

Sobre la complejidad de MemoriaSecundariaMergesort

Nos interesa determinar el número de datos leídos y escritos en memoria secundaria.

Sobre la complejidad de MemoriaSecundariaMergesort

Nos interesa determinar el número de datos leídos y escritos en memoria secundaria.

El número inicial de runs es $\left\lceil \frac{N}{\lfloor \frac{M}{B} \rfloor \cdot B} \right\rceil$

Sobre la complejidad de MemoriaSecundariaMergesort

Nos interesa determinar el número de datos leídos y escritos en memoria secundaria.

El número inicial de runs es $\left\lceil \frac{N}{\lfloor \frac{M}{B} \rfloor \cdot B} \right\rceil$

- ▶ Lo cual es aproximadamente $\frac{N}{M}$

Sobre la complejidad de MemoriaSecundariaMergesort

Nos interesa determinar el número de datos leídos y escritos en memoria secundaria.

El número inicial de runs es $\left\lceil \frac{N}{\lfloor \frac{M}{B} \rfloor \cdot B} \right\rceil$

- ▶ Lo cual es aproximadamente $\frac{N}{M}$

Por lo tanto existe una constante $c \in \mathbb{R}^+$ tal que el número de llamadas a **MemoriaSecundariaMergesort** está acotado por $c \cdot \log_2\left(\frac{N}{M}\right)$

Sobre la complejidad de MemoriaSecundariaMergesort

Nos interesa determinar el número de datos leídos y escritos en memoria secundaria.

El número inicial de runs es $\left\lceil \frac{N}{\lfloor \frac{M}{B} \rfloor \cdot B} \right\rceil$

- ▶ Lo cual es aproximadamente $\frac{N}{M}$

Por lo tanto existe una constante $c \in \mathbb{R}^+$ tal que el número de llamadas a **MemoriaSecundariaMergesort** está acotado por $c \cdot \log_2\left(\frac{N}{M}\right)$

- ▶ Concluimos entonces que el número de datos leídos y escritos en memoria secundaria está acotado por $2 \cdot N \cdot c \cdot \log_2\left(\frac{N}{M}\right)$. ¿Por qué?

Sobre la complejidad de MemoriaSecundariaMergesort

¿Es posible reducir el número de datos leídos y escritos en memoria secundaria por el algoritmo?

Sobre la complejidad de MemoriaSecundariaMergesort

¿Es posible reducir el número de datos leídos y escritos en memoria secundaria por el algoritmo?

El procedimiento **Merge** mezcla 2 runs.

- ▶ Para hacer esto utiliza 3 listas, 2 de las cuales están destinadas para los runs y la otra para el resultado de la mezcla

Sobre la complejidad de MemoriaSecundariaMergesort

¿Es posible reducir el número de datos leídos y escritos en memoria secundaria por el algoritmo?

El procedimiento **Merge** mezcla 2 runs.

- ▶ Para hacer esto utiliza 3 listas, 2 de las cuales están destinadas para los runs y la otra para el resultado de la mezcla

El procedimiento **Merge** podría mezclar 4 runs si se tiene suficiente memoria secundaria.

- ▶ Tendría que usar entonces 5 listas, 4 de las cuales estarían destinadas para los runs

Sobre la complejidad de MemoriaSecundariaMergesort

¿Cuál es el número de llamadas a MemoriaSecundariaMergesort con esta nueva versión de Merge?

Sobre la complejidad de MemoriaSecundariaMergesort

¿Cuál es el número de llamadas a MemoriaSecundariaMergesort con esta nueva versión de Merge?

- ▶ Existe una constante $d \in \mathbb{R}^+$ tal que el número de llamadas a MemoriaSecundariaMergesort está acotado por $d \cdot \log_4\left(\frac{N}{M}\right)$

Sobre la complejidad de MemoriaSecundariaMergesort

¿Cuál es el número de llamadas a **MemoriaSecundariaMergesort** con esta nueva versión de **Merge**?

- ▶ Existe una constante $d \in \mathbb{R}^+$ tal que el número de llamadas a **MemoriaSecundariaMergesort** está acotado por $d \cdot \log_4\left(\frac{N}{M}\right)$

Dado que $\log_4\left(\frac{N}{M}\right) = \frac{1}{2} \cdot \log_2\left(\frac{N}{M}\right)$, concluimos que la nueva versión del algoritmo realiza la mitad de lecturas y escrituras de datos en memoria secundaria, comparado con la primera versión del algoritmo.

Sobre la complejidad de MemoriaSecundariaMergesort

¿Cuál es el número de llamadas a **MemoriaSecundariaMergesort** con esta nueva versión de **Merge**?

- ▶ Existe una constante $d \in \mathbb{R}^+$ tal que el número de llamadas a **MemoriaSecundariaMergesort** está acotado por $d \cdot \log_4\left(\frac{N}{M}\right)$

Dado que $\log_4\left(\frac{N}{M}\right) = \frac{1}{2} \cdot \log_2\left(\frac{N}{M}\right)$, concluimos que la nueva versión del algoritmo realiza la mitad de lecturas y escrituras de datos en memoria secundaria, comparado con la primera versión del algoritmo.

- ▶ Nótese que ésta es una aproximación que funciona bien dado que las constantes c y d son similares

Sobre la complejidad de MemoriaSecundariaMergesort

Si el procedimiento **Merge** puede mezclar un mayor número de runs, entonces se puede seguir reduciendo el número de datos escritos y leídos en memoria secundaria.

- ▶ Hay que considerar que mezclar un mayor número de runs aumenta la complejidad de **Merge**, la cual debe ser comparada con el costo de leer y escribir en memoria secundaria

Sobre la complejidad de MemoriaSecundariaMergesort

Si el procedimiento **Merge** puede mezclar un mayor número de runs, entonces se puede seguir reduciendo el número de datos escritos y leídos en memoria secundaria.

- ▶ Hay que considerar que mezclar un mayor número de runs aumenta la complejidad de **Merge**, la cual debe ser comparada con el costo de leer y escribir en memoria secundaria

El número máximo de runs que puede mezclar es $\lfloor \frac{M}{B} \rfloor - 1$

Sobre la complejidad de MemoriaSecundariaMergesort

Si el procedimiento **Merge** puede mezclar un mayor número de runs, entonces se puede seguir reduciendo el número de datos escritos y leídos en memoria secundaria.

- ▶ Hay que considerar que mezclar un mayor número de runs aumenta la complejidad de **Merge**, la cual debe ser comparada con el costo de leer y escribir en memoria secundaria

El número máximo de runs que puede mezclar es $\lfloor \frac{M}{B} \rfloor - 1$

- ▶ Este número es fijo y no depende de N , por lo que el número de datos leídos y escritos en memoria secundaria siempre va a ser de la forma $e \cdot N \cdot \log_2\left(\frac{N}{M}\right)$, donde e es una constante