

Métodos de ordenamiento interno

En un vector es necesario, con frecuencia, clasificar u ordenar sus elementos en un orden particular. Por ejemplo, clasificar un conjunto de números en orden creciente o una lista de nombres por orden alfabético. La clasificación es una operación tan frecuente en programas de computadora que una gran cantidad de algoritmos se han diseñado para clasificar listas de elementos con eficacia y rapidez.

La elección de un determinado algoritmo depende del tamaño del vector o array (arreglo) a clasificar, el tipo de datos y la cantidad de memoria disponible. La ordenación o clasificación es el proceso de organizar datos en algún orden o secuencia específica, tal como creciente o decreciente para datos numéricos o alfabéticamente para datos de caracteres.

Los métodos de ordenación se dividen en dos categorías:

- Ordenación de vectores, tablas (arrays o arreglos).
- Ordenación de archivos.

La ordenación de arrays se denomina también ordenación interna, ya que se almacena en la memoria interna de la computadora de gran velocidad y acceso aleatorio. La ordenación de archivos se suele hacer casi siempre sobre soportes de almacenamiento externo, discos, cintas, etc., y, por ello, se denomina también ordenación externa. Estos dispositivos son más lentos en las operaciones de entrada/salida, pero, por el contrario, pueden contener mayor cantidad de información.

- Ordenación interna: clasificación de los valores de un vector según un orden en memoria central: rápida.
- Ordenación externa: clasificación de los registros de un archivo situado en un soporte externo: menos rápido.

Los métodos de clasificación se explicarán aplicados a vectores (arrays unidimensionales), pero se pueden extender a matrices o tablas (arrays o arreglos bidimensionales), considerando la ordenación respecto a una fila o columna.

Método de intercambio o de burbuja

El algoritmo de clasificación de intercambio o de la burbuja se basa en el principio de comparar pares de elementos adyacentes e intercambiarlos entre sí hasta que estén todos ordenados. Supongamos que se desea clasificar en orden ascendente el vector o lista:

50	15	56	14	35	1	12	9
A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]

Los pasos a dar son:

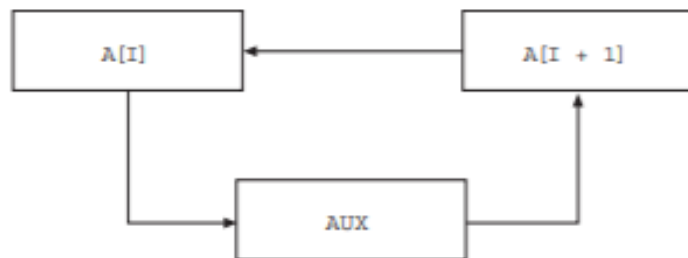
1. Comparar A[1] y A[2]; si están en orden, se mantienen como están, en caso contrario se intercambian entre sí.
2. A continuación se comparan los elementos 2 y 3; de nuevo se intercambian si es necesario.
3. El proceso continúa hasta que cada elemento del vector ha sido comparado con sus elementos adyacentes y se han realizado los intercambios necesarios.

La acción intercambiar entre sí los valores de dos elementos A[I], A[I+1] es una acción compuesta que contiene las siguientes acciones, considerando una variable auxiliar AUX.

$AUX \leftarrow A[I]$

$A[I] \leftarrow A[I+1]$

$A[I+1] \leftarrow AUX$



En realidad, el proceso gráfico es:

El elemento cuyo valor es mayor sube posición a posición hacia el final de la lista, al igual que las burbujas de aire en un depósito o botella de agua. Tras realizar un recorrido completo por todo el vector, el elemento mencionado habrá subido en la lista y ocupará la última posición. En el segundo recorrido, el segundo elemento llegará a la penúltima, y así sucesivamente.

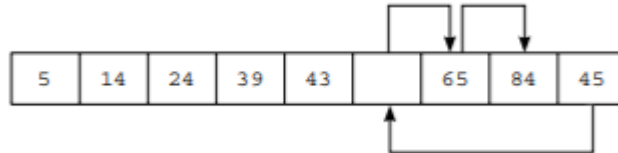
Ordenación por inserción

Este método consiste en insertar un elemento en el vector en una parte ya ordenada de este vector y comenzar de nuevo con los elementos restantes. Por ser utilizado generalmente por los jugadores de cartas se le conoce también por el nombre de método de la baraja.

Así, por ejemplo, suponga que tiene la lista desordenada

5	14	24	39	43	65	84	45
---	----	----	----	----	----	----	----

Para insertar el elemento 45, habrá que insertarlo entre 43 y 65, lo que supone desplazar a la derecha todos aquellos números de valor superior a 45, es decir, saltar sobre 65 y 84.



