Algoritmos de Ordenamiento

Programación Lógica

UNIVERSIDAD SIGLO 21

MIEMBRO DE LA RED ILUMNO

1. Ordenamiento por algoritmo de burbuja

La estrategia del método de ordenamiento por burbuja tiene su base en la comparación de pares de elementos adyacentes e intercambiarlos entre sí. Suponiendo que se desea ordenar un arreglo unidimensional por este método, entonces se deben recorrer sus elementos, se comparan de a pares en forma adyacente y se intercambian hasta llegar al final del arreglo. Este procedimiento se debe repetir hasta que todos los elementos del arreglo se encuentren ordenados.

Suponiendo un arreglo unidimensional *X* de *N* elementos, el pseudocódigo del algoritmo de ordenamiento por burbuja en orden creciente es el siguiente:

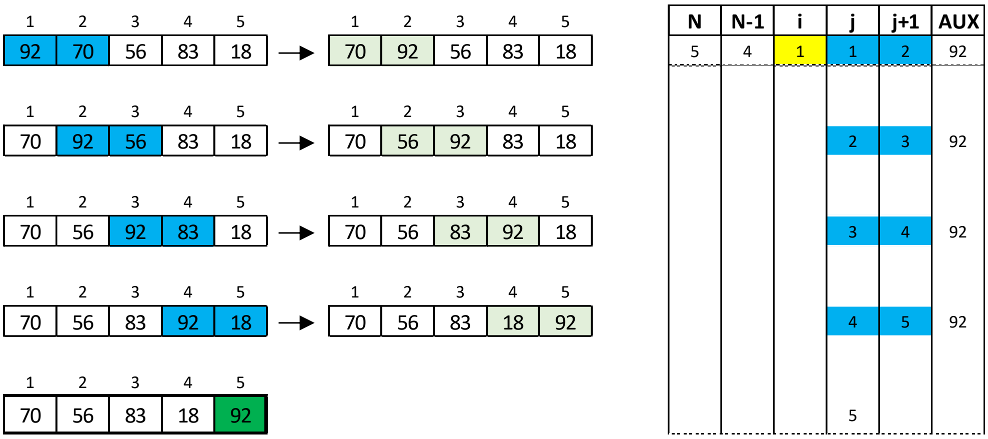
```
para i ← 1 hasta N-1 hacer
    para j ← 1 hasta N-1 hacer
        si X[j] > X[j+1] entonces
            // intercambiar
            AUX ← X[j]
            X[j] ← X[j+1]
            X[j+1] ← AUX
        fin-si
    fin-para
fin-para
```

Ejemplo de ordenamiento creciente de un arreglo unidimensional con método de burbuja

Arreglo inicial:

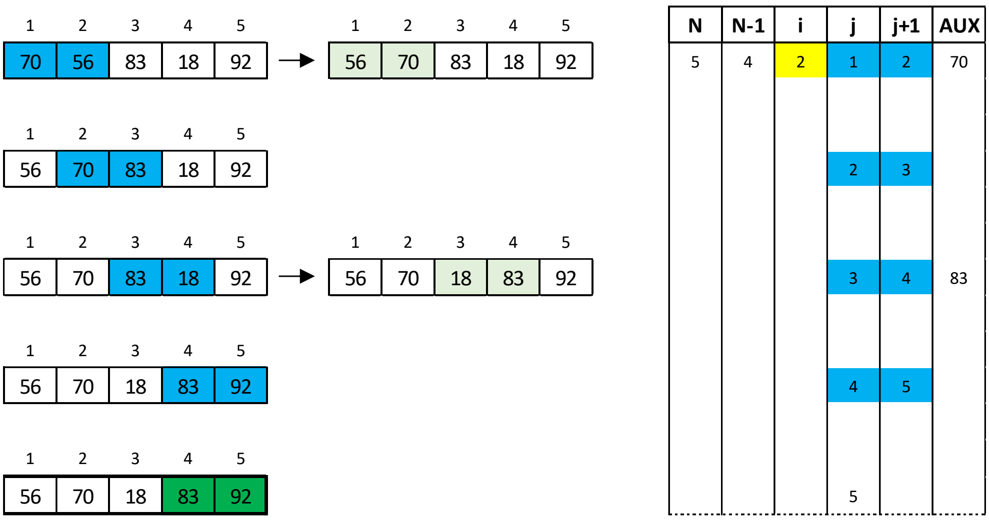
1	2	3	4	5
92	70	56	83	18

Figura 1: Iteración externa número 1 (i=1)



