

---

## Insertion Sort

**Input:** Array A[1..n]

**Output:** Array A is sorted

```
1)   for  $j \leftarrow 2$  to  $n$ 
2)      $x = A[j]$ 
3)      $i = j - 1$ 
4)     while (  $i > 0$  and  $A[i] > x$ )
5)        $A[i + 1] \leftarrow A[i]$ 
6)        $i \leftarrow i - 1$ 
       endwhile
7)      $A[i + 1] \leftarrow x$ 
     endfor
```

---

Figure 1: Insertion Sort

## 1 Analysis

The number of times each instruction is executed is given below.

1. The **for** loop has 1 assignment,  $n$  comparisons, and another  $n - 1$  assignments (the increment).
2.  $n - 1$  assignments, each with an extra memory access
3.  $n - 1$  assignments
4.  $2t_j + 2$  comparisons checking the condition in the **while** loop for each  $j$ , where  $t_j$  is the number of elements of the array we need to move to insert  $A[j]$  among  $A[1], \dots, A[j - 1]$ . Another  $t_j + 1$  AND operations. Total:  $3(n - 1) + 3 \sum_{j=2}^n t_j$
5.  $t_j$  assignments for each  $j$ , with two extra memory access each. Total:  $\sum_{j=2}^n t_j$
6.  $t_j$  assignments for each  $j$ . Total:  $\sum_{j=2}^n t_j$
7.  $n - 1$  assignments, each with an extra memory access

The total overall is  $n + 3(n - 1) + 2 \sum_{j=2}^n t_j$  assignments and  $n + 2(n - 1) + 2 \sum_{j=2}^n t_j$  comparisons, and another  $(n - 1) + \sum_{j=2}^n t_j$  AND operations. Total  $2n + 6(n - 1) + 5 \sum_{j=2}^n t_j$  “elementary” operations. Each assignment involves several (here, up to five) LOAD and STORE operations, as well as several arithmetic operations.