

	Selección directa $O(n^2)$	Inserción directa $O(n^2)$	Burbuja $O(n^2)$	Heapsort $O(n \log n)$
n=1000	0 seg	0 seg	0 seg	0 seg
n=10000	0.172 seg	0.125 seg	0.39 seg	0 seg
n=50000	4.243 seg	2.793 seg	9.641 seg	0.016 seg
n=100000	17.2 seg	12.5 seg	39 seg	0.032 seg
n=500000	430 seg	279.3 seg	964.1 seg	0.16 seg
n=1000000	1720 seg	1250 seg	3900 seg	0.32 seg

Cálculo para los métodos de selección directa, inserción directa y burbuja: $O(n^2)$

- Para n=10 000 el de selección directa lo hace en 0.172 seg. ¿para n=100 000 (son 10 veces más de datos en $O(n^2)=10^2$?

$$t = \frac{(10^5)^2 * 0.172}{(10^4)^2}$$

$$t = 10^2 * 0.172$$

$$t = 17.2 \text{ seg}$$

- Para n=10 000 el de selección directa lo hace en 0.172 seg. ¿para n=1 000 000 (son 100 veces más de datos en $O(n^2)=(10^2)^2 = 10^4$?

$$t = 10^4 * 0.172$$

$$t = 1720 \text{ seg}$$

- Para n=10 000 el de selección directa lo hace en 0.172 seg. ¿para n=500 000 (son 5*10 veces más de datos en $O(n^2)=(5*10)^2$?

$$t = (5 * 10)^2 * 0.172$$

$$t = 430 \text{ seg}$$

//Se hace lo mismo para el tiempo identificado en el resto de algoritmos de intercambio (selección directa, inserción directa e intercambio directo -burbuja-), todos ellos con complejidad $O(n^2)$.

Cálculo para Heapsort (el de complejidad $O(n \log n)$)

- Para $n=50\,000$, heapsort lo hace en 0.016 seg. ¿para $n=500\,000$ (son 10 veces más de datos en $O(n \log n)=10 * \log(10)=10$)?

$$t = 10 * 0.016$$

$$t = 0.16 \text{ seg}$$

- Para $n=50\,000$, heapsort lo hace en 0.016 seg. ¿para $n=100\,000$ (son 2 veces más de datos en $O(n \log n)=2 * \log(2)=2$)?

$$t = 2 * 0.016$$

$$t = 0.032 \text{ seg}$$

- Para $n=50\,000$, heapsort lo hace en 0.016 seg. ¿para $n=1\,000\,000$ (son 20 veces más de datos en $O(n \log n)=20 * \log(20)=20$)?

$$t = 20 * 0.016$$

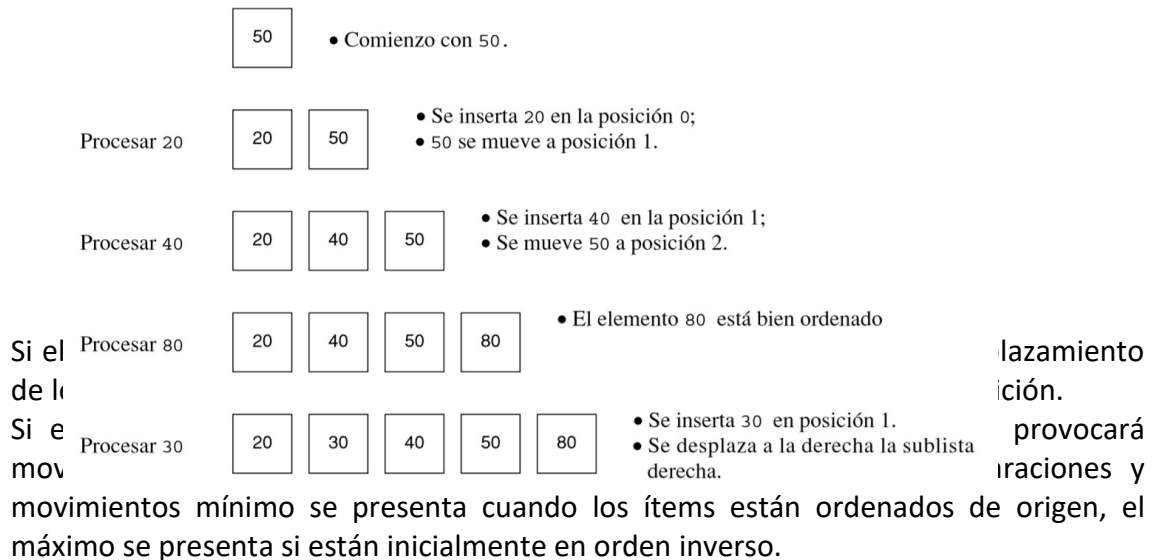
$$t = 0.32 \text{ seg}$$

Método de inserción directa:

Considera en cada paso solamente un único artículo de la secuencia origen y todos los del array de destino (cada vez mayor) para encontrar el punto de inserción.

Cartas de una baraja, lista de enteros:

A = 50, 20, 40, 80, 30



¿Cómo se puede mejorar?

El algoritmo de inserción directa se mejora fácilmente si se observa que la secuencia de destino a_1, \dots, a_{i-1} en la que debe insertarse el nuevo artículo está a ordenada. A tal fin puede utilizarse un método más rápido para determinar el lugar de inserción. Obviamente, **este método es el de búsqueda binaria que arranca del elemento central de la secuencia de destino y continúa por bisección hasta encontrar el punto de inserción.**

Método de selección directa:

Considera en cada paso todos los elementos del array de destino (cada vez menos) para encontrar el que tiene la menor clave y depositarlo como ítem siguiente de la secuencia de destino.

