

ORDENAMIENTO POR EL MÉTODO DE BURBUJA

Bubble Sort



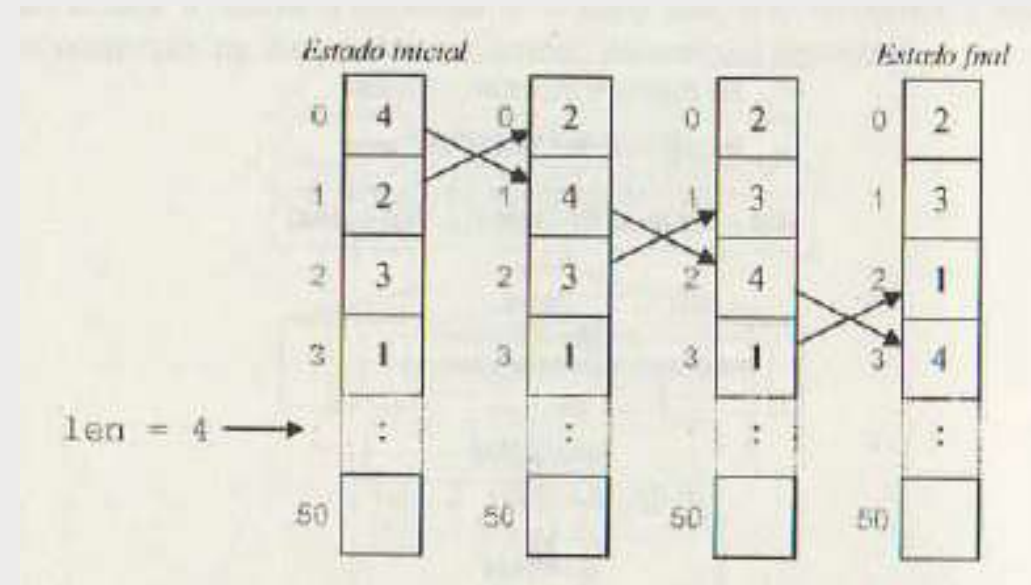
Ordenamiento por burbuja

- El algoritmo de la “burbuja” consiste en recorrer el *arreglo* analizando la relación de precedencia que existe entre cada elemento y el elemento que le sigue para determinar si estos se encuentran ordenados entre sí y, en caso de ser necesario, permutarlos para que queden ordenados.
- Es necesario revisar varias veces todo el *arreglo* hasta que no se necesiten más intercambios, lo cual significa que el *arreglo* está ordenado.

- Si **a** es el *arreglo* que vamos a ordenar e **i** es un valor comprendido entre 0 y $\text{len}-1$ entonces:
 - Si $a[i] > a[i+1]$ significa que estos dos elementos se encuentran desordenados entre si y habrá que permutarlos.
 - Entonces, se toman de a pares los elementos del arreglo, se comparan y si corresponde los permutamos para que cada par de elementos quedé ordenado entre sí.

