

Tema: Ordenación por mezcla o fusión

La ordenación por mezcla es un prototipo de la estrategia Divide & Conquer y se puede implementar en forma recursiva y en forma iterativa.

La selección y la fusión son operaciones complementarias en el sentido de que la primera divide el arreglo en dos partes y la fusión combina esas dos partes independientes para hacer una. La relación entre ambas se hace evidente si se quiere aplicar Divide & Conquer para crear un método de ordenar un arreglo. En el caso de Quicksort se hace un proceso de partición seguido de dos llamadas recursivas. El método de **mezcla** realiza dos llamadas recursivas seguidas del proceso de fusión.

La ordenación por mezcla tiene la ventaja de que ordena un arreglo de n elementos en un tiempo $O(n \log n)$ aun en el peor caso. Su principal inconveniente es que no se puede evitar la utilización de un espacio extra proporcional a n cuando se combinan 2 arreglos ordenados.

La ordenación por mezcla se puede implementar de forma que se pueda acceder secuencialmente a los datos, lo que a veces resulta una ventaja. Es el método ideal para ordenar una lista enlazada, en la que el acceso secuencial es la única forma posible de acceso. También el método adecuado para combinar dos archivos ordenados y formar un tercer archivo ordenado.

Considere el siguiente pseudocódigo para la función **MezclarLista** que permite ordenar una lista enlazada mediante el método Merge Sort.

```
FUNCION MezclarLista(c): lista  $\rightarrow$  lista
    SI c es vacía OR c tiene un solo nodo ENTONCES
        Retorna c
    SINO
        DividirLista c en dos listas: a y b
        Retorna CombinarLista (MezclarLista(a) , MezclarLista (b) )
    FIN
```

1. Diseñe una función **DividirLista** que le permita separar una lista en dos listas de tamaño similar, *reordenando los punteros, sin eliminar los nodos de la lista original*. Calcule la complejidad usando notación O grande.
2. Diseñe la función **CombinarLista** que, dadas dos listas ordenadas, devuelva una sola lista ordenada *reordenando los punteros, sin eliminar los nodos de las listas*. Calcule la complejidad usando notación O grande.
3. Estime el tiempo de ejecución de la función **MezclarLista**.
4. Implemente en C la función **MezclarLista** y pruébela.
5. ¿Es la Ordenación por **MezclarLista** estable?
6. ¿Es la Ordenación por **MezclarLista** sensible al orden inicial de la entrada?