

# Ordenación por selección directa

---

## *Complejidad de la ordenación por selección directa*

*¿Cuántas comparaciones se realizan?*

Bucle externo:  $N - 1$  ciclos

Tantas comparaciones como elementos queden en la lista:

$$(N - 1) + (N - 2) + (N - 3) + \dots + 3 + 2 + 1 =$$

$$N \times (N - 1) / 2 = (N^2 - N) / 2 \rightarrow O(N^2)$$

Mismo número de comparaciones en todos los casos

Complejidad:  $O(N^2)$  Igual que el método de inserción

Algo mejor (menos intercambios; uno en cada paso)

No es estable: intercambios “a larga distancia”

No se garantiza que se mantenga el mismo orden relativo original

Comportamiento no natural (trabaja siempre lo mismo)



# Fundamentos de la programación

---

## Método de la burbuja



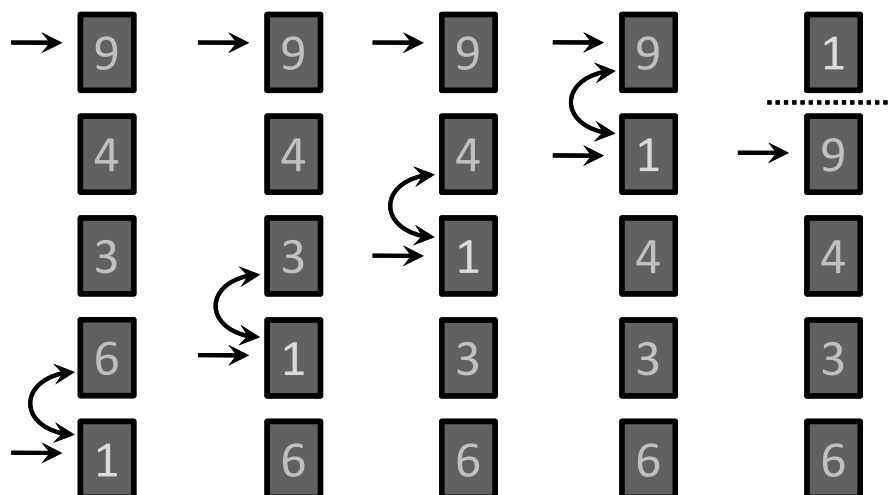
# Método de la burbuja



## Algoritmo de ordenación por el método de la burbuja

Variación del método de selección directa

El elemento menor va *ascendiendo* hasta alcanzar su posición



Luis Hernández Yáñez

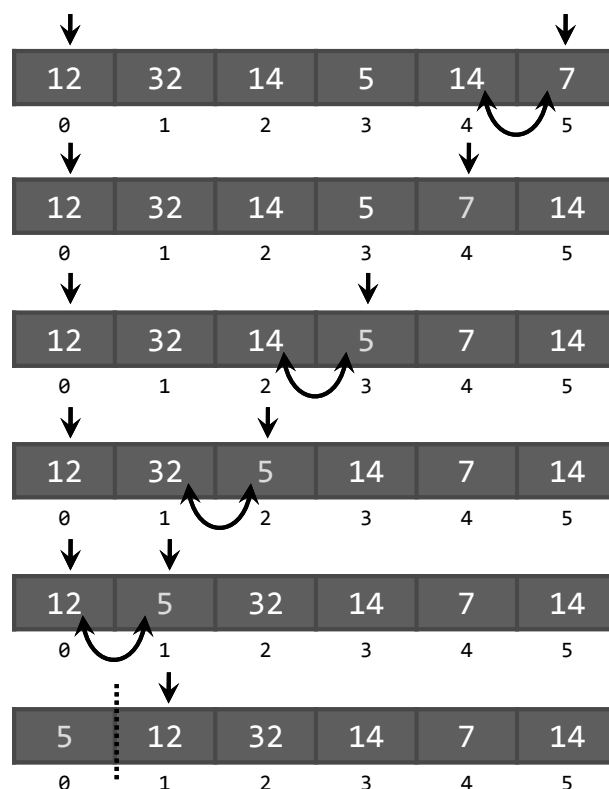


Fundamentos de la programación: Algoritmos de ordenación

Página 717



# Método de la burbuja



Luis Hernández Yáñez



Fundamentos de la programación: Algoritmos de ordenación

Página 718



## *Ordenación de un array por el método de la burbuja*

*Desde el primero ( $i = 0$ ), hasta el penúltimo ( $N - 2$ ):  
Desde el último ( $j = N - 1$ ), hasta  $i + 1$ :  
Si elemento en  $j <$  elemento en  $j - 1$ , intercambiarlos*

```
...
int tmp;
// Del primero al penúltimo...
for (int i = 0; i < N - 1; i++) {
    // Desde el último hasta el siguiente a i...
    for (int j = N - 1; j > i; j--) {
        if (lista[j] < lista[j - 1]) {
            tmp = lista[j];
            lista[j] = lista[j - 1];
            lista[j - 1] = tmp;
        }
    }
}
```

```
const int N = 10;
typedef int tLista[N];
tLista lista;
```

Luis Hernández Yáñez



# Método de la burbuja

## *Algoritmo de ordenación por el método de la burbuja*

Complejidad:  $O(N^2)$

Comportamiento no natural

Estable (mantiene el orden relativo)

Mejora:

Si en un paso del bucle exterior no ha habido intercambios:

La lista ya está ordenada (no es necesario seguir)

14	14	14	12
16	16	12	14
35	12	16	16
12	35	35	35
50	50	50	50

La lista ya está ordenada  
No hace falta seguir

Luis Hernández Yáñez



```
bool inter = true;
int i = 0;
// Desde el 1º hasta el penúltimo si hay intercambios...
while ((i < N - 1) && inter) {
    inter = false;
    // Desde el último hasta el siguiente a i...
    for (int j = N - 1; j > i; j--) {
        if (lista[j] < lista[j - 1]) {
            int tmp;
            tmp = lista[j];
            lista[j] = lista[j - 1];
            lista[j - 1] = tmp;
            inter = true;
        }
    }
    if (inter) {
        i++;
    }
}
```

Esta variación sí tiene un comportamiento natural



## Fundamentos de la programación

### Listas ordenadas