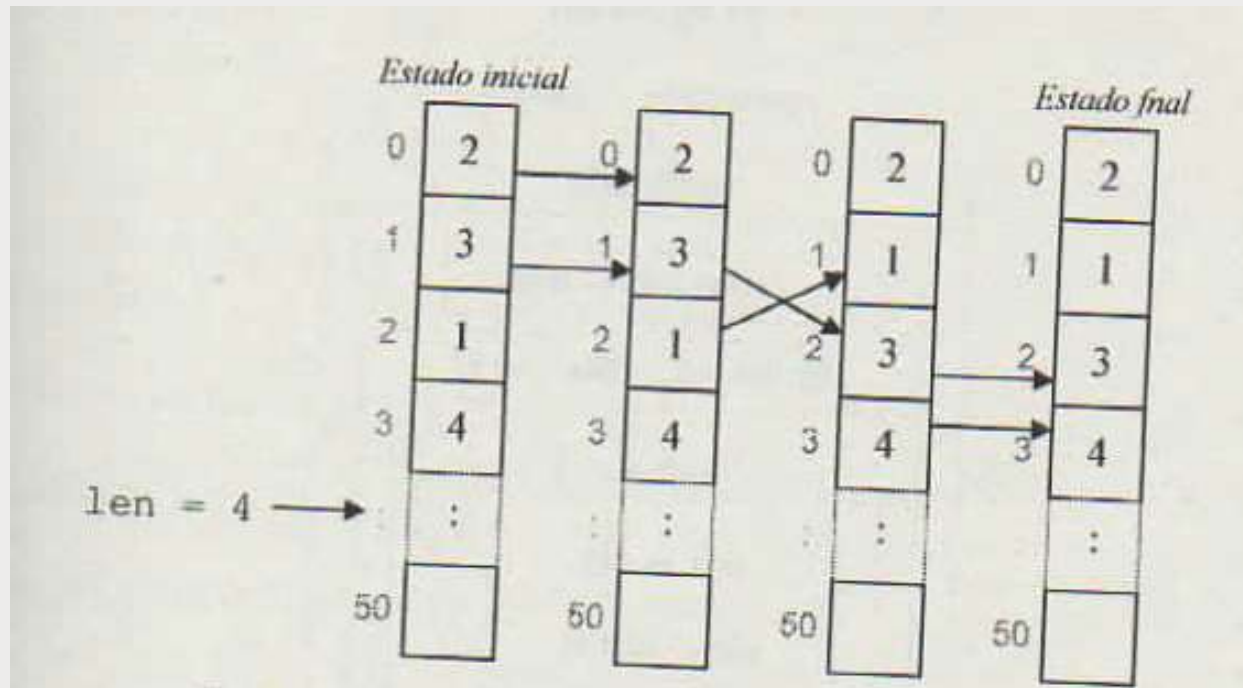
■ Entonces se evalúa:

- $a[0]$ y $a[1]$
- $a[1]$ y $a[2]$
- $a[2]$ y $a[3]$
- $a[3]$ y $a[4]$
-
- $a[n-2]$ y $a[n-1]$



Primera pasada de ordenamiento “burbuja”

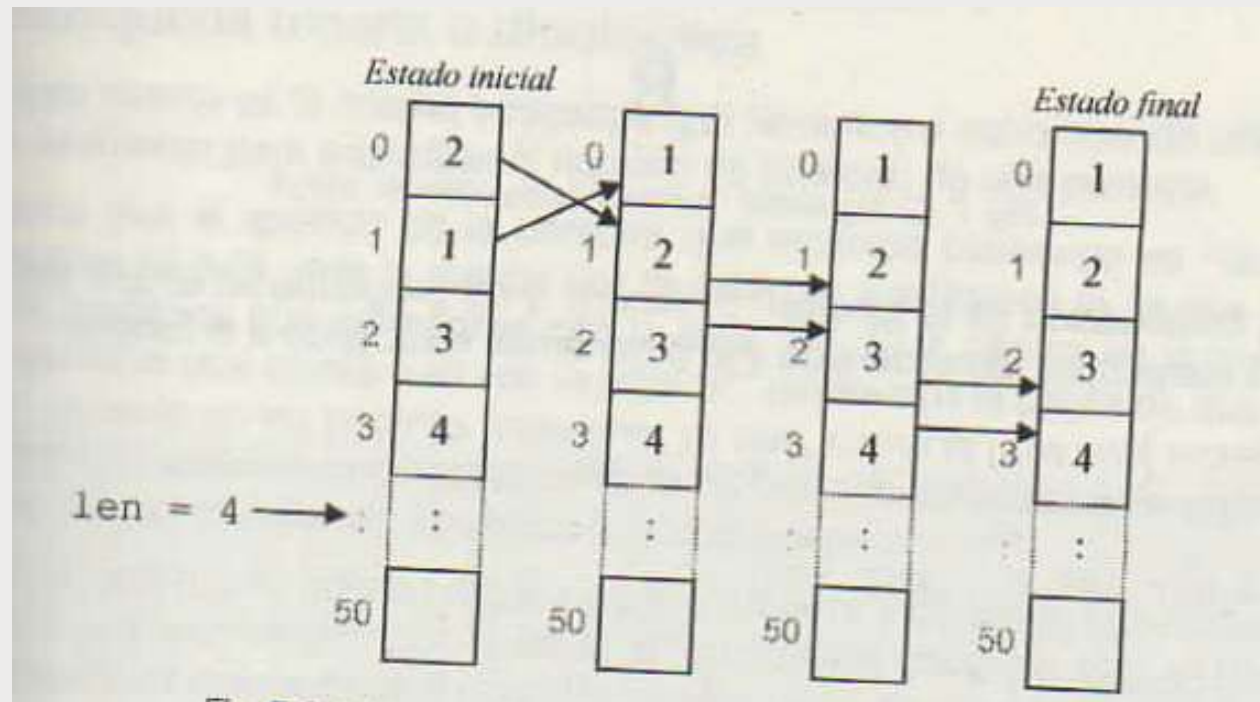
Luego de realizar todas las comparaciones y permutaciones, el arreglo quedó “un poco más ordenado” respecto de su estado inicial.



