Apuntes

Tema ***Ordenación de Archivos***

"Incluso la gente que afirma que no podemos hacer nada para cambiar nuestro destino, mira antes de cruzar la calle."
Stephen Hawking

Algoritmos de ordenación de Archivos

Tipos de Archivos

Un **archivo** o **fichero informático** es un conjunto de bits que son almacenados en un dispositivo. Un archivo es identificado por un nombre y la descripción de la carpeta o directorio que lo contiene. A los archivos informáticos se les llama así porque son los equivalentes digitales de los archivos escritos en expedientes, tarjetas, libretas, papel o microfichas del entorno de oficina tradicional.
En lo que concierne al sistema operativo un archivo es, en la mayoría de los casos, simplemente un flujo unidimensional de bits, que es tratado por el sistema operativo como una única unidad lógica. Un archivo de datos informático normalmente tiene un tamaño, que generalmente se expresa en bytes; en todos los sistemas operativos modernos, el tamaño puede ser cualquier número entero no negativo de bytes hasta un máximo dependiente del sistema. Depende del software que se ejecuta en la computadora el interpretar esta estructura básica como por ejemplo un programa, un texto o una imagen, basándose en su nombre y contenido. Los tipos especiales de archivos, como los nodos de dispositivo que representan simbólicamente partes del hardware, no consisten en un flujo de bits y no tienen tamaño de archivo.

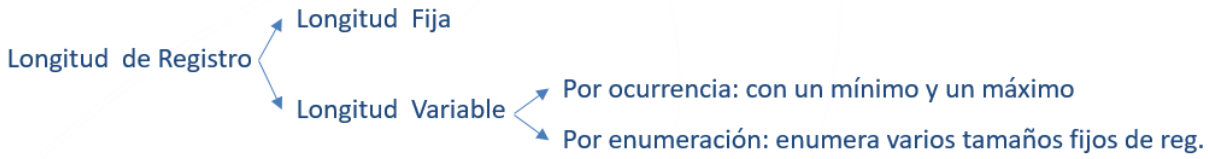
Los datos de un archivo informático normalmente consisten en paquetes más pequeños de datos *(a menudo llamados **registros** o líneas)* que son individualmente diferentes pero que comparten algún rasgo en común. Por ejemplo, un archivo de nóminas puede contener datos sobre todos los empleados de una empresa y los detalles de su nómina; cada registro del archivo de nóminas se refiere únicamente a un empleado, y todos los registros tienen la característica común de estar relacionados con las nóminas -esto es muy similar a colocar todos los datos sobre nóminas en un archivador concreto en una oficina que no tenga ninguna computadora. Un archivo de texto puede contener líneas de texto, correspondientes a líneas impresas en una hoja de papel.

La manera en que se agrupan los datos en un archivo depende completamente de la persona que diseñe el archivo. Esto ha conducido a una plétora de estructuras de archivo más o menos estandarizadas para todos los propósitos imaginables, desde los más simples a los más complejos. La mayoría de los archivos informáticos son usados por programas informáticos. Estos programas crean, modifican y borran archivos para su propio uso bajo demanda. Los programadores que crean los programas deciden qué archivos necesitan, cómo se van a usar, y (a menudo) sus nombres.

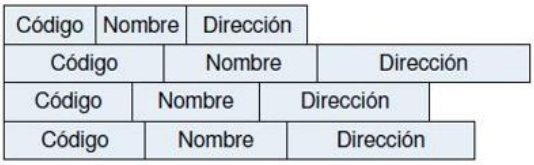
En algunos casos, los programas de computadora manipulan los archivos que se hacen visibles al usuario de la computadora. Por ejemplo, en un programa de procesamiento de texto, el usuario manipula archivos-documento a los que él mismo da nombre. El contenido del archivo-documento está organizado de una manera que el programa de procesamiento de texto entiende, pero el usuario elige el nombre y la ubicación del archivo, y proporciona la información (como palabras y texto) que se almacenará en el archivo.

Registros

Son un tipo de dato estructurado y pueden contener datos elementales o primitivos u otros datos estructurados los cuales son llamados campos.



Registros de longitud fija
Todos son iguales



Registros de longitud variable
Todos son iguales

Las claves de un registro son los campos que permiten identificar unívocamente un registro y estas pueden ser simples o compuestas. Y también Primarias y secundarias.

Tipos de Archivos

Muchas aplicaciones empaquetan todos sus archivos de datos en un único archivo, usando marcadores internos para discernir los diferentes tipo de información que contienen. Los archivos de datos usados por videojuegos como Doom y Quake son ejemplos de esto.

Los archivos de una computadora se pueden crear, mover, modificar, aumentar, reducir y borrar.