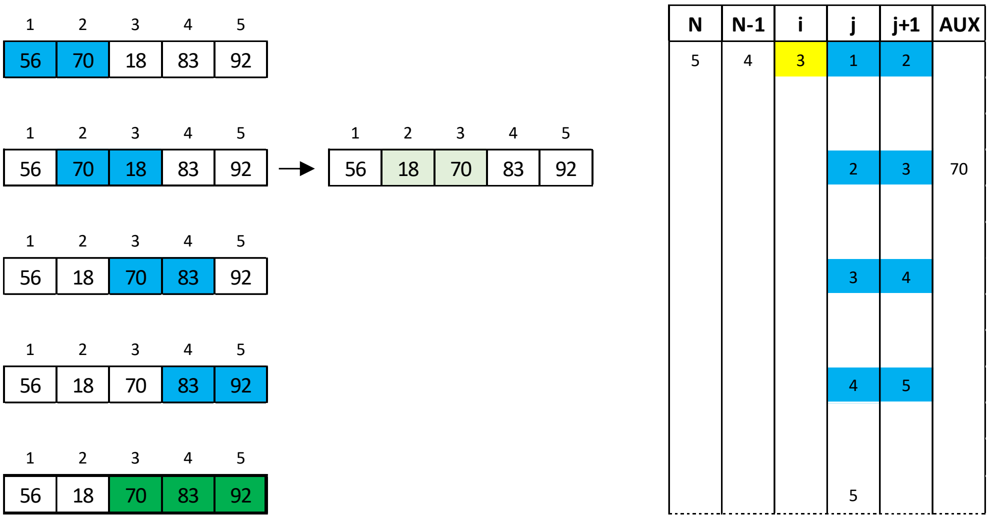
Fuente: elaboración propia.

Figura 2: Iteración externa número 2 (i=2)



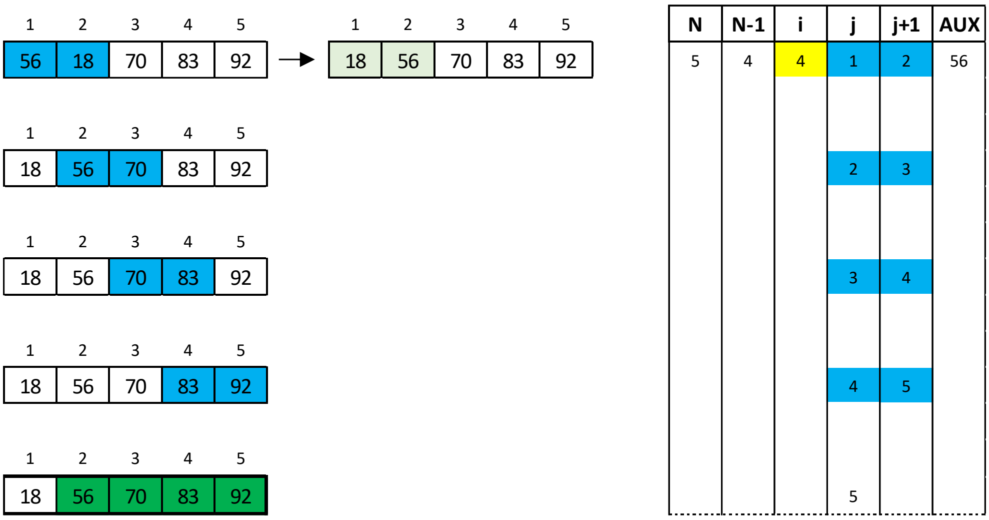
Fuente: elaboración propia.

Figura 3: Iteración externa número 3 (i=3)



Fuente: elaboración propia.

Figura 4: Iteración externa número 4 (i=4)



Fuente: elaboración propia.

