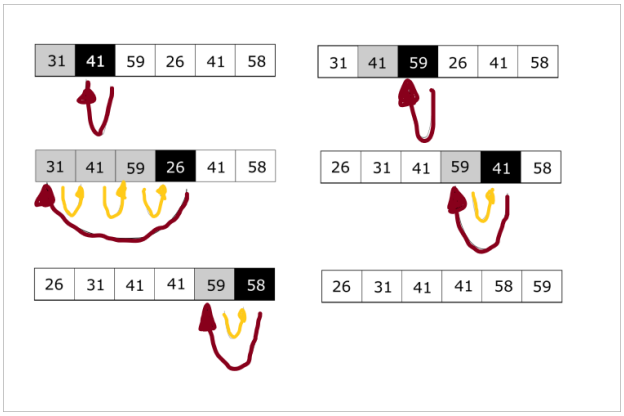
Ejercicios Cormen

Miguel Ángel Castro Cortés

I. 2.1-1

Figura 1: INSERTION-SORT on the Array A=(31; 41; 59; 26; 41; 58)



II. 2.1-2

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

```
for j = 2 to A.length
key = A[j]
i = j - 1
while i > 0 and A[i] < key
    A[i + 1] = A[i]
    i = i - 1
A[i + 1] = key
```

III. 2.1-3

.

```
for i = 1 to A.length
    if A[i]==v
        return v
return NIL
```

- Initialization: The subarray is the empty array.
- Maintenance: In each iterarion we know v is not inside A[1..i-1], else we return the value in A[i], which is correct. So we know that v is not inside A[A..i-1] and that A[i] is not the same from v, thus this continues to be an invariant.
- Termination: The loops ends when i greater than A.length . i increases by 1 and we fulfill that i greater than A.length, so every index in A is checked, therefore v is not in A and we return NIL.

IV. 2.1-4

. Input: An array of booleans A=a1,a2,...,an, an array of booleans v=v1,v2,...,vn, each representing an integer stored in binary format and each of length n.

Output: An array num=num1,num2,...,numn+1 that such that num=A+v, where A, v and num are the integers, represented by A, v and num.

```
num = new integer[A.length + 1]

cr = 0
for i = 1 to A.length
    num[i] = (A[i] + v[i] + cr) %2
    cr = (A[i] + v[i] + cr) / 2
num[i] = cr
return num
```