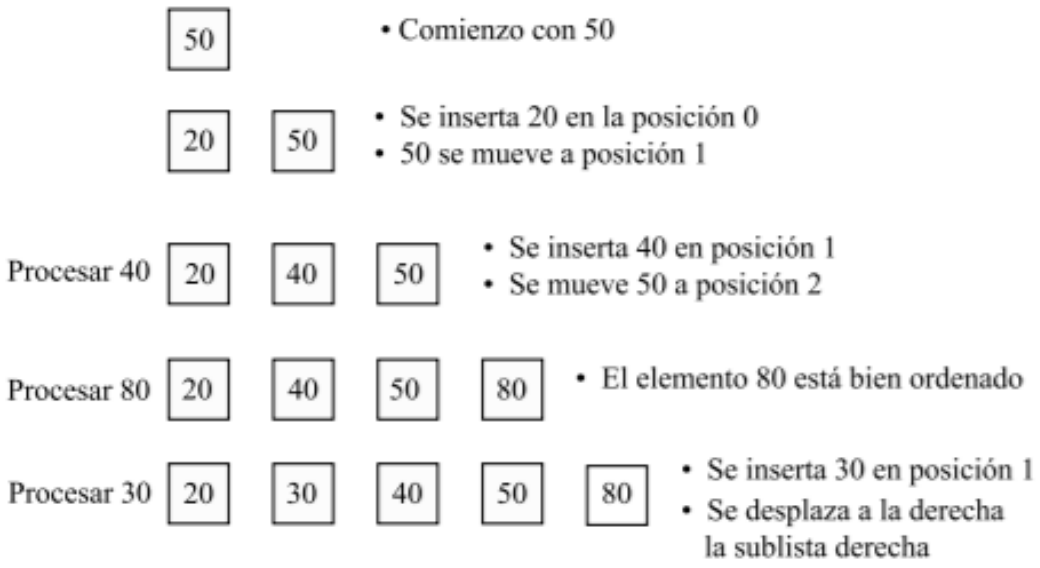
Métodos de ordenamiento externo

Ordenamiento por inserción

El método de ordenación por inserción es similar al proceso típico de ordenar tarjetas de nombres (cartas de una baraja) por orden alfabético consistente en insertar un nombre en su posición correcta dentro de una lista que ya está ordenada. El proceso en el caso de la lista de enteros es:



A la hora de analizar este algoritmo, se observa que el número de instrucciones que realiza depende del bucle automático for (bucle externo) que anida al bucle condicional while. Siendo n el número de elementos (n == a.length), el bucle externo realiza n-1 pasadas; por cada una de ellas y en el peor de los casos (aux siempre menor que a[j-1]), el bucle interno while itera un número creciente de veces que da lugar a la sucesión 1, 2, 3, ... n-1 (para i == n-1). La suma de los términos de la sucesión se ha obtenido en el apartado 6.3.2, y se ha comprobado que el término dominante es n2. Como conclusión, la complejidad del algoritmo de inserción es 0(n2).

Intercalación

La intercalación es el proceso de mezclar (intercalar) dos vectores ordenados y producir un nuevo vector ordenado. Consideremos los vectores (listas de elementos) ordenados:

A: 6 23 34

Morgado Jacome Eduardo #17211545

B: 5 22 26 27 39

El vector clasificado es:

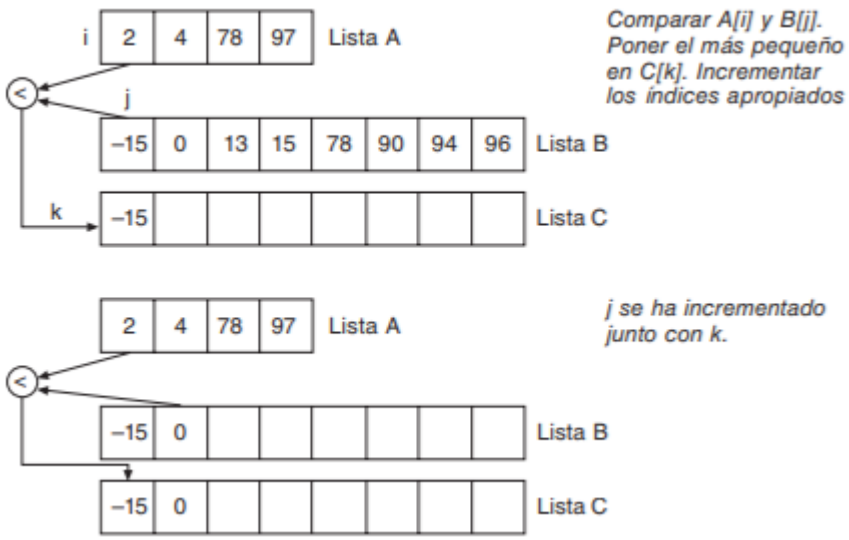
C: 5 6 22 23 24 26 27 39

La acción requerida para solucionar el problema es muy fácil de visualizar. Un algoritmo sencillo puede ser:

1. Poner todos los valores del vector A en el vector C.
2. Poner todos los valores del vector B en el vector C.
3. Clasificar el vector C.

Es decir, todos los valores se ponen en el vector C, con todos los valores de A seguidos por todos los valores de B.

Seguidamente, se clasifica el vector C. Evidentemente es una solución correcta. Sin embargo, se ignora por completo el hecho de que los vectores A y B están clasificados.



Método MergeSort

Este algoritmo consiste en partir el arreglo por la mitad, ordenar la mitad izquierda, ordenar la mitad derecha y mezclar las dos mitades ordenadas en un array ordenado.

Este último paso consiste en ir comparando pares sucesivos de elementos (uno de cada mitad) y poniendo el valor más pequeño en el siguiente hueco.

Morgado Jacome Eduardo #17211545

De todos los algoritmos de ordenamiento vistos, merge-sort es el más apropiado para ordenamiento externo, es decir, para grandes volúmenes de datos que no entran en memoria principal. Si tenemos n objetos, entonces podemos dividir a los n objetos en m bloques de b = n/m objetos cada uno. Cada bloque se almacenara en un archivo independiente.

- A diferencia del merge-sort de listas, cuando la longitud de la lista se reduce a uno, esto quiere decir que la lista tiene un solo bloque, no un solo elemento, de manera que hay que ordenar los elementos del bloque entre sı. Esto se puede hacer cargando todos los elementos del bloque en un vector y ordenándolos con algún algoritmo de ordenamiento interno.

Bibliografía

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

Luis Joyanes Aguilar, Ignacio Zahonero Martínez (2008). Estructuras de datos en Java, de Mc Graw Hill

Morgado Jacome Eduardo #17211545