

1

Introduction

In this chapter we shall study:

- several simple algorithms, set in a historical context (Section 1.1)
- the relationship between algorithms and programs (Section 1.2)
- static and dynamic data structures, and the concept of an abstract data type (Section 1.3).

1.1 Historical background

Algorithms are procedures for solving stated problems. A very simple example is the problem of multiplying two numbers. Multiplying small numbers such as 7 and 9 is trivial, because we can memorize the answers in advance. But to multiply larger numbers such as 1234 and 789 we need a step-by-step procedure, or algorithm. We all learn such an algorithm in school, using long multiplication or logarithms. (Even if we make an electronic calculator do the work, the calculator uses its own multiplication algorithm.)

Algorithms and computation have a long history, perhaps as long as civilization itself. The achievements of the ancient civilizations of China, Egypt, India, and Mesopotamia leave us in no doubt that their architects, astronomers, engineers, and merchants knew how to perform the computations associated with their professions. Unfortunately we have only fragmentary and indirect knowledge of the algorithms developed by these very ancient civilizations.

(*Note:* Much of the historical information in this chapter was culled from an excellent Web site at the University of St Andrews: www-history.mcs.st-and.ac.uk/history/.)

The scholars of the Hellenic civilization systematized many of the achievements of their predecessors, and added enormous contributions of their own. Athens, Alexandria, and other Hellenic cities contained ‘academies’ where scholars worked and taught together – forerunners of modern universities. Scholars such as Euclid (about 325–265 BCE) and Eratosthenes (276–194 BCE), discovered or perfected a variety of algorithms that are still used today.