El método se basa en comparaciones y desplazamientos sucesivos. El algoritmo de clasificación de un vector X para N elementos se realiza con un recorrido de todo el vector y la inserción del elemento correspondiente en el lugar adecuado. El recorrido se realiza desde el segundo elemento al n -ésimo.

Método de Shell

Es una mejora del método de inserción directa que se utiliza cuando el número de elementos a ordenar es grande. El método se denomina “Shell” —en honor de su inventor Donald Shell— y también método de inserción con incrementos decrecientes.

En el método de clasificación por inserción cada elemento se compara con los elementos contiguos de su izquierda, uno tras otro. Si el elemento a insertar es más pequeño —por ejemplo—, hay que ejecutar muchas comparaciones antes de colocarlo en su lugar definitivamente.

Shell modificó los saltos contiguos resultantes de las comparaciones por saltos de mayor tamaño y con eso se conseguía la clasificación más rápida. El método se basa en fijar el tamaño de los saltos constantes, pero de más de una posición.

Supongamos un vector de elementos

4 12 16 24 36 3

en el método de inserción directa, los saltos se hacen de una posición en una posición y se necesitarán cinco comparaciones. En el método de Shell, si los saltos son de dos posiciones, se realizan tres comparaciones.

4 12 16 24 36 3

El método se basa en tomar como salto $N/2$ (siendo N el número de elementos) y luego se va reduciendo a la mitad en cada repetición hasta que el salto o distancia vale 1.

Considerando la variable salto, se tendría para el caso de un determinado vector X los siguientes recorridos:

Vector X $[X[1], X[2], X[3], \dots, X[N]]$

Vector $X1$ $[X[1], X[1+\text{salto}], X[2+\text{salto}], \dots]$

Vector XN $[\text{salto1}, \text{salto2}, \text{salto3}, \dots]$

Método QuickSort

El método de ordenación rápida (quicksort) para ordenar o clasificar un vector o lista de elementos (array) se basa en el hecho de que es más rápido y fácil de ordenar dos listas pequeñas que una lista grande. Se denomina método de ordenación rápida porque, en general, puede ordenar una lista de datos mucho más rápidamente que cualquiera de los métodos de ordenación ya estudiados. Este método se debe a Hoare.

El método se basa en la estrategia típica de “divide y vencerás” (divide and conquer). La lista a clasificar almacenada en un vector o array se divide (parte) en dos sublistas: una con todos los valores menores o iguales a un cierto valor específico y otra con todos los valores mayores que ese valor. El valor elegido puede ser cualquier valor arbitrario del vector. En ordenación rápida se llama a este valor pivote.

El primer paso es dividir la lista original en dos sublistas o subvectores y un valor de separación. Así, el vector V se divide en tres partes:

- Subvector VI , que contiene los valores inferiores o iguales.
- El elemento de separación.
- Subvector VD , que contiene los valores superiores o iguales.

Los subvectores VI y VD no están ordenados, excepto en el caso de reducirse a un elemento.

Consideremos la lista de valores.

18 11 27 13 9 4 16

Se elige un pivote, 13. Se recorre la lista desde el extremo izquierdo y se busca un elemento mayor que 13 (se encuentra el 18). A continuación, se busca desde el extremo derecho un valor menor que 13 (se encuentra el 4).

18 11 27 13 9 4 16

Se intercambian estos dos valores y se produce la lista

4 11 27 13 9 18 16

Se sigue recorriendo el vector por la izquierda y se localiza el 27, y a continuación otro valor bajo se encuentra a la derecha (el 9). Intercambiar estos dos valores y se obtiene

4 11 9 13 27 18 16

Al intentar este proceso una vez más, se encuentra que las exploraciones de los dos extremos vienen juntos sin encontrar ningún futuro valor que esté “fuera de lugar”. En este punto se conoce que todos los valores a la derecha son mayores que todos los valores a la izquierda del pivote. Se ha realizado una partición en la lista original, que se ha quedado dividida en dos listas más pequeñas:

4 11 9 [13] 27 18 16

Ninguna de ambas listas está ordenada; sin embargo, basados en los resultados de esta primera partición, se pueden ordenar ahora las dos particiones independientemente. Esto es, si ordenamos la lista

4 11 9

en su posición, y la lista

27 18 16

de igual forma, la lista completa estará ordenada:

4 9 11 13 16 18 27

El procedimiento de ordenación supone, en primer lugar, una partición de la lista.

Bibliografía

Luis Joyanes Aguilar. (2003). Fundamentos de programación – Algoritmos, Estructuras de datos y Objetos., de Mc Graw Hill