



Figure 1.2 The operation of INSERTION-SORT on the array $A = \langle 5, 2, 4, 6, 1, 3 \rangle$. The position of index j is indicated by a circle.

position $j - 1$, elements are successively moved one position to the right until the proper position for $A[j]$ is found (lines 4–7), at which point it is inserted (line 8).

Pseudocode conventions

We use the following conventions in our pseudocode.

1. Indentation indicates block structure. For example, the body of the **for** loop that begins on line 1 consists of lines 2–8, and the body of the **while** loop that begins on line 5 contains lines 6–7 but not line 8. Our indentation style applies to **if-then-else** statements as well. Using indentation instead of conventional indicators of block structure, such as **begin** and **end** statements, greatly reduces clutter while preserving, or even enhancing, clarity.¹
2. The looping constructs **while**, **for**, and **repeat** and the conditional constructs **if**, **then**, and **else** have the the same interpretation as in Pascal.
3. The symbol “ \triangleright ” indicates that the remainder of the line is a comment.
4. A multiple assignment of the form $i \leftarrow j \leftarrow e$ assigns to both variables i and j the value of expression e ; it should be treated as equivalent to the assignment $j \leftarrow e$ followed by the assignment $i \leftarrow j$.
5. Variables (such as i , j , and key) are local to the given procedure. We shall not use global variables without explicit indication.

¹In real programming languages, it is generally not advisable to use indentation alone to indicate block structure, since levels of indentation are hard to determine when code is split across pages.

6. Array elements are accessed by specifying the array name followed by the index in square brackets. For example, $A[i]$ indicates the i th element of the array A . The notation “ \dots ” is used to indicate a range of values within an array. Thus, $A[1..j]$ indicates the subarray of A consisting of elements $A[1], A[2], \dots, A[j]$.
7. Compound data are typically organized into **objects**, which are comprised of **attributes** or **fields**. A particular field is accessed using the field name followed by the name of its object in square brackets. For example, we treat an array as an object with the attribute *length* indicating how many elements it contains. To specify the number of elements in an array A , we write $length[A]$. Although we use square brackets for both array indexing and object attributes, it will usually be clear from the context which interpretation is intended.

A variable representing an array or object is treated as a pointer to the data representing the array or object. For all fields f of an object x , setting $y \leftarrow x$ causes $f[y] = f[x]$. Moreover, if we now set $f[x] \leftarrow 3$, then afterward not only is $f[x] = 3$, but $f[y] = 3$ as well. In other words, x and y point to (“are”) the same object after the assignment $y \leftarrow x$.

Sometimes, a pointer will refer to no object at all. In this case, we give it the special value **NIL**.

8. Parameters are passed to a procedure **by value**: the called procedure receives its own copy of the parameters, and if it assigns a value to a parameter, the change is *not* seen by the calling routine. When objects are passed, the pointer to the data representing the object is copied, but the object’s fields are not. For example, if x is a parameter of a called procedure, the assignment $x \leftarrow y$ within the called procedure is not visible to the calling procedure. The assignment $f[x] \leftarrow 3$, however, is visible.

So copies pointer pointing
to object passed to
called procedure.

Exercises

1.1-1

Using Figure 1.2 as a model, illustrate the operation of INSERTION-SORT on the array $A = \langle 31, 41, 59, 26, 41, 58 \rangle$.

1.1-2

Rewrite the INSERTION-SORT procedure to sort into nonincreasing instead of nondecreasing order.

1.1-3

Consider the **searching problem**:

Input: A sequence of n numbers $A = \langle a_1, a_2, \dots, a_n \rangle$ and a value v .

Output: An index i such that $v = A[i]$ or the special value **NIL** if v does not appear in A .