


Ordenamiento Externo

Capítulo 13

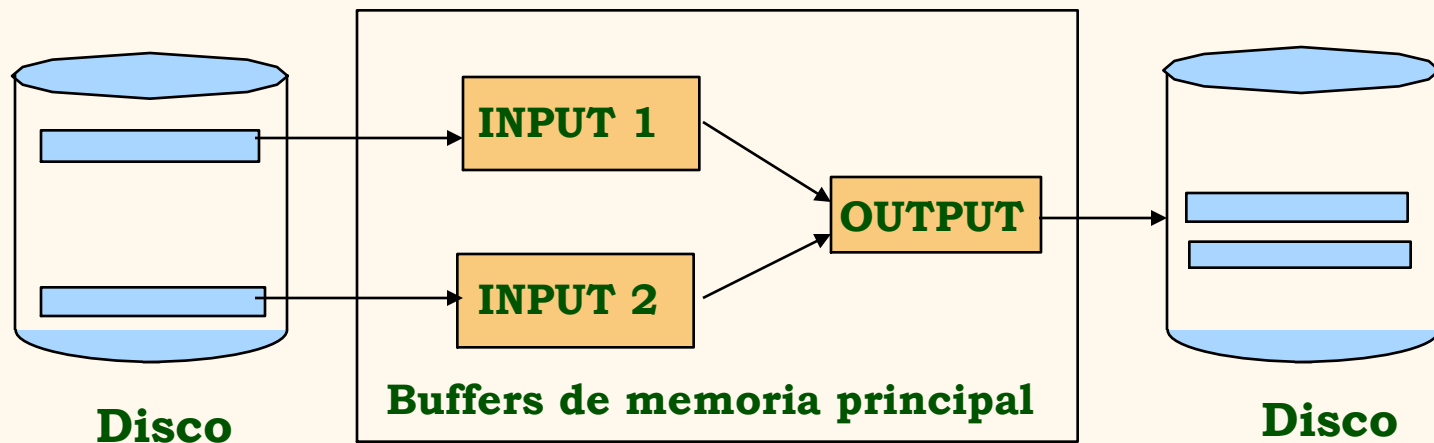


¿Porqué ordenar?

- ❖ Es un problema clásico en ciencias de computación!
- ❖ Los datos son solicitados en algún orden
 - Ej, buscar los estudiantes ordenados ascendentemente por *gpa*
- ❖ Ordenar es el primer paso en una *bulk loading* (carga a granel) en un árbol B+.
- ❖ Ordenar es útil para eliminar *copias duplicadas* en una colección de registros
- ❖ Pero no todo se puede ordenar en la memoria.
- ❖ El algoritmo de ordenamiento une las operaciones *Ordenar-Unir*. (Llamado Merge Sort)

Merge Sort de 2-Vías: Requiere 3 buffers de memoria

- ❖ Paso 1: Leer una página, ordenarla, escribirla.
 - Solo se usa una página del buffer.
- ❖ Pasos 2, 3, ..., etc.:
 - Se usan tres páginas del buffer.