Segunda pasada de ordenamiento "burbuja"

Si volvemos a repetir la operación, seguramente el arreglo quedará mejor aún.

Notemos que el estado inicial de esta nueva iteración coincide con el estado final de la iteración anterior.



Tercera pasada de ordenamiento “burbuja”

Efectivamente, el arreglo quedó mucho mejor y a simple vista observamos que basta con una nueva iteración para realizar la última permutación con la cual su contenido quedará ordenado.

- El algoritmo será recorrer el arreglo comparando $a[i]$ con $a[i+1]$ para permutarlos si no están en orden.
- El proceso finalizará cuando realicemos una iteración en la cual no haya sido necesario realizar ninguna permutación.

Ordenamiento por burbuja

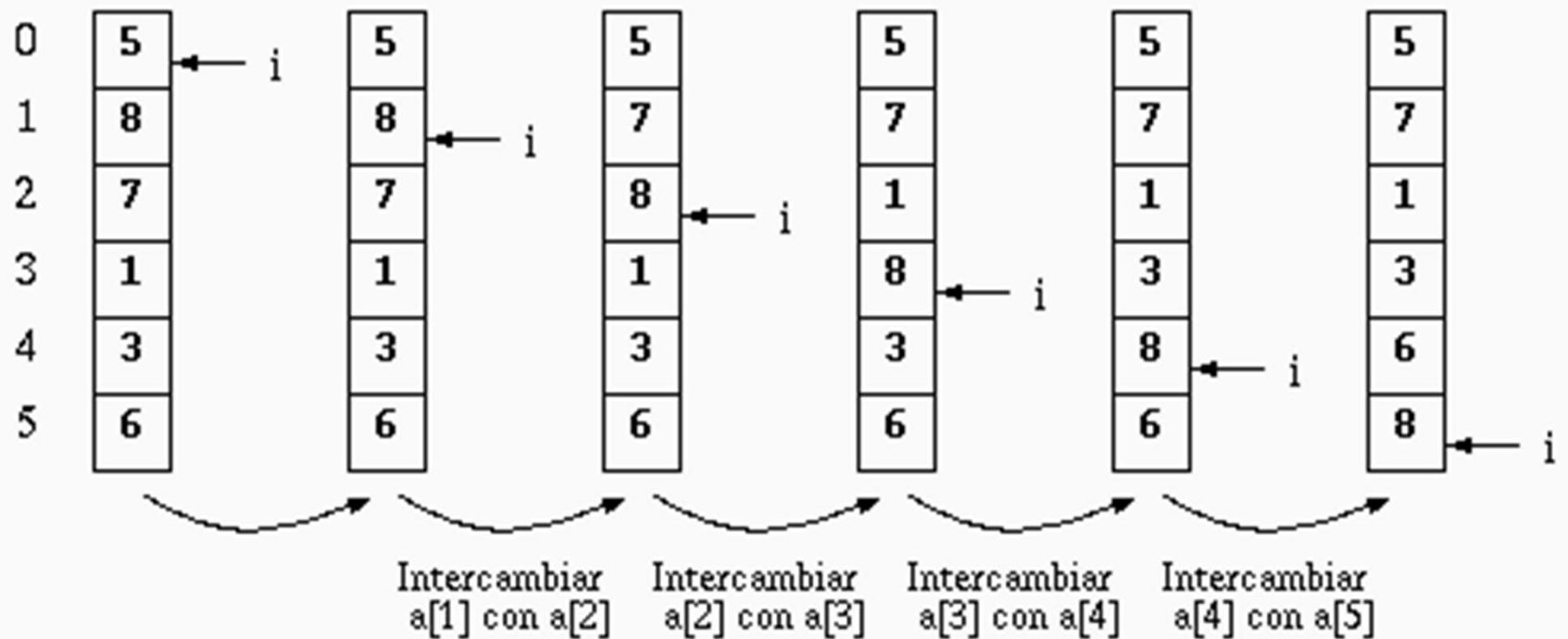
```
Inicio
  desde i=1 hasta N-1 hacer
    si elemento[i] > elemento[i+1] entonces
      intercambiar(elemento[i],elemento[i+1])
    fin_si
  fin_desde
Fin
```