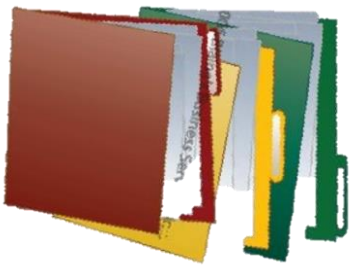
En la mayoría de los casos, los programas de computadora que se ejecutan en la computadora se encargan de estas operaciones, pero el usuario de una computadora también puede manipular los archivos si es necesario. Por ejempl o, los archivos de Microsoft Office Word son normalmente creados y modif icados por el programa Microsoft Word en respuesta a las órdenes del usuario, pero el usuario también puede mover, renombrar o borrar estos archivos directamente usando un programa gestor de archivos como Windows Explorer *(en computadoras con sistema operativo Windows)*.
También un archivo es un documento donde uno introduce algún tipo de Dato para almacenar en un objeto que lo pueda leer o modificar como una computadora.

Archivos

- Un archivo es una colección de datos en el almacenamiento masivo.
- Un archivo de datos no es una parte del código fuente de un programa.
- El mismo archivo puede ser leído o modificado por diferentes programas.
- El programa debe ser consciente del formato de los datos en el archivo.
- El sistema de archivos *(file system)* es mantenido por el sistema operativo.
- El sistema operativo proporciona comandos y / o utilidades GUI para la visualización de archivos y directorios para copiar, mover, renombrar y eliminar archivos.
- El sistema también proporciona funciones "básicas", que puede llamarse desde programas, para los directorios y archivos de lectura y escritura.

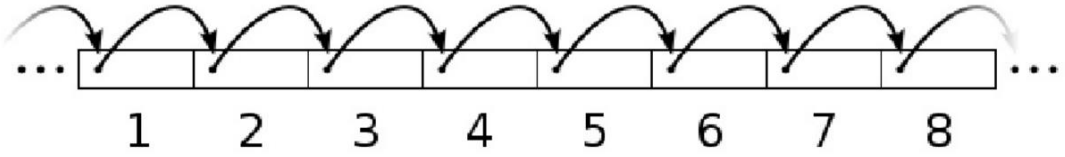
Tipos de accesos a los Archivos

Archivos Secuenciales



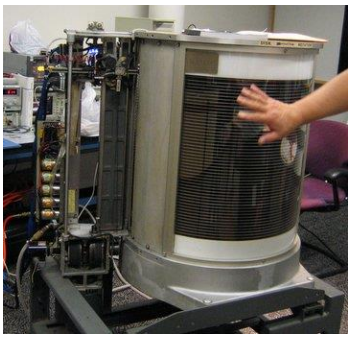
Las técnicas de archivo secuenciales proporcionan una forma sencilla de leer y escribir archivos. Los comandos de manipulación de archivos secuenciales se usan por lo general para archivos de texto básicos; siendo los caracteres **ASCII** con pares retorno de **carro/salto de línea**, los que separan los registros.

En informática, el acceso secuencial significa que un grupo de elementos *(por ejemplo, los datos en una matriz de memoria o un archivo de disco o en una cinta)* se acceden en una secuencia predeterminada, ordenada.
El acceso secuencial es a veces la única manera de acceder a los datos, por ejemplo si está en una cinta. También puede ser el método de acceso de elección, por ejemplo, si simplemente queremos procesar una secuencia de elementos de datos en orden.

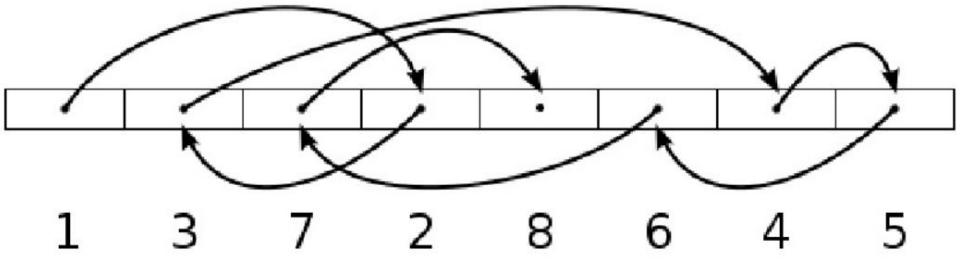


Archivos Aleatorios

Los archivos de acceso aleatorio consisten en registros que **se puede acceder en cualquier secuencia**. Esto significa que los datos se almacenan tal y como aparece en la memoria, lo que ahorra tiempo de procesamiento *(porque no hay traducción es necesario)* tanto en cuando el archivo se escribe y en cuando se lee. En informática, de acceso aleatorio *(a veces llamado acceso directo)* es la capacidad de acceder a un elemento arbitrario de una secuencia en el mismo tiempo.



- Un programa puede comenzar a leer o escribir un archivo de acceso aleatorio en cualquier lugar y leer o escribir cualquier número de bytes a la vez.
- "archivo de acceso aleatorio" es una abstracción: cualquier archivo puede ser tratado como un archivo de acceso aleatorio, pero debe cumplir con ciertos requerimientos de estructura.
- Puede abrir un archivo de acceso aleatorio, tanto para la lectura y la escritura al mismo tiempo.
- Un archivo binario que contiene los registros de datos de longitud fija es adecuado para el tratamiento de acceso aleatorio.
- Un archivo de acceso aleatorio puede ir acompañado de un "índice" (ya sea en el mismo o un archivo diferente), que cuenta la dirección de cada registro.