Merge-Sort Externo en Dos-Vias

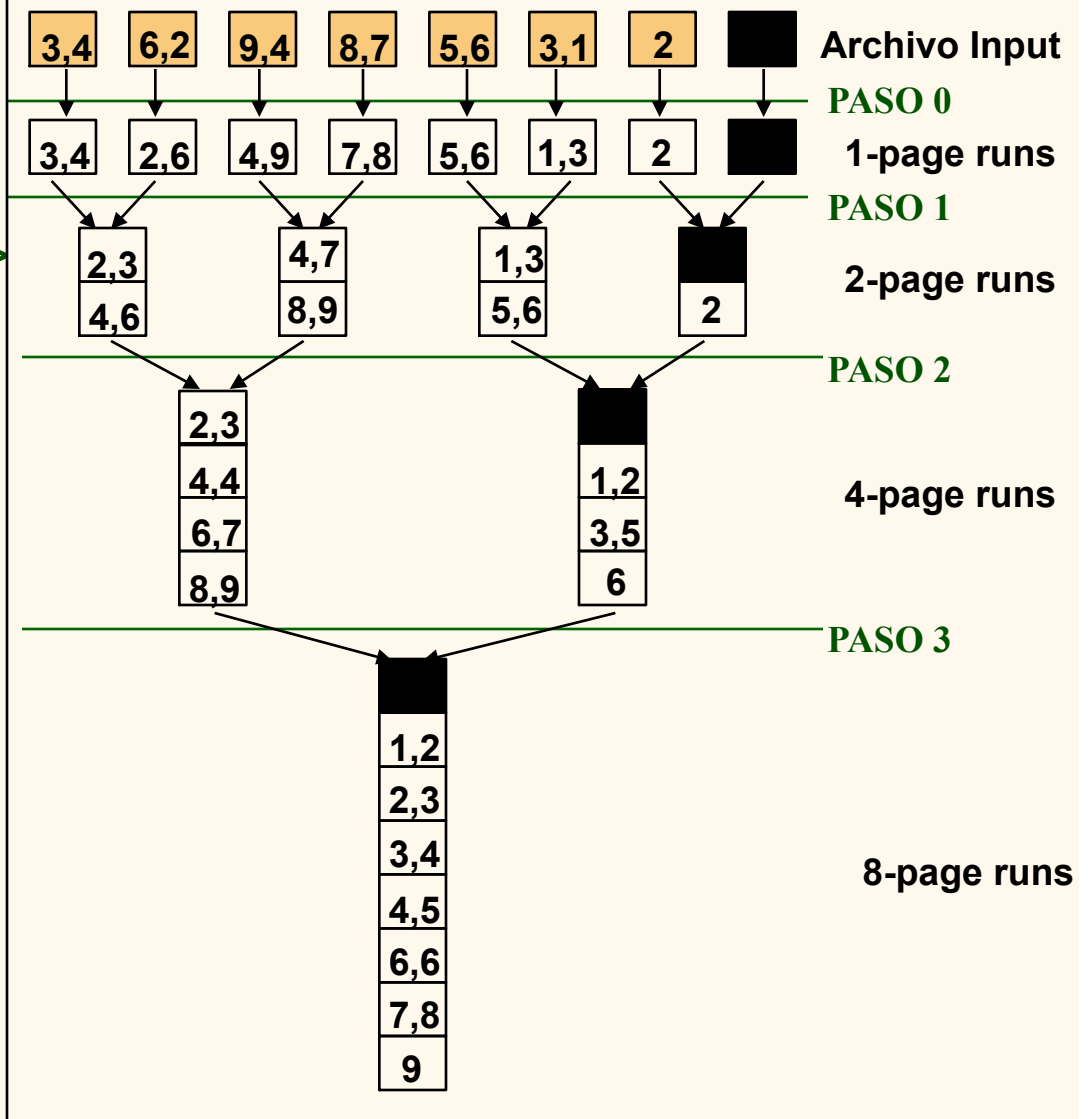
❖ En cada paso leer+escribir cada página en el archivo.

❖ N páginas en el archivo \Rightarrow el número de pasos

$$= \lceil \log_2 N \rceil + 1$$

❖ Entonces el costo total es:

