

T16 Métodos de ordenación internos

La ordenación o clasificación de datos (*sort* en inglés) es una operación consistente en disponer un conjunto estructura de datos en algún determinado orden con respecto a uno de los campos de elementos del conjunto. Ordenar significa reagrupar o reorganizar un conjunto de datos u objetos en una secuencia específica. Los elementos numéricos se pueden ordenar en orden creciente o decreciente de acuerdo al valor numérico del elemento. En terminología de ordenación, el elemento por el cual está ordenado un conjunto de datos (o se está buscando) se denomina clave. Una colección de datos (estructura) puede ser almacenada en un archivo, un array (vector atabla), un array de registros, una lista enlazada o un árbol. Cuando los datos están almacenados en un array, una lista enlazada o un árbol, se denomina ordenación interna. Si los datos están almacenados en un archivo, el proceso de ordenación se llama ordenación externa.

Los métodos de ordenación se suelen dividir en dos grandes grupos:

- **directos:** burbuja, selección, inserción.
- **indirectos (avanzados):** Shell, ordenación rápida, ordenación por mezcla.

Método burbuja: La ordenación por burbuja se basa en comparar elementos adyacentes de la lista (vector) e intercambiar sus valores si están desordenados. De este modo se dice que los valores más pequeños burbujan hacia la parte superior de la lista (hacia el primer elemento), mientras que los valores más grandes se hunden hacia el fondo de la lista.

Puede trabajar de dos maneras diferentes: llevando los elementos más pequeños hacia la parte izquierda del arreglo o los elementos más grandes a la parte derecha. La idea básica de este algoritmo consiste en comparar pares de elementos adyacentes e intercambiarlos entre sí hasta que todos se encuentren ordenados. Se realizan $n-1$ pasadas transportando en cada una de ellas el menor o mayor de elementos a su posición ideal.

A: 15 67 08 16 44 27 12 35

Las comparaciones que se realizan son:

PRIMERA PASADA

A[7] > A[8]	(12 > 35)	no hay intercambio
A[6] > A[7]	(27 > 12)	sí hay intercambio
A[5] > A[6]	(44 > 12)	sí hay intercambio
A[4] > A[5]	(16 > 12)	sí hay intercambio
A[3] > A[4]	(08 > 12)	no hay intercambio
A[2] > A[3]	(67 > 08)	sí hay intercambio
A[1] > A[2]	(15 > 08)	sí hay intercambio

Luego de la primera pasada el arreglo queda así:

A: 08 15 67 12 16 44 27 35

Observe que el elemento más pequeño, en este caso 08, fue situado en la parte izquierda del arreglo.

SEGUNDA PASADA

A[7] > A[8]	(27 > 35)	no hay intercambio
A[6] > A[7]	(44 > 27)	sí hay intercambio
A[5] > A[6]	(16 > 27)	no hay intercambio

A[4] > A[5]	(12 > 16)	no hay intercambio
A[3] > A[4]	(67 > 12)	sí hay intercambio
A[2] > A[3]	(15 > 12)	sí hay intercambio

Luego de la segunda pasada el arreglo queda así:

A: 08 12 15 67 16 27 44 35

y el segundo elemento más pequeño del arreglo, en este caso 12, fue situado en la segunda posición.

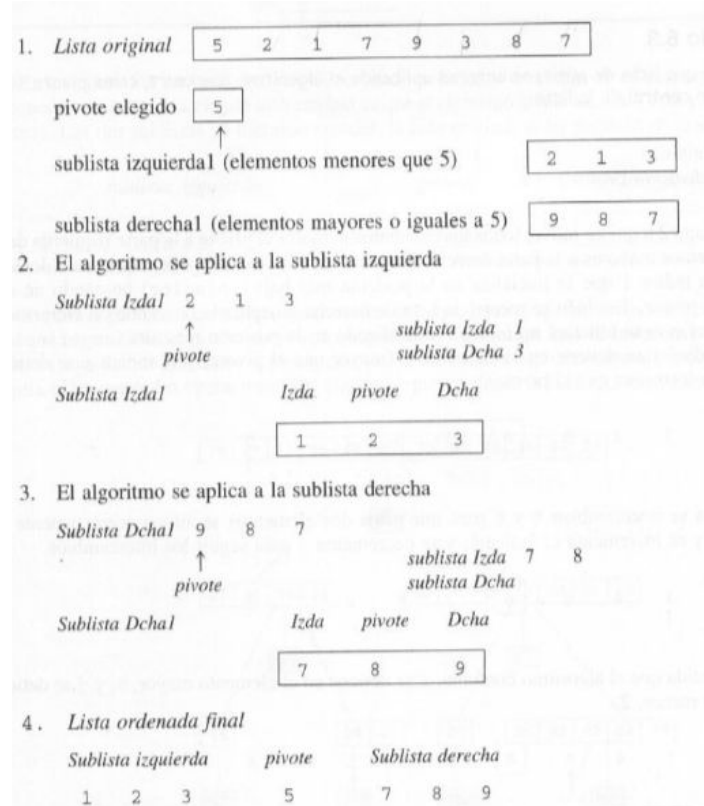
En la tabla 8.1 se presenta el resultado de las pasadas restantes.