Ordenamiento basado en archivos

El ordenamiento externo se refiere al ordenamiento de archivos. El objetivo es reducir el número de accesos a los archivos, que son la parte más costosa de los algoritmos. En esta lectura, estudi aremos varios métodos de ordenamiento externo.

El ordenamiento externo se usa para ordenar secuencias grandes de elementos. En estos casos, los datos a ordenar no caben en memoria principal. Dicho ordenamiento está relacionado con archivos y los dispositivos en donde estén esos archivos. El tiempo de acceso al archivo es mucho mayor que el tiempo de ordenamiento en memoria, y depende notablemente del dispositivo de almacenamiento. Por ejemplo, un dispositivo secuencial, tal como una cinta, es mucho más lento que uno que permita acceso directo, tal como un disco.

Si un archivo esta formado por una secuencia de n registros, cada registro puede ser comparable si dispone de una clave K(i). En ese caso, diremos que el archivo esta ordenado respecto de la clave si

$$\forall\, i < j \Rightarrow K(i) < K(j)$$

Por ejemplo, si tenemos un archivo con registros de estudiantes, podríamos ordenarlo por varias claves, tales como DNI, número de registro, alfabéticamente, por edad, entre otros criterios. En ese caso, diremos que el archivo está ordenado por un criterio dado K, si dados cualesquiera dos registros de alumnos i y j, se cumple que si i está antes que j en el archivo, entonces la clave K para i es menor que dicha clave para j.

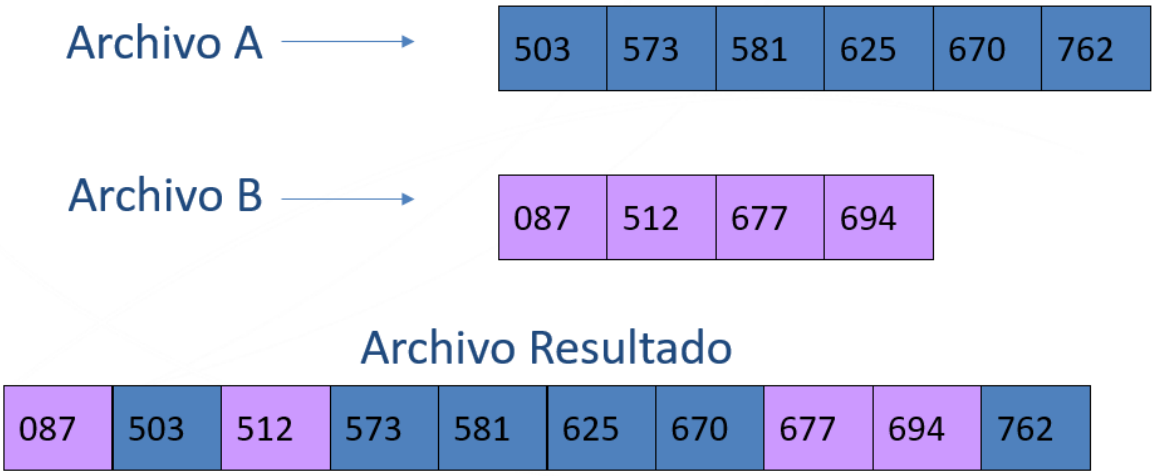
Los algoritmos de ordenamiento externo usan un esquema de separación y mezcla, donde la separación es la distribución de secuencias de registros ordenados en varios archivos, y la mezcla es la combinación de 2 o mas secuencias ordenadas en una sola secuencia ordenada. A continuación, veremos varios métodos de ordenamiento externo.

Mezcla de archivos

- Mezclar significa **combinar dos o mas archivos ordenados en un archivo simple**, algunos métodos dividen un archivo en dos partes para aplicar la mezcla.
- Suponer que existen dos archivos tales que:
  Archivo A = a0<= a1<= a2<= a3<= … <= an
  Archivo B = b0 <= b1<= b2<= b3<= … <= bk
- El archivo de resultado será:
  Archivo X = x0 <= x1<= x2<= x3<= … <= xk+n donde cada elemento xi es un elemento de A o de B

Mezcla de 2 archivos por intercalación

En este método de ordenamiento existen dos archivos con llaves ordenadas, los cuales se mezclan para formar un solo archivo. La longitud de los archivos puede ser diferente.
El proceso consiste en leer un registro de cada archivo y compararlos, el menor es almacenando en el archivo de resultado y el otro se compara con el siguiente elemento del archivo si existe. El proceso se repite hasta que alguno de los archivos quede vacío y los elementos del otro archivo se almacenan directamente en el archivo resultado.