$$2N(\lceil \log_2 N \rceil + 1)$$

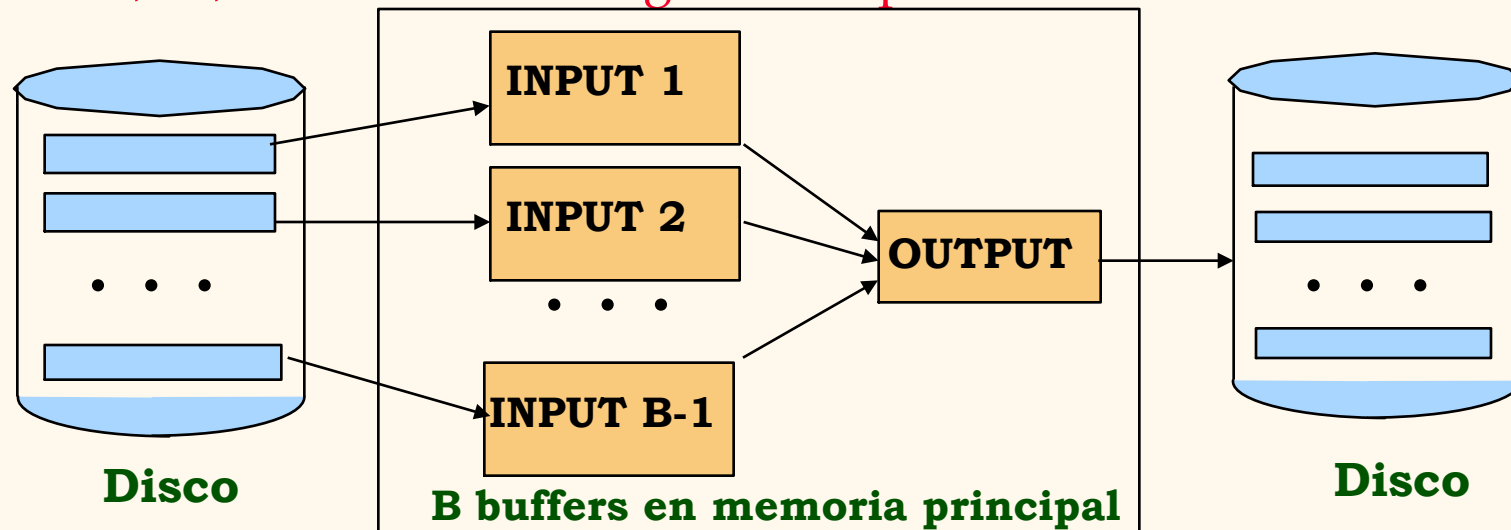


Merge Sort Externo General

➡ *Mas de 3 páginas de buffer. ¿Cómo podemos utilizarlas?*


❖ Para ordenar un archivo con N páginas usando B páginas de buffer:

- Paso 0: usar B páginas de buffer. Produce $\lceil N / B \rceil$ imágenes de ordenamiento de B páginas cada una.
- Paso 2, ..., etc.: une $B-1$ imágenes del paso anterior.




Costo del ordenamiento externo Merge Sort

- ❖ Número de pasos: $1 + \lceil \log_{B-1} \lceil N / B \rceil \rceil$
- ❖ Costo = $2N * (\# \text{ de pasos})$
- ❖ Ej, con 5 páginas de buffer, para ordenar un archivo de 108 páginas:
 - Paso 0: $\lceil 108/5 \rceil = 22$ imágenes de ordenamiento de 5 páginas cada una (la última corrida de 3 páginas solamente)
 - Paso 1: $\lceil 22/4 \rceil = 6$ imágenes de ordenamiento de 20 páginas cada una (la última corrida de 8 páginas solamente)
 - Paso 2: 2 imágenes de ordenamiento, 80 páginas y 28 páginas
 - Paso 3: Archivo ordenado de 108 páginas



Número de pasos del ordenamiento externo

N	B=3	B=5	B=9	B=17	B=129	B=257
100	7	4	3	2	1	1
1,000	10	5	4	3	2	2
10,000	13	7	5	4	2	2
100,000	17	9	6	5	3	3
1,000,000	20	10	7	5	3	3
10,000,000	23	12	8	6	4	3
100,000,000	26	14	9	7	4	4
1,000,000,000	30	15	10	8	5	4



Minimizando el número de imágenes

- ❖ El ordenamiento por reemplazo, puede escribir imágenes de aproximadamente $2*B$ páginas ordenadas.
- ❖ Se leen páginas de el archivo con tuplas a ser ordenadas, R , hasta que el buffer este lleno, y se reserva una página para usarla como un buffer de entrada y una página para usarlo como buffer de salida.
- ❖ Nos referimos al las $B-2$ páginas de R tuplas que no están en los buffer de entrada o salida como el *conjunto actual*. Si el archivo se va a ordenar en orden ascendente sobre alguna llave de búsqueda k , las tuplas se agregan al buffer de salida en orden ascendente por el valor de k .