Intercambio de datos

```
aux = A[i]
A[i] = A[i+1]
A[i+1] = aux
```


Ordenamiento por burbuja

Algoritmo de la burbuja



Algoritmo burbuja 1

```
procedimiento DeLaBurbuja ( $a_0, a_1, a_2, \dots, a_{(n-1)}$ )  
  para  $i \leftarrow 1$  hasta  $n - 1$  hacer  
    para  $j \leftarrow 0$  hasta  $n - i - 1$  hacer  
      si  $a_{(j)} > a_{(j+1)}$  entonces  
         $aux \leftarrow a_{(j)}$   
         $a_{(j)} \leftarrow a_{(j+1)}$   
         $a_{(j+1)} \leftarrow aux$   
      fin si  
    fin para  
  fin para  
fin procedimiento
```

Algoritmo burbuja 2

procedimiento DeLaBurbuja2 ($a_{(0)}, a_{(1)}, a_{(2)}, \dots, a_{(n-1)}$)

$i \leftarrow 1$

$\text{ordenado} \leftarrow \text{no}$

mientras $(i < n) \wedge (\text{ordenado} = \text{no})$ *hacer*

$i \leftarrow i + 1$

$\text{ordenado} \leftarrow \text{si}$

para $j \leftarrow 0$ *hasta* $n - i$ *hacer*

si $a_{(j)} > a_{(j+1)}$ *entonces*

$\text{ordenado} \leftarrow \text{no}$

$\text{aux} \leftarrow a_{(j)}$

$a_{(j)} \leftarrow a_{(j+1)}$

$a_{(j+1)} \leftarrow \text{aux}$

fin si

fin para

fin mientras

fin procedimiento

Algoritmo burbuja 3

procedimiento DeLaBurbuja3 ($a_{(0)}, a_{(1)}, a_{(2)}, \dots, a_{(n-1)}$)

$i \leftarrow 1$

repetir

$i \leftarrow i + 1$

$\text{ordenado} \leftarrow \text{si}$

para $j \leftarrow 0$ *hasta* $n - i$ *hacer*

si $a_{(j)} > a_{(j+1)}$ *entonces*

$\text{ordenado} \leftarrow \text{no}$

$\text{aux} \leftarrow a_{(j)}$

$a_{(j)} \leftarrow a_{(j+1)}$

$a_{(j+1)} \leftarrow \text{aux}$

fin si

fin para

hasta que $\neg(i < n) \vee (\text{ordenado} = \text{si})$

fin procedimiento

Complejidad algorítmica del algoritmo
de ordenamiento por burbuja

$$O(n^2)$$