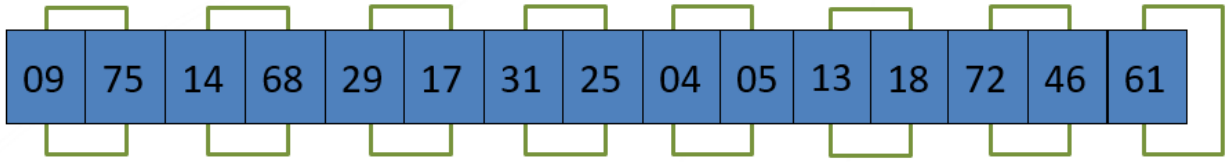
Ordenamiento por Mezcla Directa

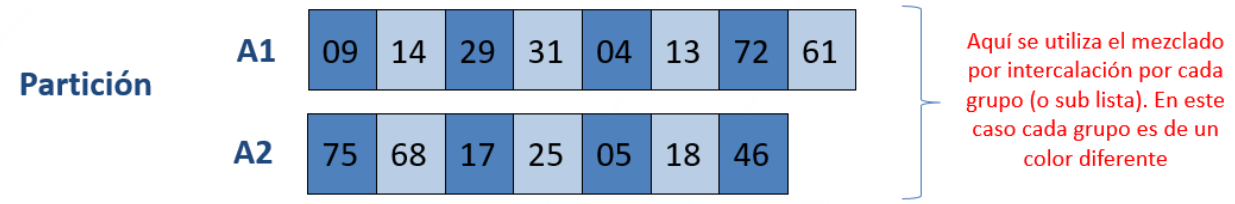
El algoritmo de **mezcla directa** es el método más simple de ordenamiento externo. Dicho algoritmo consiste en un esquema iterativo de separar secuencias de registros y su mezcla posterior.
En este esquema, se separan registros del **archivo original O** en 2 archivos **A1** y **A2**, a continuación se mezclan **A1** y **A2** combinando registros aislados y formando pares ordenados, que serán escritos en **O**. El algoritmo continúa separando ahora pares de registros de **O** en **A1** y **A2**, y efectuando un paso de mezcla de **A1** y **A2** a través de la combinación de pares de registros formando ahora cuádruplos ordenados, que serán escritos en **O**.

Como el se puede deducir, el algoritmo repite los pasos anteriores ***hasta que la longitud de la subsecuencia sea la del archivo original***.
Luego de **i** pasadas el archivo **O** tiene subsecuencias ordenadas de longitud **2i**, por lo que si el archivo tiene **n** registros, estará ordenado cuando **2i > n**. De este modo, ya que el algoritmo requiere (log n) pasadas, y cada pasada escribe n registros, podemos deducir que el número total de movimientos es **O(n log n)**.

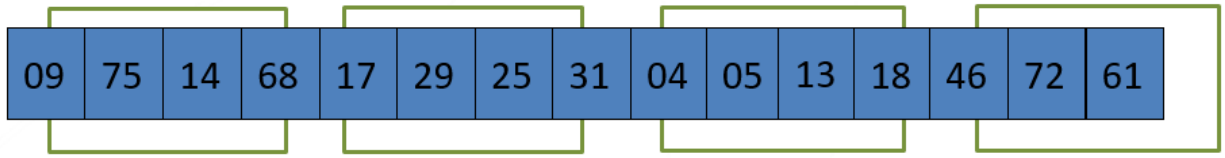
Suponemos un archivo formado por registros con clave entera