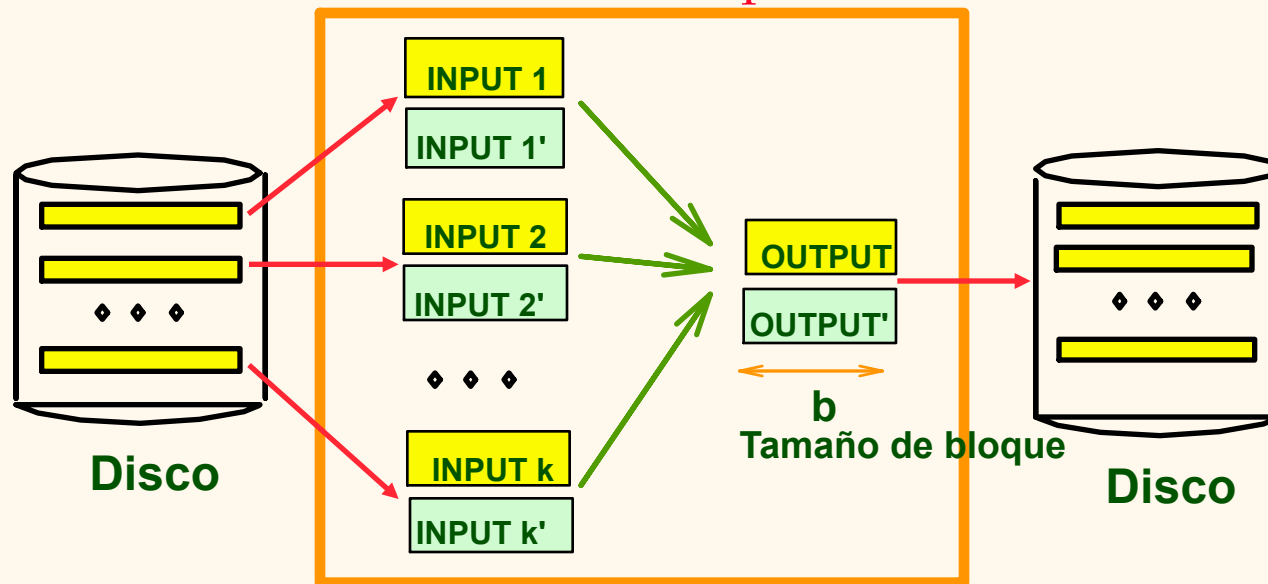


Minimizando el número de imágenes

- ❖ IDEA: escoger varias veces la tupla en el conjunto actual con el menor valor k que sigue siendo mayor que el mas grande en el buffer de salida y se agrega al buffer de salida.
- ❖ Cuando todas las tuplas en el buffer de entrada han sido procesadas de esta manera, se lee la siguiente página del archivo. Se escribe el buffer de salida cuando esta lleno, de modo que se amplia la imagen actual.
- ❖ Cuando cada tupla en el conjunto actual es menor que la mayor tupla en el archivo de salida, el archivo de salida se escribe y se convierte en la última pagina de la imagen actual.

Buffer Doble

- ❖ Para reducir el tiempo de espera de una petición de Lectura o Escritura, es posible realizar *pre-obtención* dentro de un 'bloque sombra'.
 - Se tendrán mas pasos; en la práctica la mayoría de los archivos son ordenados en *2-3 pasos*.



