Tema 5 Métodos de Ordenamiento Externos

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 = $a_0 <= a_1 <= a_2 <= a_3 <= ... <= a_n$
 - Archivo B = $b_0 \le b_1 \le b_2 \le b_3 \le ... \le b_k$
- El archivo de resultado será:
 - Archivo $X = x_0 <= x_1 <= x_2 <= x_3 <= \dots <= x_{k+n}$ donde cada elemento x_i es un elemento de A o de B

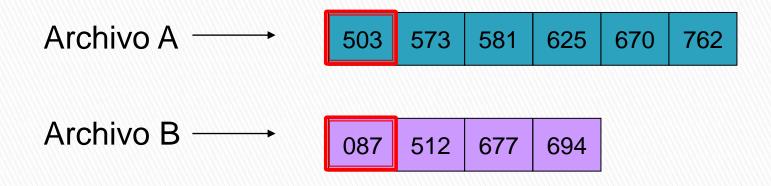
Tipos de mezclas

- Por intercalación
- Mezcla directa
- Mezcla natural o equilibrada
- Mezcla por listas

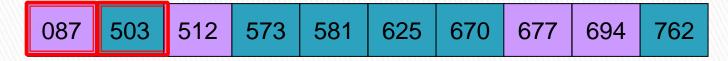
Mezcla por intercalación

- 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.

Mezcla por intercalación



Archivo resultado



Mezcla por intercalación

```
Inicio {
                                        // procesa archivo A
  abrir archivo A
                                           mientras (!eof(A)){
   abrir archivo B
                                                 almacena en X a
  abrir archivo X
  a = leer archivo A
                                                 a = leer archivo A }
   b = leer archivo B
                                        // procesa archivo B
  // procesa los dos archivos
                                           mientras (!eof(B)){
   mientras (!eof(A) && !eof(B)){
                                                 almacena en X b
       si (a < b) { almacena en X a
                                                 b = leer archivo B }
                 a = leer archivo A
                                           cerrar archivos A,B,X
  }
       sino { almacena en X b
                 b = leer archivo B
```

Intercalación

Cual es la complejidad de tiempo del algoritmo de intercalación?

Suponer:

M es el tamaño del archivo A N es el tamaño del archivo B

Los dos archivos se procesan al mismo tiempo hasta que uno de los dos llega a su fin M-N o N-M Y después se procesa el resto del archivo que no ha terminado,

y despues se procesa el resto del archivo que no ha terminado, por lo tanto la complejidad es el mayor de M y N

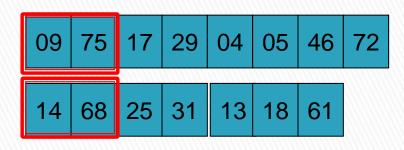
Mezcla Directa

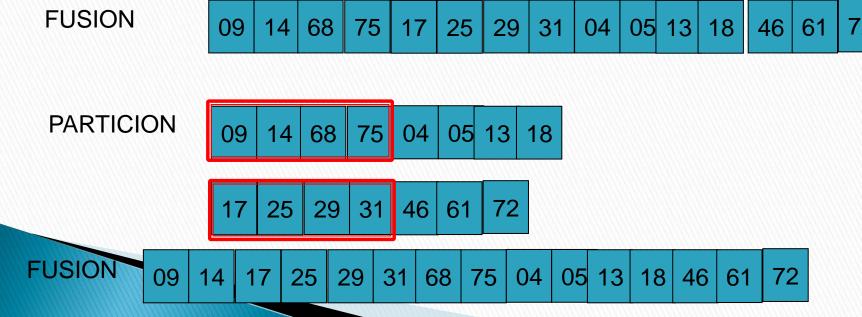
- Este método de ordenamiento realiza sucesivamente una partición y una fusión que produce secuencias ordenadas de longitud cada vez mayor.
- En la primera pasada la partición es de 1 y la fusión produce secuencias de longitud 2. Las particiones y las fusiones doblan su tamaño en cada pasada del procesamiento hasta lograr una partición del tamaño del archivo.