Primero tomamos de a pares

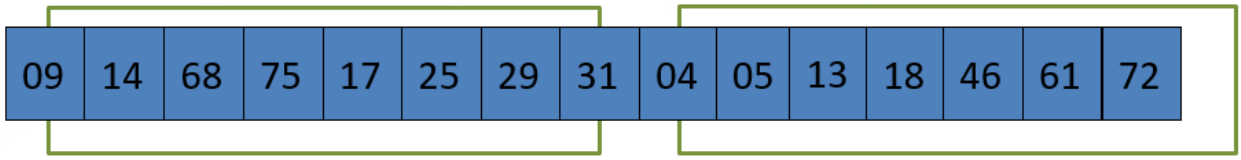


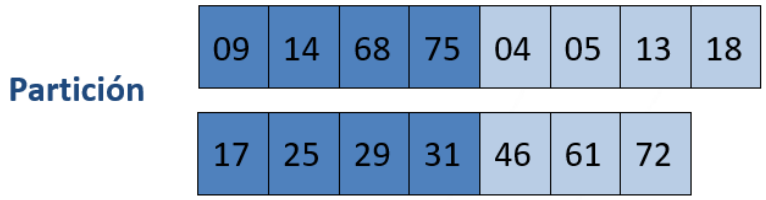


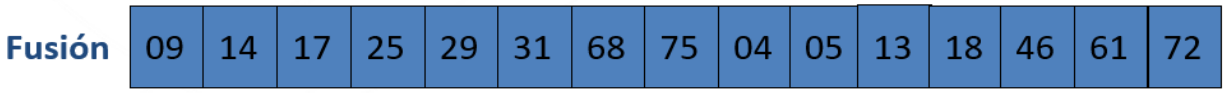
Segundo tomamos de a 4



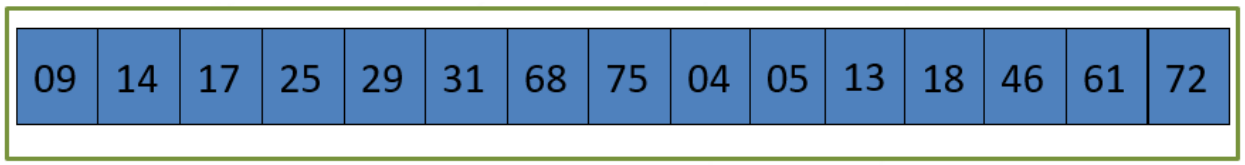
Tercero tomamos de a 8

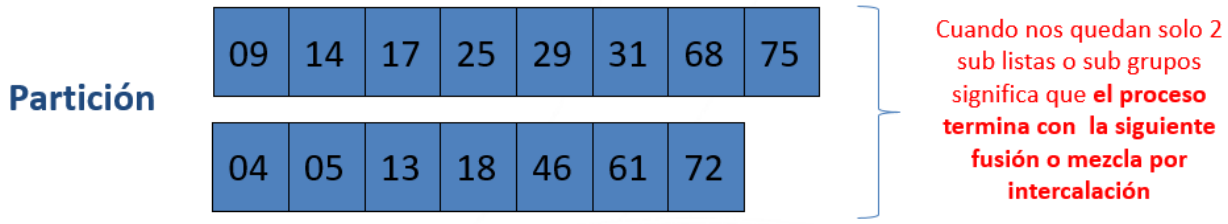


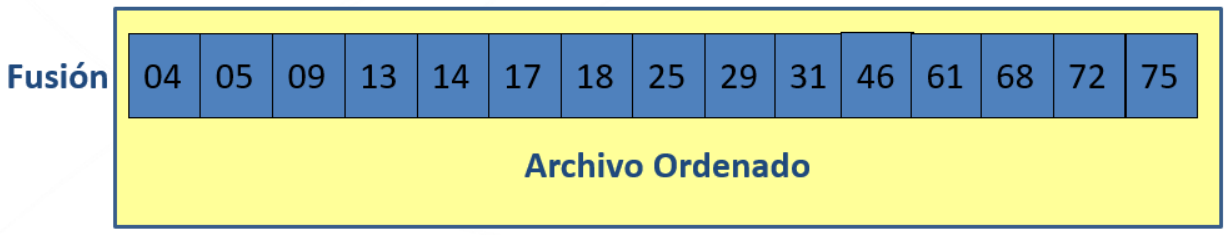




Cuarto tomamos de a 16







Ordenamiento por Mezcla Natural

El método de mezcla natural mejora el tiempo de ejecución de la mezcla directa. Para ello introduce una pequeña variación respecto a la longitud de las secuencias de registros. En la mezcla directa las secuencias de registros tienen longitudes múltiplos de dos: 1, 2, 4, 8,16…
En la mezcla natural, por el contrario, se distribuye secuencias ordenadas lo más largas posibles. Para ello, se determinan secuencias ordenadas máximas, que se denominan tramos.
Una secuencia ordenada ai…aj cumple las siguientes condiciones

* $a_k \leq a_{k+1}$ para k=i..j-1
* $a_{i-1} > a_i$
* $a_j > a_{j+1}$

Por ejemplo, dada la secuencia

4 9 11 5 8 12 9 17 18 21 26 18

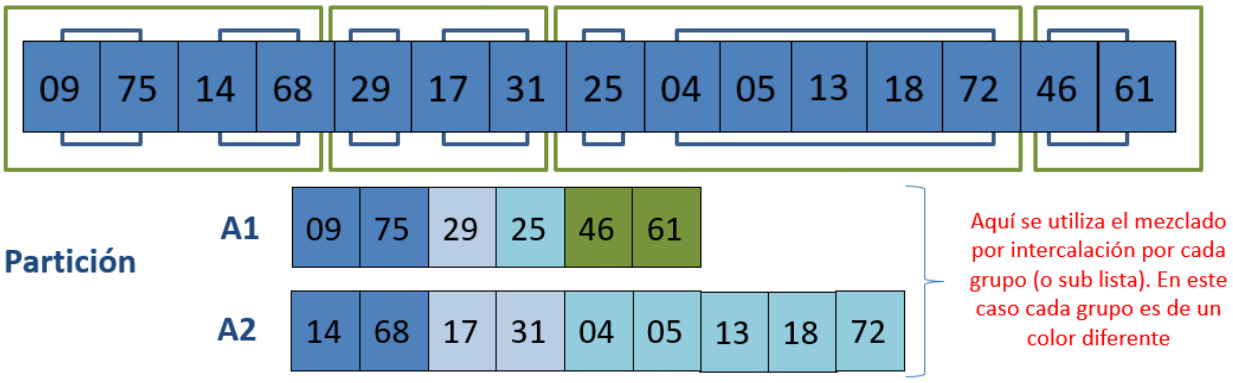
encontramos los siguientes tramos

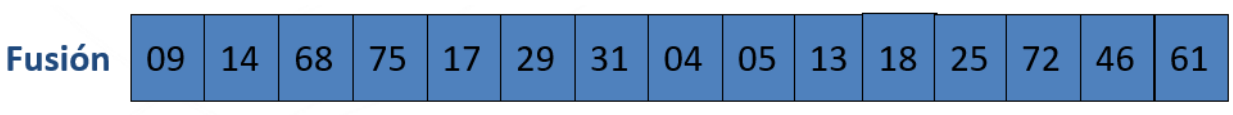
4 9 11; 5 8 12; 9 17 18 21 26; 18

La mezcla natural funde secuencias ordenadas máximas en lugar de secuencias de tamaño fijo y predeterminado. El número total de tramos disminuye en cada pasada del algoritmo de mezcla natural.

