Note Title 9/7/2010

(0) Nuevas Notaciones

B = Tamaño página (e.g. en bytes)

$n = \frac{N}{B}$ = # páginas a ordenar

N = # cantidad de elementos

$m = \frac{M}{B}$ > # páginas que quedan en memoria principal

M = # cantidad de elementos que quedan en memoria principal

hint

$n << N$

$m << M$

M, N en elementos

n, m en páginas

Ordenamiento en Memoria Secundaria

Note Title 9/7/2010

① Como Ordenar?

(Quicksort)

Heapsort → Binary Heap

→ Heap en Memoria Secundaria

$O(N \log_B N)$

Merge Sort

Insertion Sort → Diccionario

→ B-arbol, ver Ende Boas

$O(N \log_B N)$

② Eso es optimal?

(a) en RAM

① suponer input es permutacion

② Cuanto Log? ($n!$)

③ cuanto información me da una comparación?

Arbol de decision Binario

$\Rightarrow$ dividiendo ~~el menos~~ el maximo por dos la cantidad de permutaciones posibles en el peor caso.

④ Cuanto comparaciones hasta que puedo identificar la permutacion

$\log_2(n!) = \lg(n!) \simeq n \lg n$

(b) En Memoria Secundaria

① Supone que cada pagina ya es ordenada.

$\Rightarrow \frac{N!}{(B!)^n}$ permutaciones al inicio

② Cada acceso a secundaria reduce de $1/\binom{M}{B}$ la cantidad de permutaciones

③ dos pues de t accesos independientes

Quedan $\frac{N!}{(B!)^n} \frac{1}{\left(\binom{M}{B}\right)^t}$

④ $\frac{N!}{(B!)^n} \frac{1}{\left(\binom{M}{B}\right)^t} \leq 1$) α

$\Rightarrow N! \leq (B!)^n \left(\binom{M}{B}\right)^t$) lg + Stirling

$\Rightarrow N \lg N \leq \underline{n} B \lg B + t B \lg \frac{M}{B}$
N

$$\Rightarrow t \geq \frac{N \lg N - N \lg B}{B \lg \frac{n}{B}}$$

$$= \left(\frac{N}{B}\right)\left(\frac{\lg \frac{N}{B}}{\lg \frac{n}{B}}\right)$$

($\frac{N}{B}$: n; $\frac{n}{B}$: m)

$$t \geq n \frac{\log_2 n}{\log_2 m}$$

$$\boxed{t \geq n \log_m n}$$

③ Cota Superior: MergeSort

- Similar a mergeSort binario,

pero con grado $(m-1)$, y $n = \frac{N}{B}$ hojas (ordenados)

$\Rightarrow \log_{m-1} n$ niveles, cada nivel cuesta n accessos

$$\Rightarrow O(n \log_m n + n)$$

Conclusion(es)

- MergeSort es optimal (en el peor caso por N, B, M fijos) por la cantidad de accesos a memoria externa
- HeapSort y InsertionSort (elementos por elemento) no son optimales (tendrían que manejar una página en cada operación ⇝ se acerca a mergesort)