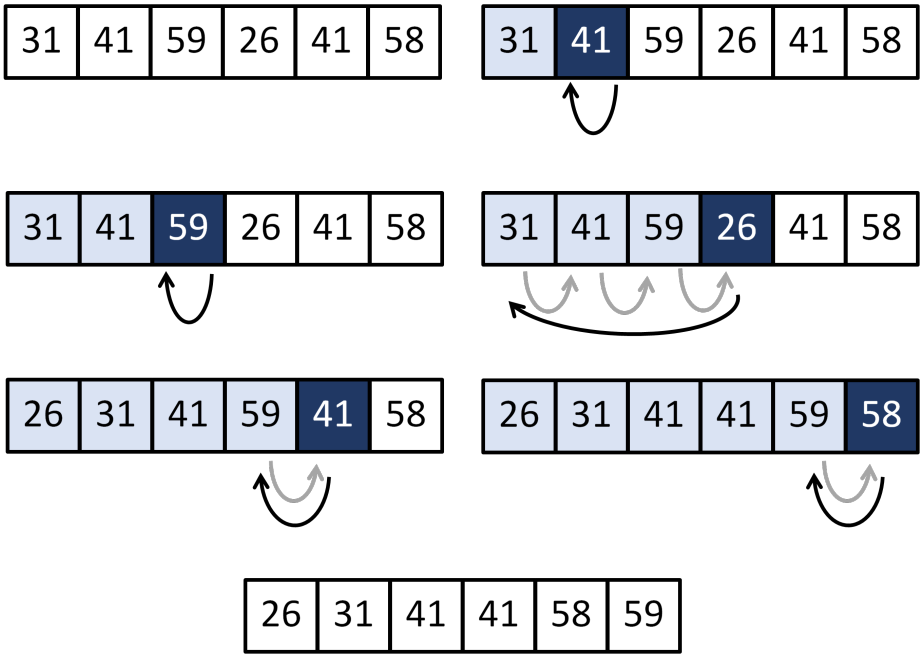
Laboratory #2

Christian Fernando Ortiz Pulido

September 2018

1 Exercise 2.1-1

Using Figure 2.2 as a model, illustrate the operation of INSERTION-SORT on the array A = (31, 41, 59, 26, 41, 58).



2 Exercise 2.1-2

Rewrite the INSERTION-SORT procedure to sort into non increasing instead of non decreasing order.

```
j = 2
```

```
while j < A.length
    element = A[j]
    i = j-1
    while i > 0 and A[i] < element
        A[i+1] = A[i]
        i = i-1
    A[i+1] = element
    j = j+1
```

3 Exercise 2.1-3

Consider the **searching problem**:
Input: A sequence of n numbers A $=(a_1, a_2, ..., a_n)$ and a value v.
Output: An index i such that v = A[i] or the special value NIL if does not appear in A.
Write pseudocode for **linear search**, which scans through the sequence, looking for. Using a loop invariant, prove that your algorithm is correct. Make sure that your loop invariant fulfills the three necessary properties. Pseudocode for linear search

```
i = 1
while i < A.length
    if A[i] == v
        return i
    i = i+1
return NULL
```

Initialization: The subarray at the start is empty.
Maintenance: In each i-th iteration, it is verified if the position i of the array A, A [i] is equal to v. If it is the same, the cycle is completed and the iteration value returns, if it does not continue with the cycle.
Termination: The cycle is terminated when a value of the array is found equal to v, A [i] = v, or when i is greater than or equal to the size of the array, i¿ = A.length

4 Exercise 2.1-4

Consider the problem of adding two n-bit binary integers, stored in two n-element arrays **A** and **B**. The sum of the two integers should be stored in binary form in an (n + 1)-element array **C**. State the problem formally and write pseudocode for adding the two integers.
Input: Two arrays of binary numbers (0,1) of size n, A and B.
Output: An array of size n + 1, C such that C = A + B.

```
n = A.length
C = []
i = 0
while i < (n+1)
    C[i] = 0
    i = i+1
c = 0
i = n
while i > 0
    sum = A[i-1] + B[i-1] + c
    C[i] = sum % 2
    c = sum/2
    i = i-1
C[i] = c
```