Cantidad de comparaciones: se realizaron 4 iteraciones externas (N - 1) y, para cada una de ellas, se debieron realizar 4 comparaciones (N - 1).

$$\begin{aligned} \textit{Cantidad de comparaciones} &= (N - 1) \times (N - 1) \\ &= (5 - 1) \times (5 - 1) \\ &= 4 \times 4 \\ &= 16 \end{aligned}$$

2. Ordenamiento por algoritmo de burbuja mejorada

La cantidad de comparaciones que se realizan utilizando el algoritmo de burbuja se puede reducir con una pequeña modificación. Si se ordena crecientemente un arreglo unidimensional utilizando el método de burbuja, una vez finalizado el primer recorrido, el mayor elemento se encuentra en la última posición y, por consiguiente, no es necesario comparar el último par de elementos en el siguiente recorrido, ya que el elemento más grande ya fue ordenado. Esta lógica de evitar comparaciones innecesarias se puede repetir a medida que se ordenan los mayores elementos en las últimas posiciones. Esta variante del algoritmo de burbuja se denomina *burbuja mejorada*.

Suponiendo un arreglo unidimensional *X* de *N* elementos, el pseudocódigo del algoritmo de ordenamiento por burbuja mejorada en orden creciente es el siguiente:

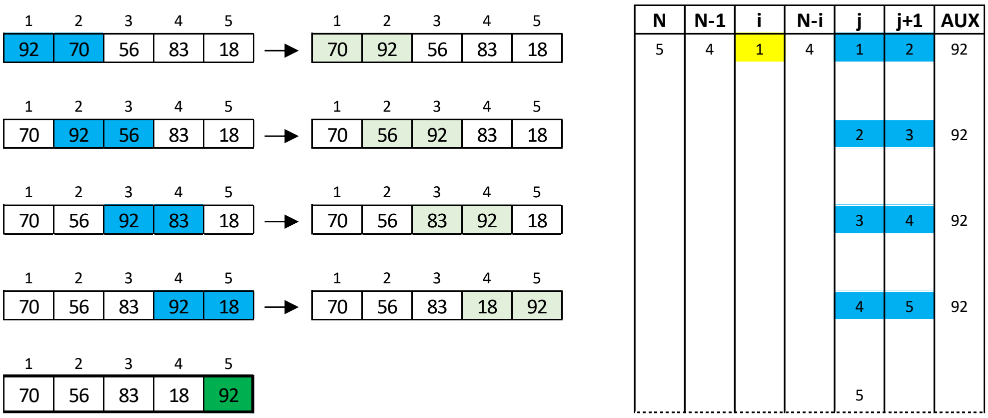
```
para i ← 1 hasta N-1 hacer
    para j ← 1 hasta N-i hacer
        si X[j] > X[j+1] entonces
            // intercambiar
            AUX ← X[j]
            X[j] ← X[j+1]
            X[j+1] ← AUX
        fin-si
    fin-para
fin-para
```

Ejemplo de ordenamiento creciente de un arreglo unidimensional con método de burbuja mejorada

Arreglo inicial:

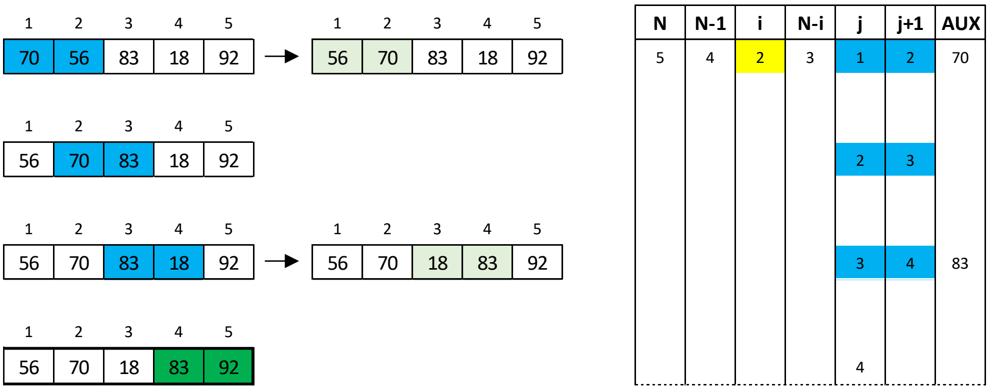
1	2	3	4	5
92	70	56	83	18

Figura 5: Iteración externa número 1 (i=1)



