

Chapter 2

Arrays, Iteration, Invariants

Data is stored in computers as patterns of bits, but most programming languages deal with higher level objects such as characters, integers, and floating point numbers. To build algorithms that manipulate collections of such objects, we need procedures for storing and sequentially processing them.

2.1 Arrays

In computer science, the obvious way to store an ordered collection of items is as an array. Array items are typically stored in a sequence of computer memory locations, but to discuss them, we need a convenient way to write them down on paper. We can just write the items in order, separated by commas and enclosed by square brackets. Thus,

$$[1, 4, 17, 3, 90, 79, 4, 6, 81]$$

is an example of an array of integers. If we call this array a , we can write it as:

$$a = [1, 4, 17, 3, 90, 79, 4, 6, 81]$$

This array a has 9 items, and hence we say that its size is 9. In everyday life, we usually start counting from 1. When we work with arrays in computer science, however, we more often (though not always) start from 0. Thus, for our array a , its positions are 0, 1, 2, ..., 7, 8. The element in the 8th position is 81, and we use the notation $a[8]$ to denote this element. More generally, for any integer i denoting a position, we write $a[i]$ to denote the element in the i th position. This position i is called an index (and the plural is indices). Then, in the above example, $a[0] = 1$, $a[1] = 4$, $a[2] = 17$, and so on. It is worth noting at this point that the symbol $=$ is quite overloaded. In mathematics, it stands for equality. In most modern programming languages, $=$ denotes assignment, while equality is expressed by $==$. We will typically use $=$ in its mathematical meaning, unless it is written as part of code or pseudocode.

The individual items in an array $a[i]$ are accessed using their index i , and can be moved sequentially by incrementing or decrementing that index. Algorithms that process data stored as arrays need to visit all the items in the array and apply appropriate operations on them.

2.2 Loops and Iteration

The standard approach in most programming languages for repeating a process a certain number of times, such as moving sequentially through an array to perform