Mezcla directa

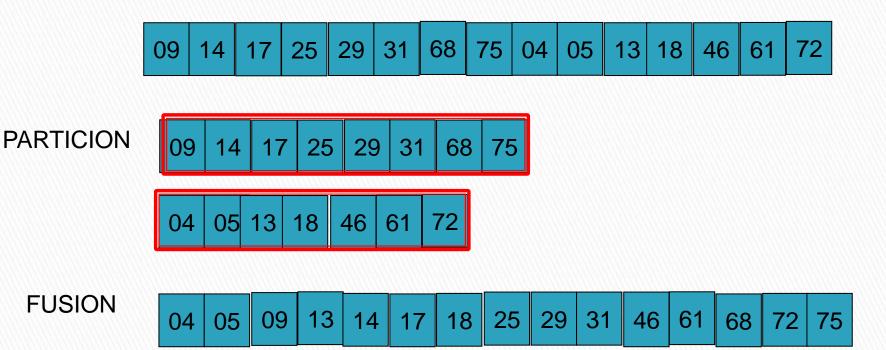


Mezcla directa





Mezcla directa



Capítulo 8



MÉTODOS DE ORDENACIÓN

Algoritmo 8.17 Mezcla_directa

Mezcla_directa (F, F1, F2, N)

{El algoritmo ordena los elementos del archivo F por el método de mezcla directa. Utiliza desarchivos auxiliares F1 y F2. N es el número de elementos del archivo F} {PART es una variable de tipo entero}

- L Hacer PART ← 1
- Mientras (PART < parte entera ((N + 1) / 2)) Repetir Llamar al algoritmo Particiona con F, F1, F2 y PART Llamar al algoritmo Fusiona con F, F1, F2 y PART Hacer PART ← PART * 2
- 3. {Fin del ciclo del paso 2}

Particiona (F, F1, F2, PART)

{El algoritmo genera dos archivos auxiliares, F1 y F2, a partir del archivo F. PART es la longitud de la partición que se va a realizar} {K, L y R son variables de tipo entero}

- Abrir el archivo F para lectura.
- Abrir los archivos F1 y F2 para escritura.
- 3. Mientras (no sea el fin de archivo de F) Repetir

Hacer $K \leftarrow 0$

3.1 Mientras ((K < PART) y (no sea el fin de archivo de F)) Repetir Leer R de F

Escribir R en F1

Hacer $K \leftarrow K + 1$

3.2 {Fin del ciclo del paso 3.1} Hacer $L \leftarrow 0$

3.3 Mientras ((L < PART) y (no sea el fin de archivo de F)) Repetir

Leer R de F

Escribir R en F2

Hacer $L \leftarrow L + 1$

- 3.4 (Fin del ciclo del paso 3.3)
- 4. {Fin del ciclo del paso 3}

Algoritmo 8.19 Fusiona

Fusiona (F, F1, F2, PART)

{El algoritmo fusiona los archivos F1 y F2 en el archivo F. PART es la longitud de la partición que se realizó previamente}

{R1, R2, K y L son variables de tipo entero. B1 y B2 son variables de tipo booleano}

- L Abrir el archivo F para escritura.
- Abrir los archivos F1 y F2 para lectura.
- 3. Hacer B1 ← VERDADERO y B2 ← VERDADERO
- 4. Si (no es el fin de archivo de F1) entonces

Leer R1 de F1

Hacer B1 ← FALSO

- 5. {Fin del condicional del paso 4}
- 6. Si (no es el fin de archivo de F2) entonces

Leer R2 de F2

Hacer B2 ← FALSO

- 7, {Fin del condicional del paso 6}
- Mientras ((no sea el fin de archivo de F1) o (B1 = FALSO)) y ((no sea el fin de archivo de F2) o (B2 = FALSO)) Repetir

