

3

The Array Data Structure

In this chapter we shall study:

- arrays, their general properties, and their specific properties in Java (Section 3.1)
- array insertion, deletion, searching, merging, and sorting algorithms (Sections 3.2–6).

3.1 Arrays

Arrays are directly supported by nearly all major programming languages, and are familiar to every programmer. Nevertheless, a brief review of arrays will be useful here, and will serve as an opportunity to disentangle the general properties of arrays from their properties in specific programming languages such as Java. The array algorithms discussed in this chapter are expressed entirely in terms of the general properties of arrays.

An *array* is a sequence of indexed *components*, with the following general properties:

- The *length* of the array (its number of components) is fixed when the array is constructed.
- Each component of the array has a fixed and unique *index*. The indices range from a *lower index bound* through a *higher index bound*.
- Any component of the array can be accessed (inspected or updated) using its index. This is an efficient operation, having time complexity $O(1)$.

Figure 3.1 shows an abstract view of an array, with each box representing a single component.

In this book we shall use the common notation ' $a[i]$ ' to denote the component of array a whose index is i .

Other properties of arrays vary from one programming language to another. In some languages (such as C, C++, and Java) indices are integers, with the lower index bound being zero; in some other languages (such as Pascal and Ada) the programmer is free to choose the lower and upper index bounds, and even to choose the type of the indices. In strongly-typed languages an array must be *homogeneous*, i.e., all its components must