B buffers de memoria principal, merge sort de k-vías

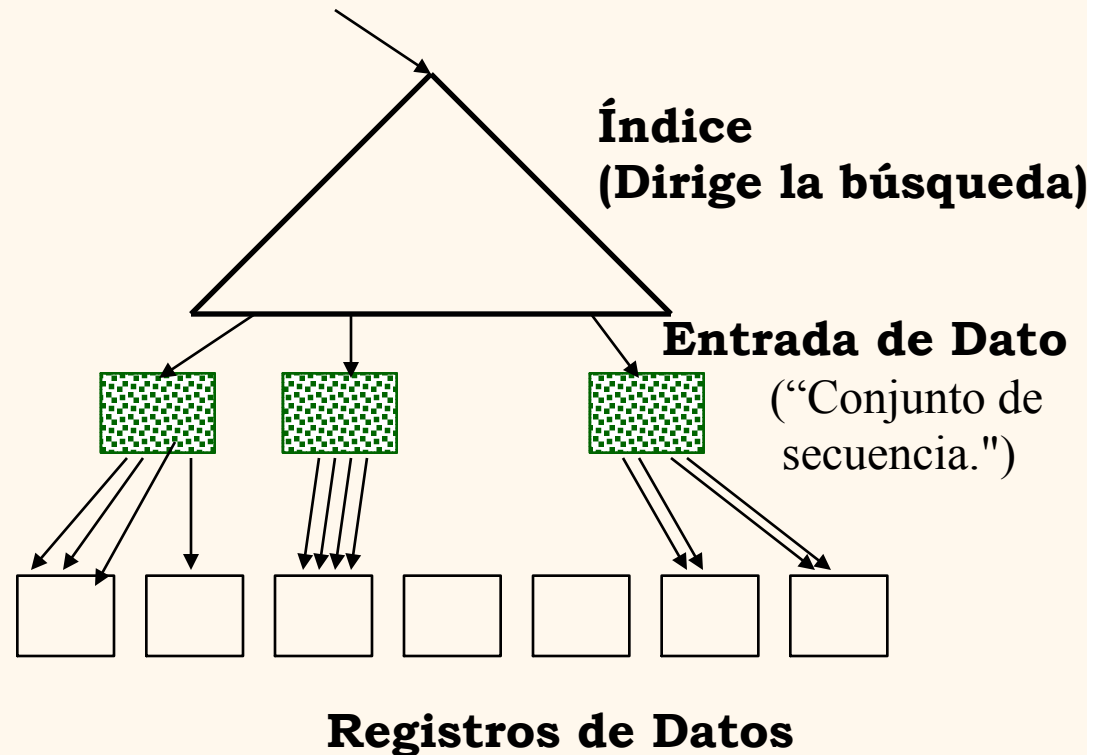


Usando Árboles B+ para ordenar

- ❖ Escenario: La tabla que se va a ordenar tiene un índice tipo árbol B+ sobre columna(s) de ordenamiento.
- ❖ **Idea:** Se obtienen los registros de las páginas hoja.
- ❖ ***Es buena idea?***
- ❖ Casos:
 - Árbol B+ sea un índice **clustered** *Buena idea!*
 - Árbol B+ sea un índice **no clustered** *Podría ser una muy mala idea!*

Uso de Árbol B+ Cluster para ordenamiento

- ❖ Costo: de la raíz a la hoja de mas a la izquierda, luego recuperar todas las páginas hoja (Alternativa 1)
- ❖ ¿y si se usa la Alternativa 2? El costo adicional de recuperar los registros de datos: Cada página se obtiene una sola vez.



➡ *Siempre es mejor que el ordenamiento externo!*

Uso de Árbol B+ No Cluster para ordenamiento

- ❖ Alternativa (2) para entrada de datos; cada entrada de datos contiene el *rid* de un registro de dato. En general, **un operación de Entrada/Salida por registro de datos.**