Hacer $K \leftarrow 0$ y $L \leftarrow 0$

```
8.1 Mientras ((K < PART) y (B1 = FALSO)) y ((L < PART) y ((B2 = FALSO))
       Repetir
    8.1.1 Si(R1 \le R2)
           entonces
                   Escribir R1 en F
                   Hacer B1 \leftarrow VERDADERO y K \leftarrow K + 1
           8.1.1.1 Si (no es el fin de archivo de F1) entonces
                       Leer R1 de F1
                       Hacer B1 ← FALSO
          8.1.1.2 (Fin del condicional del paso 8.1.1.1)
          si no
                   Escribir R2 en F
                   Hacer B2 \leftarrow VERDADERO y L \leftarrow L + 1
           8.1.1.3 Si (no es el fin de archivo de F2) entonces
                       Leer R2 de F2
                       Hacer B2 \leftarrow FALSO
          8.1.1.4 (Fin del condicional del paso 8.1.1.3)
    8.1.2 (Fin del condicional del paso 8.1.1)
8.2 {Fin del ciclo del paso 8.1}
8.3 Mientras ((K < PART) y (B1 = FALSO)) Repetir
           Escribir R1 en F
           Hacer B1 \leftarrow VERDADERO y K \leftarrow K + 1
    8.3.1 Si (no es el fin de archivo de F1) entonces
              Leer R1 de F1
              Hacer B1 \leftarrow FALSO
    8.3.2 (Fin del condicional del paso 8.3.1)
8.4 (Fin del condicional del paso 8.3)
```

8.5 Mientras ((L < PART) y (B2 = FALSO)) Repetir Escribir R2 en F Hacer $B2 \leftarrow VERDADERO y L \leftarrow L + 1$ 8.5.1 Si (no es el fin de archivo de F2) entonces Leer R2 de F2 Hacer B2 ← FALSO 8.5.2 (Fin del condicional del paso 8.5.1) 8.6 (Fin del ciclo del paso 8.5) 9. {Fin del ciclo del paso 8} 10. Si(B1 = FALSO) entonces Escribir R1 en F (Fin del condicional del paso 10) Si (B2 = FALSO) entonces Escribir R2 en F 4 (Fin del condicional del paso 12) 14. Mientras (no sea el fin de archivo de F1) Repetir Leer R1 de F1 Escribir R1 en F 15. {Fin del condicional del paso 14} 16. Mientras (no sea el fin de archivo de F2) Repetir Leer R2 de F2 Escribir R2 en F 17. {Fin del ciclo del paso 16} 18. {Cerrar los archivos F, F1 y F2}

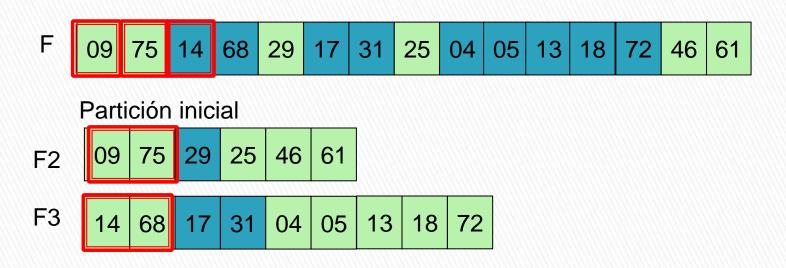
Complejidad del método mezcla directa

Un tiempo de ejecución de $O(n\log_2 n)$ para el caso promedio.

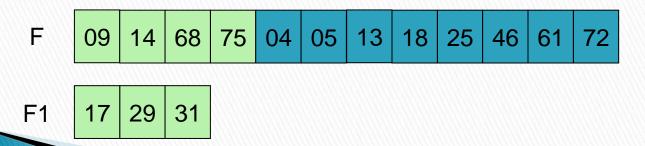
Mezcla natural

- Este método, conocido también como mezcla equilibrada, es una optimización del método de mezcla directa.
- La diferencia es que las particiones se realizan en secuencias ordenadas de tamaño variable en lugar de secuencias de tamaño fijo. La fusión de las secuencias ordenadas se realiza en dos archivos. El proceso termina cuando el segundo archivo queda vacío en el proceso de fusión-partición.