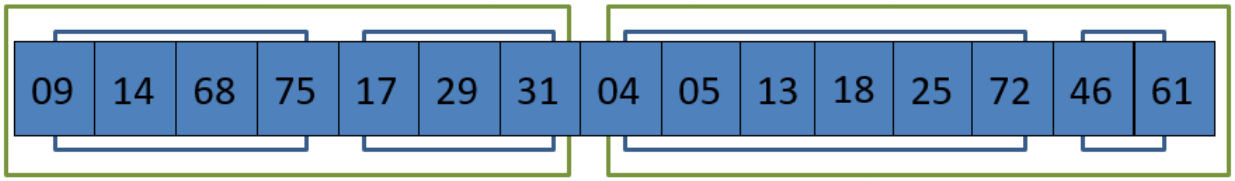
Suponemos un archivo formado por registros con clave entera

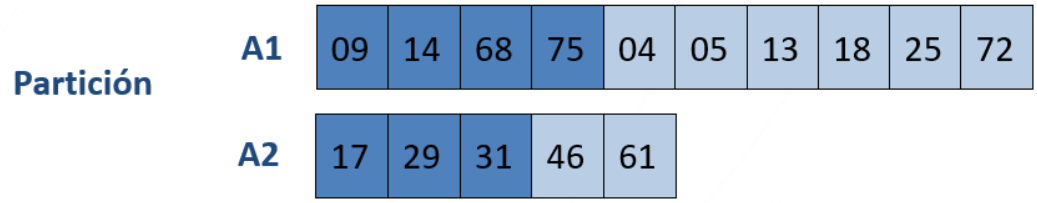
Primer pasada



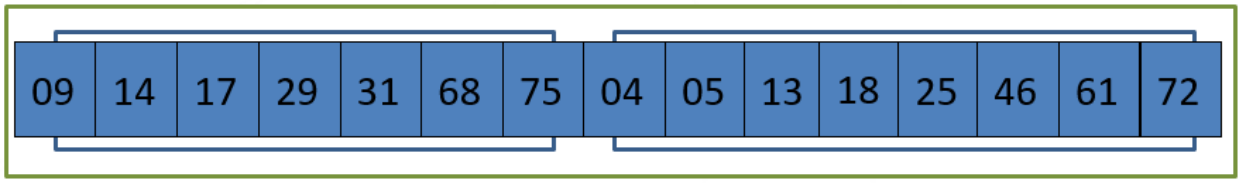


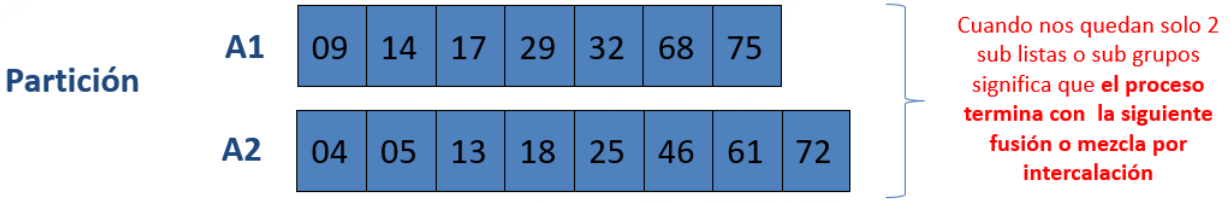
Segunda pasada

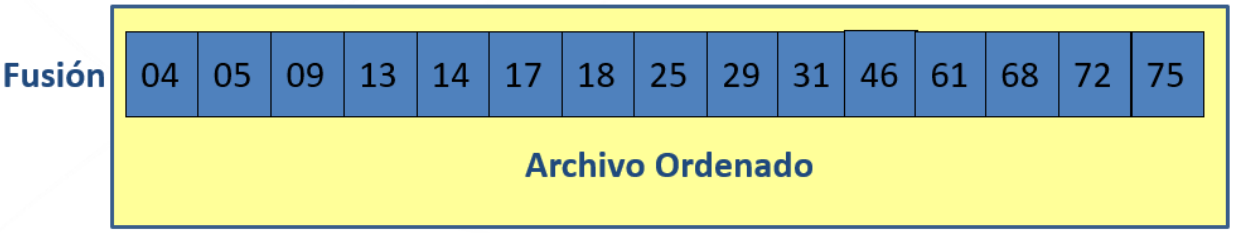




Tercera pasada





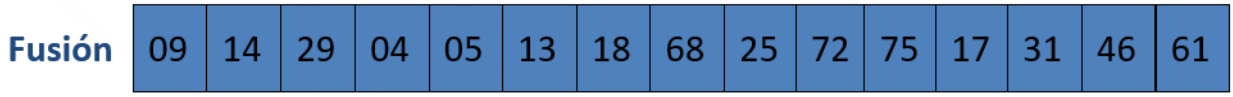


En el peor caso el número de movimientos es de orden **n log n**, e inferior en el caso promedio. El número de comparaciones es mucho mayor, pero al ser el coste de una comparación muy inferior al de un movimiento, este incremento no resulta significativo.

