

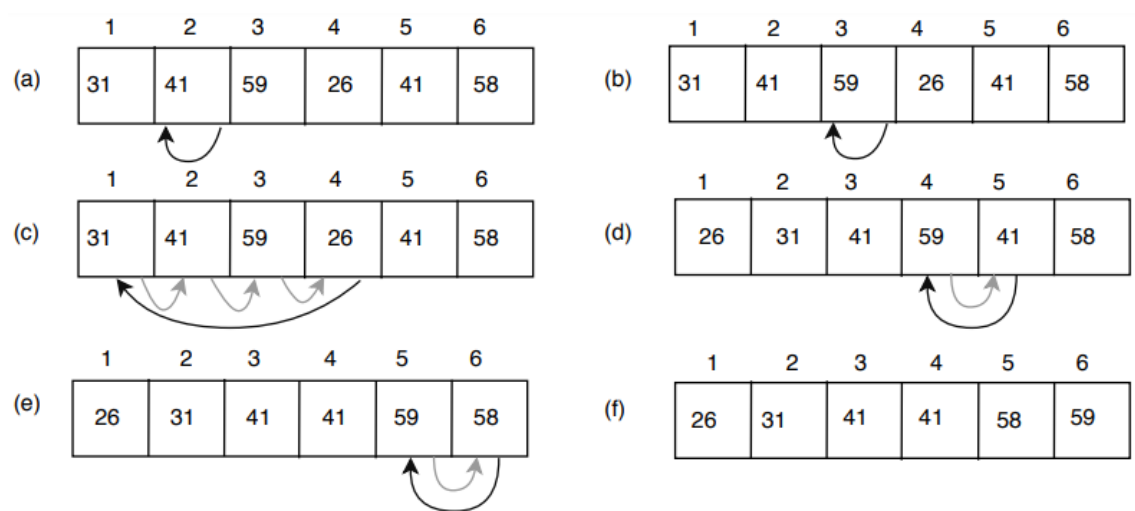
Algoritmos

David Fernando Guerrero Alvarez

September 2018

1 Taller 1

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



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

```
def insertionSortInverse(array):
    for i in range(1, len(array)):
        aux = array[i]
        j = i - 1
        while j > -1 and aux > array[j]:
            array[j + 1] = array[j]
            j = j - 1
        array[j + 1] = aux
```

In: [5, 4, 6, 3, 7, 2, 8, 1, 9]
Out: [9, 8, 7, 6, 5, 4, 3, 2, 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 .

Write pseudocode for *linear search*, which scans through the sequence, looking for v . Using a loop invariant, prove that your algorithm is correct. Make sure that your loop invariant fulfills the three necessary properties.

```

Programa: Búsqueda secuencial
Variables
tam=10:entero
A[tam]:arreglo de tamaño tam
temp,i,j,n:entero
x:booleano
Inicio
  //Llenar arreglo
  Para i=0 hasta i< tam incremento 1 hacer
    A[i]=númeroaleatorio
  Fin Para
  //Busqueda lineal
  leer n
  x=falso
  Para i=0 hasta i< tam incremento 1 hacer
    Si A[i] = n entonces
      escribir "Valor encontrado"
      escribir 'Posición:', i
      x=verdadero
    Fin Si
  Fin Para
  Si x=falso entonces
    escribir 'No se encontró el número'
  Fin Si
Fin

```

Inicializacion: El inicio se da al llenar un arreglo de tamaño 10 con numeros aleatorios. Esto da paso a la lectura del número que se desea buscar y se compara dicho número con el valor que está en la primera posicion del arreglo.

Mantenimiento: Se realiza la comparación del valor a encontrar con cada uno de los terminos del arreglo, de manera secuencial, de manera que el valor de x se mantiene invariante a menos que encuentre el valor deseado, para luego proseguir con el proceso.

Terminacion:

El algoritmo finaliza cuando ha recorrido cada uno de los elementos del arreglo $i = tam+1$, si no se encuentra ninguna coincidencia entre el valor inicial y los valores del arreglo, entonces $x = falso$ y escribe "No se encontro el número".

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.

Declaramos las variables A,B,C:

```

A[0] ... A[n-1] (tam = n)
B[0] ... B[n-1] (tam = n)
C[0] ... C[n] (tam = n+1)
A[0] y B[0] son los bits más significativos.

```

Pseudocódigo:

```

Carry = 0
Para i = n - 1 a 0
  C[i+1] = (A[i] + B[i] + Carry) mod 2
  Carry = (A[i] + B[i] + Carry) / 2
C[0] = Carry

```