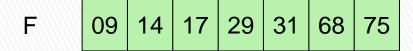
Mezcla Natural



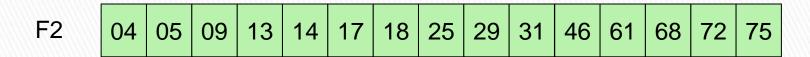
Primera fusión-partición



Segunda fusión-partición



Tercera fusión-partición



Archivo vacío

F3

Algoritmo 8.20 Mezcla_equilibrada

Mezcla_equilibrada (F, F1, F2, F3)

{El algoritmo ordena los elementos del archivo F por el método de mezcla equilibrada. Utiliza tres archivos auxiliares F1, F2 y F3} {BAND es una variable de tipo booleano}

- Llamar al algoritmo Partición_inicial con F, F2 y F3
- Llamar al algoritmo Partición_fusión con F2, F3, F y F1
- 3. Hacer BAND ← FALSO
- Mientras ((F1 ≠ VACÍO) o (F3 ≠ VACÍO)) Repetir
 - 4.1 Si (BAND = VERDADERO)

entonces

Llamar al algoritmo Partición_fusión con F2, F3, F y F1 Hacer BAND \leftarrow FALSO

si no

Llamar al algoritmo Partición_fusión con F, F1, F2 y F3

Hacer BAND ← VERDADERO

- 4.2 {Fin del condicional del paso 4.1}
- § {Fin del ciclo del paso 4}

Algoritmo 8.21 Partición_inicial

Partición_inicial (F, F2, F3)

{El algoritmo produce la partición inicial del archivo F en dos archivos auxiliares, F2 y F3} {AUX y R son variables de tipo entero. BAND es una variable de tipo booleano}

- 👢 Abrir el archivo F para lectura.
- Abrir los archivos F2 y F3 para escritura.
- 3. Leer R de F.
- Escribir R en F2.
- \leq Hacer BAND ← VERDADERO y AUX ← R
- 6. Mientras (no sea el fin de archivo de F) Repetir

Leer R de F

6.1 Si (R ≥ AUX)

entonces

Hacer AUX $\leftarrow R$

61.1 Si (BAND = VERDADERO)

entonces

Escribir R en F2

si no

Capítulo 8



MÉTODOS DE ORDENACIÓN

```
Escribir R en F3
      6.1.2 {Fin del condicional del paso 6.1.1}
         si no
             Hacer AUX \leftarrow R
       6.1.3 Si (BAND = VERDADERO)
                 entonces
                    Escribir R en F3
                    Hacer BAND ← FALSO
                 si no
                    Escribir R en F2
                    Hacer BAND ← VERDADERO
       6.1.4 {Fin del condicional del paso 6.1.3}
   6.2 {Fin del condicional del paso 6.1}
7. {Fin del ciclo del paso 6}
{Cerrar los archivos F, F2 y F3}
```

Algoritmo 8.22 Partición_fusión

Partición_fusión (FA, FB, FC, FD)

{El algoritmo produce la partición y la fusión de los archivos FA y FB, en los archivos FC y FD}

{R1, R2 y AUX son variables de tipo entero. BAN1, BAN2 y BAN3 son variables de tipo booleano}