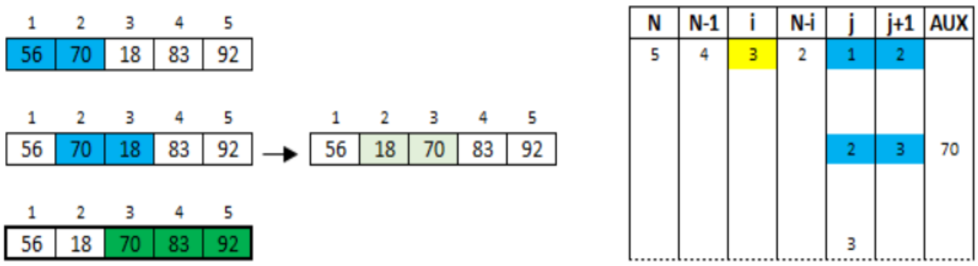
Fuente: elaboración propia.

Figura 6: Iteración externa número 2 (i=2)



Fuente: elaboración propia.

Figura 7: Iteración externa número 3 (i=3)



Fuente: elaboración propia.

Figura 8: Iteración externa número 4 (i=4)

1	2	3	4	5
56	18	70	83	92

→

1	2	3	4	5
18	56	70	83	92

1	2	3	4	5
18	56	70	83	92

N	N-1	i	N-i	j	j+1	AUX
5	4	4	1	1	2	56
				2		

Fuente: elaboración propia.

Cantidad de comparaciones $= (N^2 - N) / 2$
$= (5^2 - 5) / 2$
$= (25 - 5) / 2$
$= 20 / 2$
$= 10$

3. Ordenamiento por algoritmo de inserción

La estrategia de este método de ordenamiento consiste en tomar un elemento del conjunto e insertarlo en una parte ya ordenada. Se debe repetir este procedimiento para el resto de los elementos que restan por ordenar.

Suponiendo un arreglo unidimensional *X* de *N* elementos, el pseudocódigo del algoritmo de ordenamiento por inserción en orden creciente es el siguiente:

```
para i ← 2 hasta N hacer
    AUX ← X[i]
    k ← i-1
    sw ← falso
    mientras NO(sw) y (K >= 1) hacer
        si AUX < X[k] entonces
            X[k+1] ← X[k]
            k ← k -1
        si-no
            sw ← verdadero
```

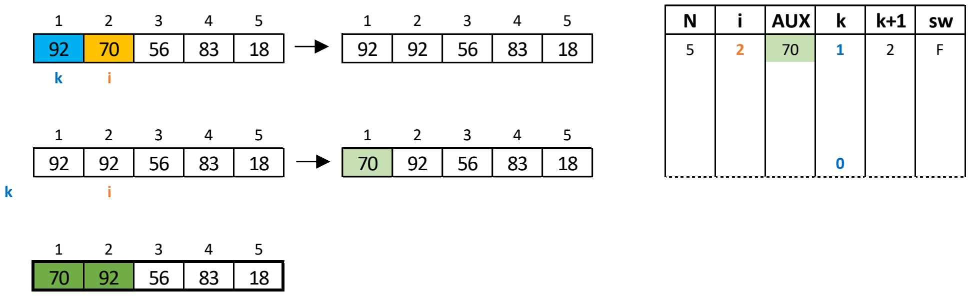
fin-si
fin-mientras
X[k+1] ← AUX
fin-para

Ejemplo de ordenamiento creciente de un arreglo unidimensional con método de inserción

Arreglo inicial:

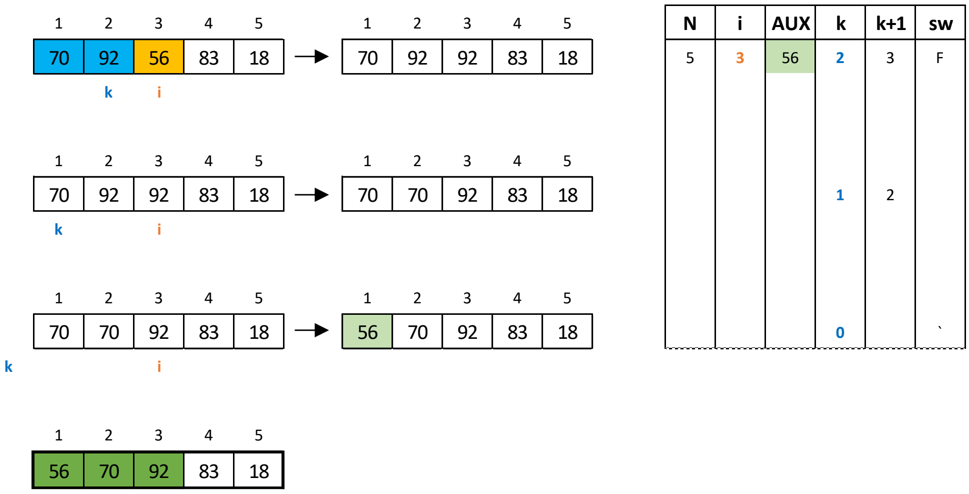
1	2	3	4	5
92	70	56	83	18

Figura 9: Iteración externa número 1 (i=2)



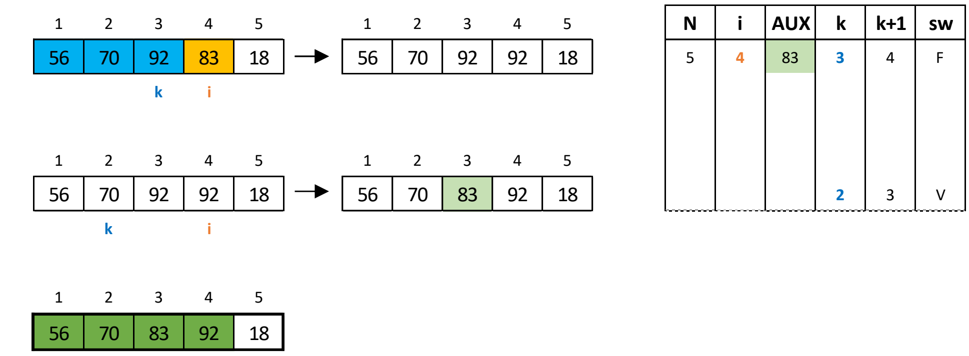
Fuente: elaboración propia.

Figura 10: Iteración externa número 2 (i=3)



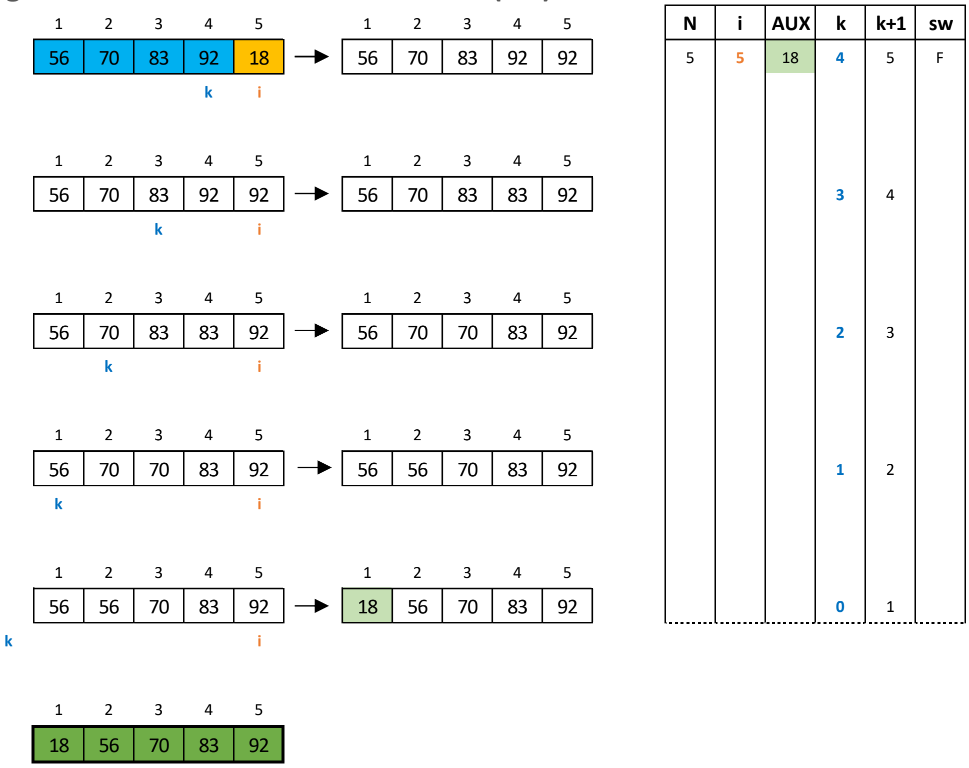
Fuente: elaboración propia.