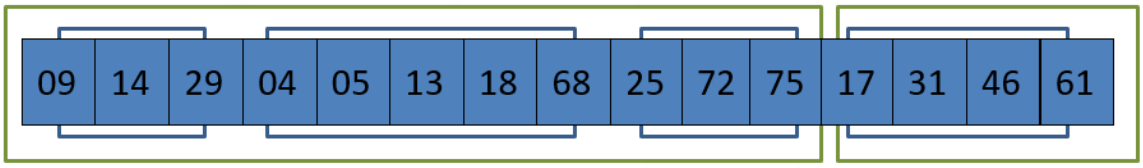
Ordenamiento por Mezcla Equilibrada Múltiple

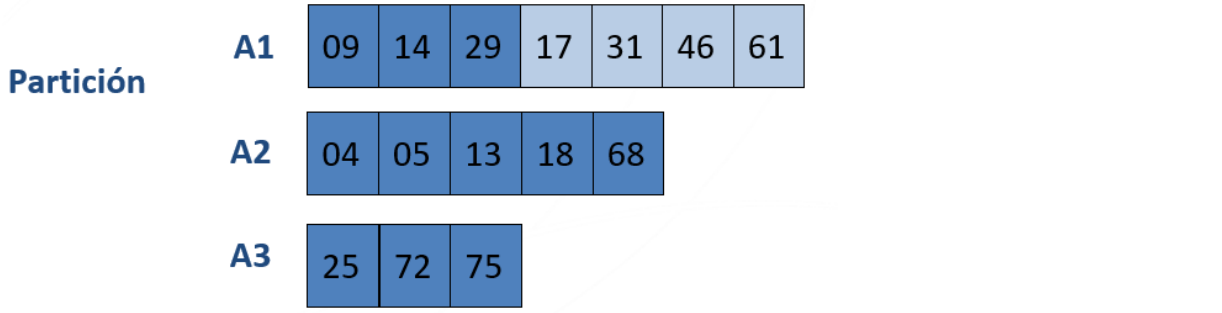
La eficiencia de los métodos de ordenamiento externo es directamente proporcional al número de pasadas, ya que cada pasada implica accesos al disco, que son muy costosos. De modo que nos interesa reducir el número de pasadas que hagamos en los algoritmos de ordenamiento externo.
Los métodos de mezcla directa y natural usan 2 archivos auxiliares. Podemos reducir el número de pasadas incrementando el número de archivos auxiliares.
Si tenemos **w tramos** distribuidos en **m archivos**, entonces en la primera pasada mezclamos **w** tramos dando lugar a $^{w}/_{m}$ tramos, en la segunda pasada mezclamos $^{w}/_{m}$ tramos dando lugar a $^{w}/_{m^2}$ tramos, y en la i-ésima pasada mezclaremos $^{w}/_{m^{i-1}}$ dando lugar a $^{w}/_{m^{i}}$ tramos.
Si suponemos que un archivo tiene **n** tramos iniciales, entonces en el peor de los casos, usando **m** archivos auxiliares, el número de pasadas necesarias para el ordenamiento completo será $(log_m n)$. Como cada pasada realiza **n** operaciones de E/S, entonces el costo total es $\mathbf{O}(n\ log_m n)$.
En los métodos de mezcla equilibrada múltiple, de los **m** archivos utilizados, $^{m}/_{2}$ son de entrada y $^{m}/_{2}$ son de salida *(a diferencia de los métodos anteriormente vistos, donde sólo 1 archivo era de salida)*.
El proceso de mezcla se hace en una fase en lugar de las dos fases utilizadas por los algoritmos previos. El algoritmo consiste en distribuir registros del archivo original por tramos en los $^{m}/_{2}$ primeros archivos auxiliares *(archivos de entrada)*, y a continuación mezclar tramos de los $^{m}/_{2}$ archivos de entrada y escribirlos consecutivamente en los $^{m}/_{2}$ archivos de salida.
Los archivos de entrada pasan a ser de salida y viceversa. Luego deberemos repetir los pasos de mezcla hasta que quede un único tramo.