3a. pasada:	08	12	15	16	67	27	35	44
4a. pasada:	08	12	15	16	27	67	35	44
5a. pasada:	08	12	15	16	27	35	67	44
6a. pasada:	08	12	15	16	27	35	44	67
7a. pasada:	08	12	15	16	27	35	44	67

Quicksort:

También conocido como ordenamientos por partición e intercambio. La idea del algoritmo se basa en la división en particiones de la lista a ordenar, por lo que se puede considerar que aplica la técnica divide y vencerás. El método se basa en dividir los n elementos de la lista a ordenar en dos partes o particiones separadas por un elemento: una partición izquierda, un elemento central denominado pivote, y una partición derecha. La partición se hace de tal forma que todos los elementos de la primera sublista (izquierdo) son menores que todos los elementos de la segunda sublista (derecha). Las dos sublistas se ordenan entonces independientemente.

Para dividir la lista en sublistas se elige uno de los elementos de la lista y se utiliza como pivote. Una vez elegido, se utiliza para ordenar el resto de la lista en dos sublistas: una tiene todas las claves menores que el pivote, y la otra, todas las claves mayores o iguales. Estas dos listas se ordenan recursivamente utilizando el mismo algoritmo. La primera etapa es la división recursiva de la lista hasta que todas las sublistas constan de un solo elemento.



Shellsort:

Este método también se conoce como inserción con incrementos decrecientes. Se considera que el método Shell es una mejora de los métodos de inserción directa. En el algoritmo de inserción, cada elemento se compara con los elementos contiguos de su izquierda, uno tras otro. Si el elemento a insertar es el más pequeño hay que realizar muchas comparaciones antes de colocarlo en su lugar definitivo. Este algoritmo modifica los saltos contiguos resultantes de las comparaciones por saltos de mayor tamaño y con ello se consigue que la ordenación sea más rápida. Generalmente se toma como salto inicial $n/2$ (siendo n el número de elementos), luego se reduce el salto a la mitad en cada repetición hasta que el salto es de tamaño 1.

Los pasos a seguir son:

1. Dividir la lista original en $n/2$ grupos de dos, considerando el incremento o salto entre los elementos $n/2$.
2. Clarificar cada grupo por separado, comparando las parejas de elementos, y si no están ordenados, se intercambian.
3. Se divide ahora la lista en la mitad de grupos ($n/4$), con un incremento o salto entre los elementos también mitad ($n/4$), y nuevamente se clasifica cada grupo por separado.
4. Así sucesivamente, se sigue dividiendo la lista en la mitad de grupos que en el recorrido anterior con un incremento o salto decreciente en la mitad que el salto anterior, y luego clasificado cada grupo por separado.
5. El algoritmo termina cuando se consigue que el tamaño del salto es 1.

Radix:

Este método puede considerarse como una generalización de la clasificación por urnas. Aprovecha la estrategia de la forma más antigua de clasificación manual, consiste en hacer diversos montones de fichas, cada uno caracterizado por tener sus componentes un mismo dígito en la misma posición; estos montones se recogen en orden ascendente y se reparte de nuevo en montones según el siguiente dígito de la clave. La idea clave de la ordenación Radix (por residuos) es clasificar por urnas primero respecto al dígito menos significativo, después concatenar las urnas, clasificar de nuevo respecto al siguiente dígito y así sucesivamente se sigue con el siguiente dígito hasta alcanzar el dígito más significativo, en ese momento la secuencia estará ordenada. La concatenación de las urnas consiste en enlazar el final de una con el frente de la siguiente.

Bibliografía:

- Joyanes A. L., Zahonero M. I. (2004) *Algoritmos y estructuras datos. Una perspectiva en C*. España: McGraw-Hill
- Cairó, O., Guardati, S. (2006). *Estructuras de Datos* (3ª edición). México: McGraw-Hill.