- Abrir los archivos FA y FB para lectura.
- Abrir los archivos FC y FD para escritura.
- Hacer BAN1 ← VERDADERO, BAN2 ← VERDADERO, BAN3 ← VERDADERO y
 AUX ← -32 768 {AUX se inicializa con un valor negativo alto}
- Mientras ((no sea el fin de archivo de FA) o (BAN1 = FALSO)) y ((no sea el fin de archivo de FB) o ((BAN2 = FALSO)) Repetir

```
4.1 Si (BAN1 = VERDADERO) entonces
          Leer R1 de FA
          Hacer BAN1 ← FALSO
Fin del condicional del paso 4.1 
Si (BAN2 = VERDADERO) entonces
          Leer R2 de FB
          Hacer BAN2 ← FALSO
[44] {Fin del condicional del paso 4.3}
4.5 Si (R1 < R2)
       entonces
   4.5.1 Si (R1 \geq AUX)
            entonces
       4.5.1.1 Si (BAN3 = VERDADERO)
                entonces
                   Escribir R1 en FC
                si no
```

```
Escribir R1 en FD
       4.5.1.2 (Fin del condicional del paso 4.5.1.1)
              Hacer BAN1 \leftarrow VERDADERO y AUX \leftarrow R1
            si no
       4.5.1.3 Si (BAN3 = VERDADERO)
                entonces
                   Escribir R2 en FC
                  Hacer BAN3 ← FALSO
                si no
                   Escribir R2 en FD
Hacer BAN3 ← VERDADERO
       4.5.1.4 {Fin del condicional del paso 4.5.1.3}
   Hacer BAN2 ← VERDADERO y AUX ← -32 768
    4.5.2 (Fin del condicional del paso 4.5.1)
       si no
    4.5.3 Si (R2 \ge AUX)
             entonces
       4.5.3.1 Si (BAN3 = VERDADERO)
                entonces
                   Escribir R2 en FC
                     si no
                   Escribir R2 en FD
       4.5.3.2 {Fin del condicional del paso 4.5.3.1}
```

```
Hacer BAN2 \leftarrow VERDADERO y AUX \leftarrow R2
              si no
           4.5.3.3 Si (BAN3 = VERDADERO)
                    entonces
                       Escribir R1 en FC
                       Hacer BAN3 ← FALSO
                    si no
                       Escribir R1 en FD
                       Hacer BAN3 ← VERDADERO
           4.5.3.4 (Fin del condicional del paso 4.5.3.3)
                  Hacer BAN1 ← VERDADERO y AUX ← -32 768
       4.5.4 {Fin del condicional del paso 4.5.3}
   4.6 {Fin del condicional del paso 4.5}
5. {Fin del ciclo del paso 4}
6. Si (BAN1 = FALSO) entonces
   6.1 Si (BAN3 = VERDADERO)
          entonces
             Escribir R1 en FC
       6.1.1 Mientras (no sea el fin de archivo de FA) Repetir
                  Leer R1 de FA
                  Escribir R1 en FC
       6.1.2 {Fin del ciclo del paso 6.1.1}
         si no
             Escribir R1 en FD
       6.1.3 Mientras (no sea el fin de archivo de FA) Repetir
                  Leer R1 de FA
                  Escribir R1 en FD
```

```
6.1.4 (Fin del ciclo del paso 6.1.3)
   6.2 (Fin del condicional del paso 6.1)

    {Fin del condicional del paso 6}

 Si (BAN2 = FALSO) entonces
   8.1 Si (BAN3 = VERDADERO)
          entonces
             Escribir R2 en FC
       8.1.1 Mientras (no sea el fin de archivo de FB) Repetir
                   Leer R2 de FB
                   Escribir R2 en FC
       8.1.2 (Fin del ciclo del paso 8.1.1)
         si no
             Escribir R2 en FD
       8.1.3 Mientras (no sea el fin de archivo de FB) Repetir
                   Leer R2 de FB
                   Escribir R2 en FD
       8.1.4 (Fin del ciclo del paso 8.1.3)
   8.2 {Fin del condicional del paso 8.1}
 9. (Fin del condicional del paso 8)
10. {Cerrar los archivos FA, FB, FC y FD}
```

Complejidad del método mezcla Natural

Un tiempo de ejecución de $O(n\log_2 n)$ para el caso promedio.

Ejercicios

Fuentes de información

Estructura de datos, Cairo- Guardati, Ed. Mc Graw Hill, Cap. 8