Figura 11: Iteración externa número 3 (i=4)



Fuente: elaboración propia.

Figura 12: Iteración externa número 4 (i=5)



Fuente: elaboración propia.

La aplicación de este método asegura la siguiente cantidad de comparaciones:

Cantidad de comparaciones mínimas = $N - 1$

Cantidad de comparaciones máximas = $\underline{N \times (N - 1)}$

2

Cantidad de comparaciones en promedio $= \dfrac{N^2 + N - 2}{4}$

4. Ordenamiento por algoritmo de selección

Suponiendo que se desea ordenar un arreglo unidimensional en forma creciente, el método de ordenamiento por selección consiste en recorrer el arreglo para buscar el menor elemento e intercambiarlo con el elemento de la primera posición. A continuación, se debe recorrer nuevamente el arreglo para buscar el segundo menor elemento e intercambiarlo con el elemento de la segunda posición. Este procedimiento se debe aplicar para el resto de los elementos hasta que el arreglo se encuentre ordenado.

Suponiendo un arreglo unidimensional *X* de *N* elementos, el pseudocódigo del algoritmo de ordenamiento por selección en orden creciente es el siguiente:

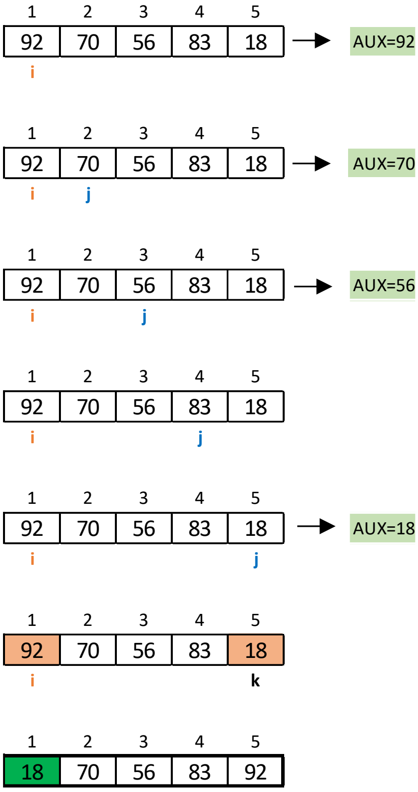
```
para i ← 1 hasta N-1 hacer
    AUX ← X[i]
    k ← i
    para j ← i+1 hasta N hacer
        si X[j] < AUX entonces
            AUX ← X[j]
            k ← j
        fin-si
    fin-para
    X[k] ← X[i]
    X[i] ← AUX
fin-para
```

Ejemplo de ordenamiento creciente de un arreglo unidimensional con método de selección

Arreglo inicial:

1	2	3	4	5
92	70	56	83	18

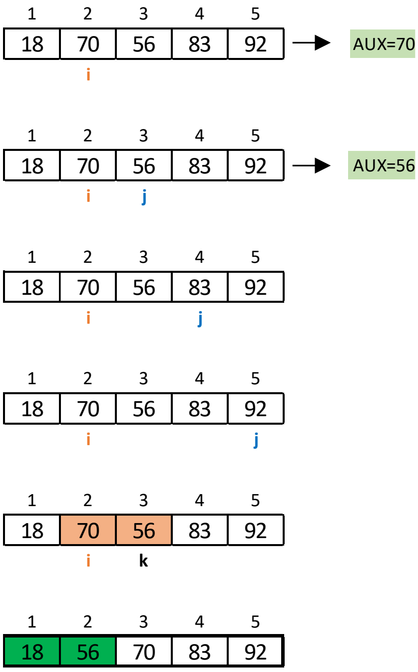
Figura 13: Iteración externa número 1 (i=1)



N	N-1	i	AUX	k	j
5	4	1	92	1	
			70	2	2
			56	3	3
					4
			18	5	5
					6

Fuente: elaboración propia.