Selecciono el valor mínimo
Intercambio con el primero a_1
Repetir con los ítems $n-1$, $n-2$, etc.

En general, el algoritmo de selección directa es más rápido que el de inserción apenas tiene que tratar elementos salvo inserciones y por tanto sería algo más rápido.

A[0] A[1] A[2] A[3] A[4]

51	21	39	80	36
----	----	----	----	----

Pasada 0: Seleccionar 21
Intercambiar 21 y A[0]

pasada 0

21	51	39	80	36
----	----	----	----	----

Pasada 1: Seleccionar 36
Intercambiar 36 y A[1]

pasada 1

21	36	39	80	51
----	----	----	----	----

Pasada 2: Seleccionar 39
Intercambiar 39 y A[2]

pasada 2

21	36	39	80	51
----	----	----	----	----

Pasada 3: Seleccionar 51
Intercambiar 51 y A[3]

pasada 3

21	36	39	51	80
----	----	----	----	----

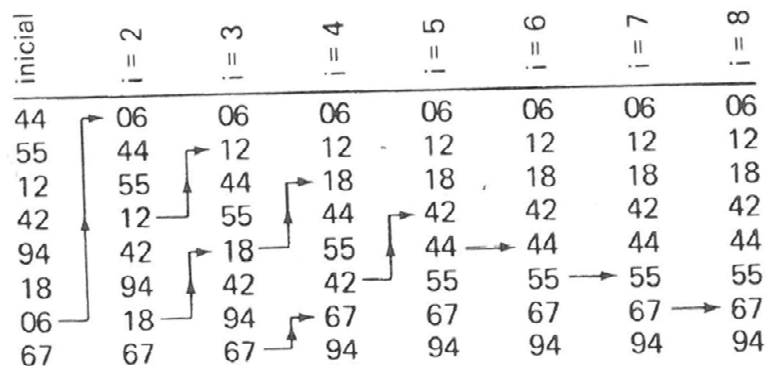
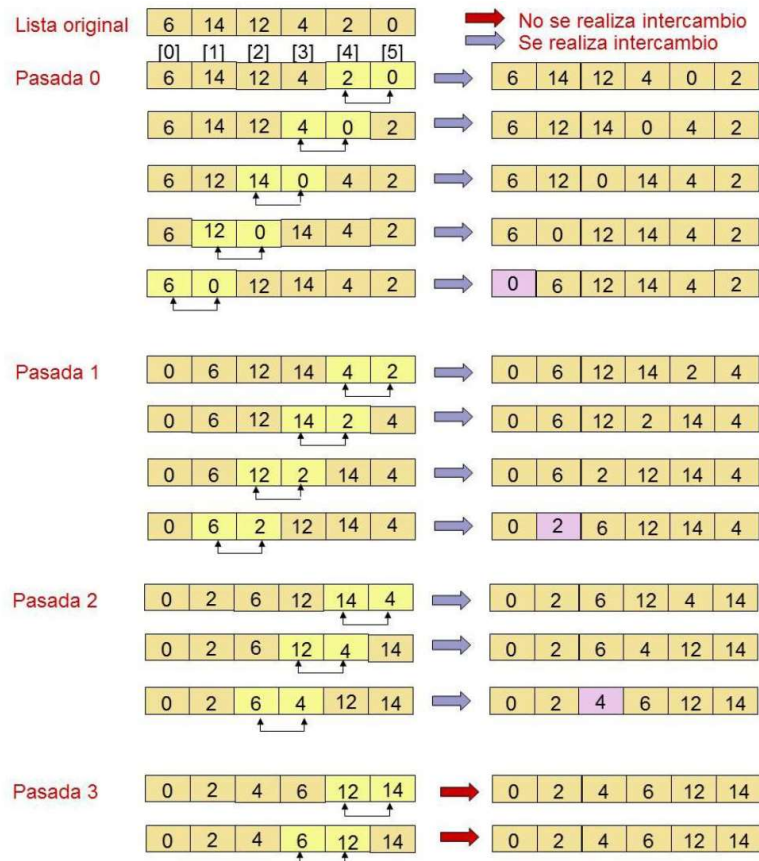
Lista ordenada

Método de la burbuja o de intercambio directo.

Considera el intercambio entre pares de ítems, en el principio de comparar e intercambiar pares de ítems adyacentes hasta que todos estén ordenados. Como ocurre en los métodos anteriores, aquí se hacen repetidas pasadas sobre el array, moviendo en cada una el elemento de clave mínima hasta el extremo izquierdo del array.

Ejemplo:

Se llama método de la burbuja porque si se mira el array como si estuviera en posición vertical en vez de horizontal, se puede observar como los ítems se consideran burbujas en un depósito de agua con "pesos" acordes con sus claves. De cada pasada sobre el array resulta la ascensión de una burbuja hasta el nivel de peso que le corresponde (ver imagen siguiente).



Este algoritmo es mejorable ya que, como puede observarse en el ejemplo anterior, en las últimas pasadas, no se requiere intercambio en los elementos por estar ya ordenados.

Otro ejemplo:

V = [12, 18, 42, 44, 55, 67, 94, 06] sólo requiere una pasada para su ordenación

W = [94, 06, 12, 18, 42, 44, 55, 67] requerirá 7 pasadas

Existen mejoras como el método de la sacudida, que no veremos en este curso.

Método de ordenación según un árbol

<https://www.youtube.com/watch?v=bj-H47-puSU&feature=share>