

算法在计算中的作用

什么是算法？为什么算法值得研究？相对于计算机中使用的其他技术来说算法的作用是什么？本章我们将回答这些问题。

1.1 算法

非形式地说，算法(algorithm)就是任何良定义的计算过程，该过程取某个值或值的集合作为输入并产生某个值或值的集合作为输出。这样算法就是把输入转换成输出的计算步骤的一个序列。

我们也可以把算法看成是用于求解良说明的计算问题的工具。一般来说，问题陈述说明了期望的输入/输出关系。算法则描述一个特定的计算过程来实现该输入/输出关系。

例如，我们可能需要把一个数列排成非递减序。实际上，这个问题经常出现，并且为引入许多标准的设计技术和分析工具提供了足够的理由。下面是我们

关于排序问题的形式定义。

输入： n 个数的一个序列 $\langle a_1, a_2, \dots, a_n \rangle$ 。

输出：输入序列的一个排列 $\langle a'_1, a'_2, \dots, a'_n \rangle$ ，满足 $a'_1 \leq a'_2 \leq \dots \leq a'_n$ 。

例如，给定输入序列 $\langle 31, 41, 59, 26, 41, 58 \rangle$ ，排序算法将返回序列 $\langle 26, 31, 41, 41, 58, 59 \rangle$ 作为输出。这样的输入序列称为排序问题的一个实例(instance)。一般来说，问题实例由计算该问题解所必需的(满足问题陈述中强加的各种约束的)输入组成。

因为许多程序使用排序作为一个中间步，所以排序是计算机科学中的一个基本操作。因此，已有许多好的排序算法供我们任意使用。对于给定应用，哪个算法最好依赖于以下因素：将被排序的项数、这些项已被稍微排序的程度、关于项值的可能限制、计算机的体系结构，以及将使用的存储设备的种类(主存、磁盘或者磁带)。

若对每个输入实例算法都以正确的输出停机，则称该算法是正确的，并称正确的算法解决了给定的计算问题。不正确的算法对某些输入实例可能根本不停机，也可能以不正确的回答停机。与人们期望的相反，不正确的算法只要其错误率可控有时可能是有用的。

在第 31 章，当我们研究求大素数算法时，将看到一个具有可控错误率的算法例子。但是通常我们只关心正确的算法。

算法可以用英语说明，也可以说明成计算机程序，甚至说明成硬件设计。唯一的要求是这个说明必须精确描述所要遵循的计算过程。

算法解决哪种问题

排序绝不是已开发算法的唯一计算问题(当看到本书的厚度时，你可能觉得算法也同样多)。算法的实际应用无处不在，包括以下例子：

- 人类基因工程已经取得重大进展，其目标是识别人类 DNA 中的所有 10 万个基因，确定构成人类 DNA 的 30 亿个化学基对的序列，在数据库中存储这类信息并为数据分析开发工具。这些工作都需要复杂的算法。虽然对涉及的各种问题的求解超出了本书的范围，但是求解这些生物问题的许多方法采用了本书多章内容的思想，从而使得科学家能够有效地使用资源以完成任务。因为可以从实验技术中提取更多的信息，所以既能节省人和机器的时间又能节省金钱。

- 互联网使得全世界的人都能快速地访问与检索大量信息。借助于一些聪明的算法，互联网上的网站能够管理和处理这些海量数据。必须使用算法的问题示例包括为数据传输寻找好的路由(求解这些问题的技术在第 24 章给出)，使用一个搜索引擎来快速地找到特定