

## Métodos de ordenamiento interno.

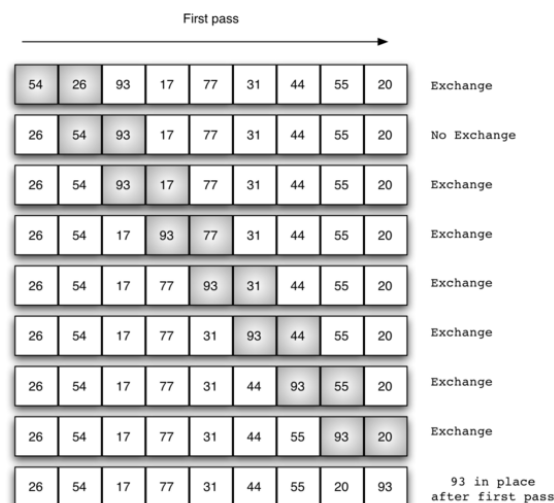
Las computadoras emplean gran parte de su tiempo en operaciones de ordenación, de búsqueda y de mezcla. Los arrays se utilizan con mucha frecuencia para almacenar datos, por ello los algoritmos para diseño de estas operaciones son fundamentales y se denominan operaciones internas, debido a que los arrays guardan sus datos de modo temporal en memoria interna y desaparecen cuando se apaga la computadora.

Dichos métodos de ordenación, búsqueda y mezcla son numerosos, por lo que abordaremos pocos.

### Ordenación por burbuja.

Este método es clásico y muy sencillo, aunque por desgracia poco eficiente. La ordenación por burbuja (bubble sort) se basa en comparar elementos adyacentes de la lista 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.

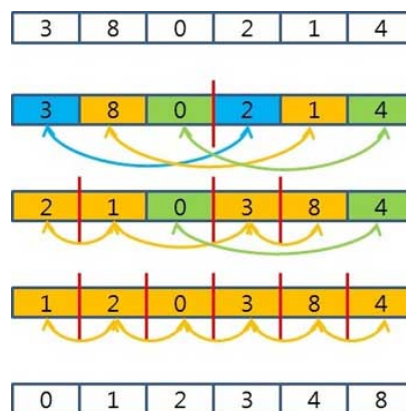
La técnica de ordenación por burbuja compara elementos consecutivos de la lista, de modo que si en una pasada no ocurrieran intercambios, significa que la lista está ordenada. El algoritmo burbuja se puede mejorar si disponemos de algún tipo de indicador que registre si se han producido intercambios en la pasada. Cuando se explore la lista y el indicador no refleja intercambios, la lista estará ya ordenada y se terminarán las comparaciones.



### Ordenación Shell (Shell sort)

La ordenación Shell debe el nombre a su inventor, D. L. Shell [CACM 2 (julio, 1959), 30-32]. Se suele denominar también ordenación por disminución de incremento (gap). La idea general del método (algoritmo) es la siguiente:

- Se divide la lista original por la mitad.
- Se clasifica cada grupo por separado (se comparan las parejas de elementos y si no están ordenado se intercambian entre sí de posiciones).
- Se divide ahora la lista en cuatro grupos (intervalo o salto de un cuarto de la lista) y nuevamente se clasifica cada grupo por separado.



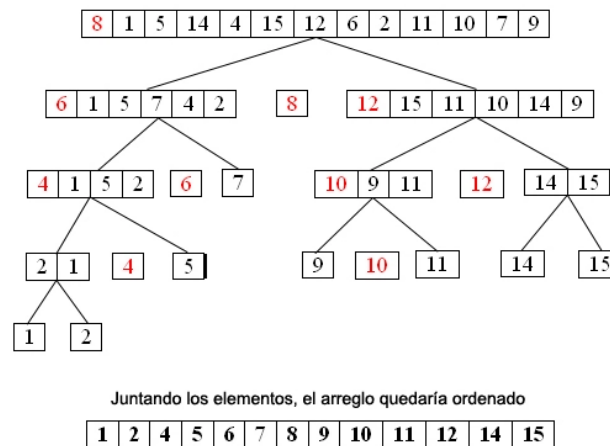
### Ordenamiento rápida (Quicksort)

Uno de los métodos más rápidos y más frecuentemente utilizando en ordenación de arrays es el conocido como ordenación rápida (Quicksort). Fue inventado por C. H. Hoare, y la cantidad de código necesario es sorprendentemente pequeño comparado con la excelente velocidad que proporciona.

La idea básica de la ordenación rápida de un array (lista) es:

- Elegir un elemento del array denominado pivote.
- Dividir o partir el array original en dos subarray o mitades (sublistas), de modo que en una de ellas estén todos los elementos menores que el pivote y en la otra sublista todos los elementos mayores que el pivote.
- Las sublistas deben ser ordenadas, independientemente, del mismo modo, lo que conduce a un algoritmo recursivo.

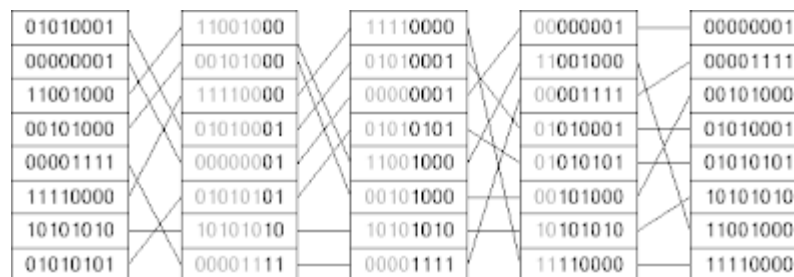
La elección del pivote es arbitraria, aunque por comodidad es usual utilizar el término central de la lista original, o bien el primero o último elemento de la misma.



## Ordenamiento Radix.

Este método se puede considerar como una generalización de la clasificación por urnas. Aprovecha la estrategia de tal forma mas antigua de clasificación manual, consistente en hacer diversos montones de fichas, cada uno caracterizado por tener sus componentes un mismo dígito (letra si es alfabética) 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 ordenacion Radix (también llamada residuos) es clasificar por urnas primero respecto al dígito de menor peso (menos significativo)  $d_k$ , después concatenar las urnas, clasificar de nuevo con respecto al siguiente dígito  $d_{k-1}$ , y así sucesivamente se sigue con el siguiente dígito más significativo  $d_1$ . en este momento la secuencia estará ordenada.



Una urna se considera una lista enlazada en la que se almacenan los elementos leídos con un determinado criterio, el valor del dígito en la secuencia creciente de pesos (unidad, decenas, etc.)

## Referencia:

Luis Joyane Aguilar, Ignacio Zahonero Martínez. (1998). Estructura de datos. España: McGraw-Hill.