n的值而增长，但增长的速度是不同的，我们在第二章种分析了这两种算法的运行时间，并给出了一种有用的表示方法来表达这些运行时间。

第三章给出了这种表示法 的准确定义，称为渐进表示。在第三章的一开始，首先定义了集中渐进记号，它们主要用于表示算法运行时间的上界或下界，第三章余下的部分主要给出了一些数学表示方法。这一部分的作用更多的是为了确保读者所用的记号能与本书中的记号体系相匹配，而不主要是教授新的数学概念。

第四章更深入地讨论了第二章给引入的分治方法。特别地第四章包含了解决递归式的方法。递归式主要用于描述递归算法的运行时间。“主方法”是一种功能很强的技术，它可以用于解决分治算法中出现的递归式。第四章中的相当一部分内容都是在证明主办法的正确性，如果跳过这一部分证明内容的话，也没有什么太大的影响。

第五章介绍了概率分析和随机化算法。概率分析一般用于确定一些算法的运行时间，在这些算法中，由于同于规模的不同输入可能有着内在的概率分布，因而在这些不同输入之下，算法的运行时间可能有所不同。在有些情况下，我们假定算法的输入符合某种已知的概率分布，于是，算法的运行时间就是在所有可能的输入之下，运行时间的平均值，在其他情况下，概率分布不是来自于输入，而是来自于算法执行过程中所做出的随机选择。如果一个算法的行为不仅由其输入决定，还要由一个随机数生成器所生成的值来决定的话，他就是一个随机化算法。我们可以利用随机化算法，强行使算法的输入符合某种概率分布，从而确保不会有某一输入会始终导致算法的性能变坏；或者，对于那些允许产生不正确结果的算法，甚至能够将其错误率限制在某个范围之内。

附录A~附录C包含了另一些数学知识，它们对读者阅读本书可能会有所帮助。在阅读本书之前，读者很可能已经知道了附录中给出的大部分知识（我们采用的某些符号约定与读者过去见过的可能会有所不同）。因而，可以将附录视为参考材料。另一方面，你很可能从未见过第一部分中给出的内容。第一部分中的所有各章和附录都是以一种入门指南的风格来编写的。

1. 算法在计算中的作用

什么是算法？为什么要对算法进行研究?相对于计算机中使用的其他技术来说，算法的作用是什么？本章中，我们就要来回答这些问题。

* 1. 算法

简单来说，所谓算法（alogorithm）就是定义良好的计算过程，它取一个或一组值作为输入，并产生出一个或一组值作为输出。亦即，算法就是一系列的计算步骤，用来将输入数据转换成输出结果。

我们还可以将算法看作是一种工具，用来解决一个具有良好规格说明的计算问题。有关该问题的表述可以用通用的语言，来规定所需的输入/输出关系。与之对应的算法则描述了一个特定的计算过程，用于实现这一输入/输出关系。

例如，假设需要将一系列数按非降顺序进行排序。再实践中，这一问题经常出现。它为我们引入许多标准的算法设计技术和分析工具提供了吩咐的问题场景。下面是有关该排序问题的形式化定义：

输入：由n个数构成的一个序列（a1，a2，a3，…,an）。

输出：对输入序列的一个排列（重排），使得a1<a2<a3<…<an.

例如，给定一个输入序列