

C. L. R. S' Introduction to Algorithms - A partial solutions manual

George Lydakis

May 19, 2025

Contents

I	Foundations	5
2	Getting Started	7
2.1	Insertion sort	7

Part I

Foundations

Chapter 2

Getting Started

2.1 Insertion sort

Exercise 1

Using Figure 2.2 as a model, illustrate the operation of INSERTION-SORT on an array initially containing the sequence $\langle 31, 41, 59, 26, 41, 58 \rangle$.

Solution.

We show the steps of the algorithm below, underlining the element being processed in each iteration. The element is shown as already positioned in its correct place, implying that the elements that follow it (up to and including position i for the i -th iteration) have been shifted rightwards:

$$\begin{aligned} \langle \underline{31}, 41, 59, 26, 41, 58 \rangle &\rightarrow \langle 31, \underline{41}, 59, 26, 41, 58 \rangle \rightarrow \langle 31, 41, \underline{59}, 26, 41, 58 \rangle \rightarrow \\ &\langle \underline{26}, 31, 41, 59, 41, 58 \rangle \rightarrow \langle 26, 31, 41, \underline{41}, 59, 58 \rangle \rightarrow \langle 26, 31, 41, 41, \underline{58}, 59 \rangle \end{aligned}$$

Exercise 2

Consider the procedure SUM-ARRAY:

SUM-ARRAY(A, n)

```
1  sum = 0
2  for i = 1 to n
3      sum = sum + A[i]
4  return sum
```

It computes the sum of the n numbers in array $A[1 : n]$. State a loop invariant for this procedure, and use its initialization, maintenance, and termination properties to show that the SUM-ARRAY procedure returns the sum of the numbers in $A[1 : n]$.

Solution.

The loop invariant is: at the start of the i -th iteration, the variable sum contains precisely the sum of A 's elements at indices $1, 2, \dots, i - 1$.

- **Initialization:** At the start of iteration 1, line 1 has caused sum to contain 0, which equals the sum of A 's elements for the empty collection of indices.
- **Maintenance:** Suppose the property holds before iteration i , which means $sum = \sum_{k=1}^{i-1} A[k]$. Then, the execution of iteration i means that line 3 now causes sum to be incremented by $A[i]$, and thus $\sum_{k=1}^i A[k]$. But this means precisely that the invariant is true before iteration $i + 1$ as well.
- **Termination:** The loop trivially terminates due to running for a constant number of iterations n . Furthermore, we can think of its end as the start of an “imaginary” iteration $n + 1$. Then, due to the invariant, it will hold that $sum = \sum_{k=1}^n A[k]$, which is the sum of all elements in the array.