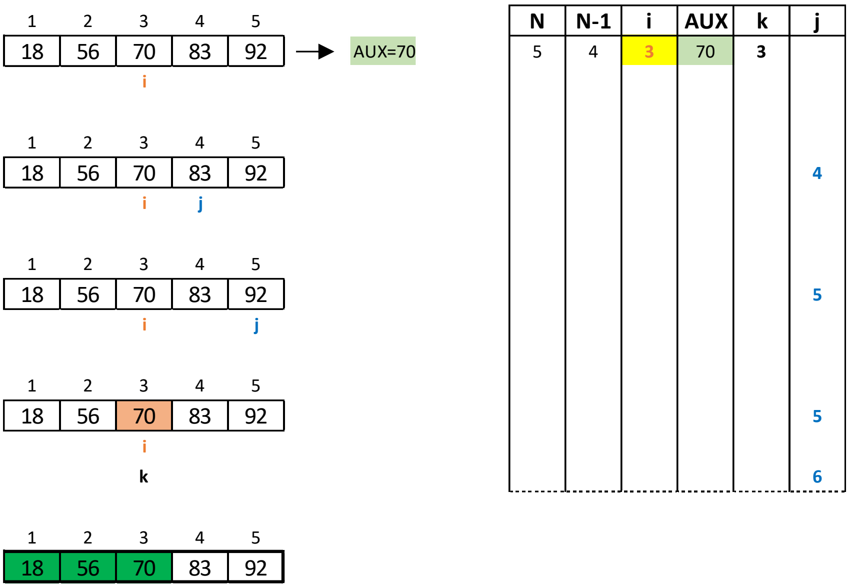
Figura 14: Iteración externa número 2 (i=2)



N	N-1	i	AUX	k	j
5	4	2	70	2	
			56	3	3
					4
					5
					6

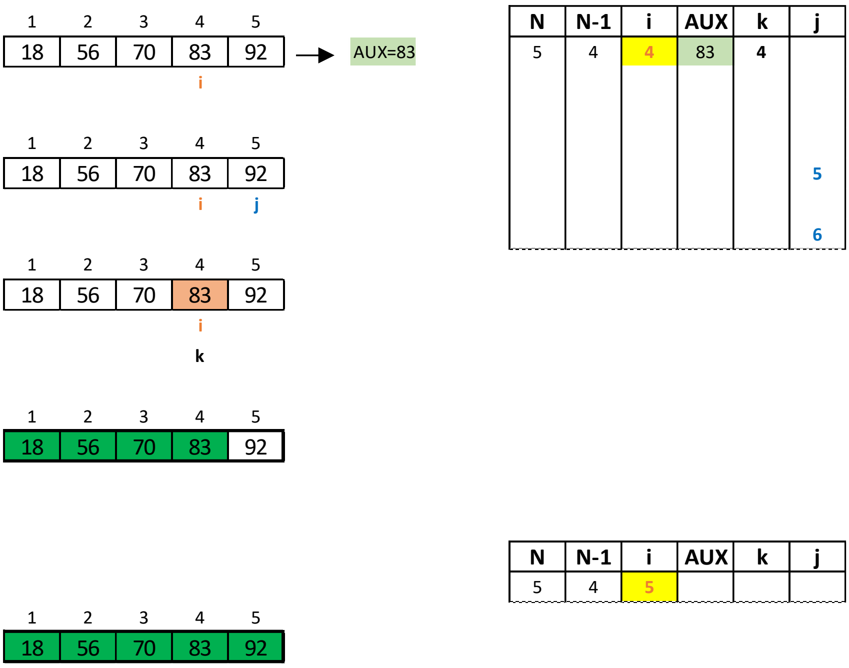
Fuente: elaboración propia.

Figura 15: Iteración externa número 3 (i=3)



Fuente: elaboración propia.

Figura 16: Iteración externa número 4 (i=4)



Fuente: elaboración propia.

$$
\begin{aligned}
\textit{Cantidad de comparaciones} &= (N^2 - N) / 2 \\
&= (5^2 - 5) / 2 \\
&= (25 - 5) / 2 \\
&= 20 / 2 \\
&= 10
\end{aligned}
$$