Suponemos un archivo formado por registros con clave entera

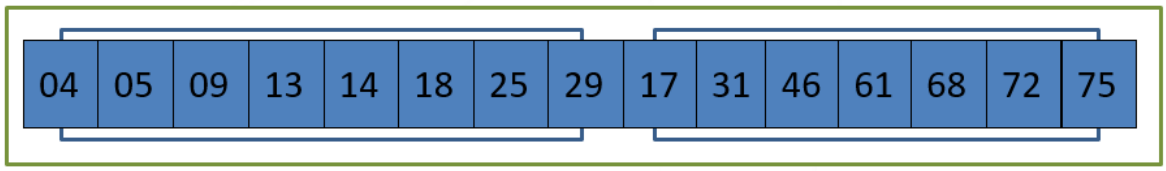


Segunda pasada





Tercera pasada



Ordenamiento por Método Polifásico

El método de mezcla equilibrada múltiple emplea **m archivos** para ordenar **n registros**. Si los **n** registros están distribuidos en **m** tramos, en una pasada quedan ordenados $^{m}/_{2}$ archivos de entrada y $^{m}/_{2}$ de salida.

El método polifásico también utiliza **m archivos** para ordenar **n registros**, pero usa **m-1** archivos de entrada y **1** de salida. En el momento en que uno de los archivos de entrada alcanza su final hay un cambio de cometido, pasa a ser considerado como archivo de salida, y el archivo que en ese momento era de salida pasa a ser de entrada y la mezcla de tramos continúa. La sucesión de pasadas continúa hasta alcanzar el archivo ordenado.

Cabe recordar la propiedad base de todos los métodos de mezcla: la mezcla de k tramos de los archivos de entrada se transforma en k tramos en el archivo de salida.

Para N menos de 8 archivos de trabajo, un tipo de fusión polifásico logra un factor de reducción de recuento de ejecución efectivo más alto al distribuir de manera desigual los ensayos ordenados entre los archivos de trabajo N-1 (explicado en la siguiente sección). Cada iteración se funde desde N-1 archivos de trabajo en un único archivo de trabajo de salida. Cuando se alcanza el final de uno de los N-1 archivos de trabajo, entonces se convierte en el nuevo archivo de salida y lo que era el archivo de salida se convierte en uno de los N-1 archivos de trabajo de entrada, comenzando una nueva iteración de polifase merge sort. Cada iteración fusiona sólo una fracción del conjunto de datos (aproximadamente 1/2 a 3/4), excepto la última iteración que combina todo el conjunto de datos en una sola ejecución ordenada. La distribución inicial se configura de manera que sólo se vacía un archivo de trabajo de entrada a la vez, excepto para la iteración de fusión final que combina N-1 ejecuciones individuales (de tamaño variable, esto se explica a continuación) de los archivos de trabajo de entrada N-1 Al archivo de salida único, lo que resulta en una sola ejecución ordenada, el conjunto de datos ordenado.

Para cada iteración polifásica, el número total de ejecuciones sigue un patrón similar a un número de Fibonacci invertido de secuencia de orden superior. Con 4 archivos y un conjunto de datos que consta de 57 ejecuciones, el recuento de ejecución total en cada iteración sería 57, 31, 17,

9, 5, 3, 1. Tenga en cuenta que, excepto en la última iteración, el factor de reducción del recuento de ejecución es un poco menor que 2, 57/31, 31/17, 17/9, 9/5, 5/3, 3/1, alrededor de 1,84 para un archivo de 4 , Pero cada iteración excepto la última reduce el recuento de ejecución mientras procesa aproximadamente el 65% del conjunto de datos, por lo que el factor de reducción de recuento de ejecución por conjunto de datos procesado durante las iteraciones intermedias es aproximadamente 1,84 / 0,65 = 2,83. Para un conjunto de datos que consta de 57 ejecuciones de 1 registro cada uno, luego, después de la distribución inicial, el tipo de fusión polifásico mueve 232 registros durante las 6 iteraciones que se tarda en ordenar el dataset, para un factor de reducción global de 2,70 .

Después de la primera iteración polifásica, cuál era el archivo de salida ahora contiene los resultados de la fusión de N-1 originales, pero los restantes archivos de trabajo de entrada N-2 todavía contienen las ejecuciones originales restantes, por lo que la segunda iteración de combinación produce series de tamaño -1) + (N - 2) = (2N - 3) recorridos originales. La tercera iteración produce carreras de tamaño (4N - 7) carreras originales. Con 4 archivos, la primera iteración crea series de tamaño 3 originales, la segunda iteración 5 originales, la tercera iteración 9 originales y así sucesivamente, siguiendo el patrón Fibonacci, 1, 3, 5, 9, 17, 31, 57, ..., por lo que el aumento en el tamaño de ejecución sigue el mismo patrón que la disminución en el conteo de marcha en sentido inverso. En el caso de ejemplo de 4 archivos y 57 ejecuciones de 1 registro cada uno, la última iteración fusiona 3 ejecuciones de tamaño 31, 17, 9, resultando en una sola ordenada de tamaño 31 + 17 + 9 = 57 registros, el conjunto de datos ordenado. Un ejemplo de los conteos de ejecución y los tamaños de ejecución para 4 archivos, 31 registros se pueden encontrar en la tabla 4.3 de [3]