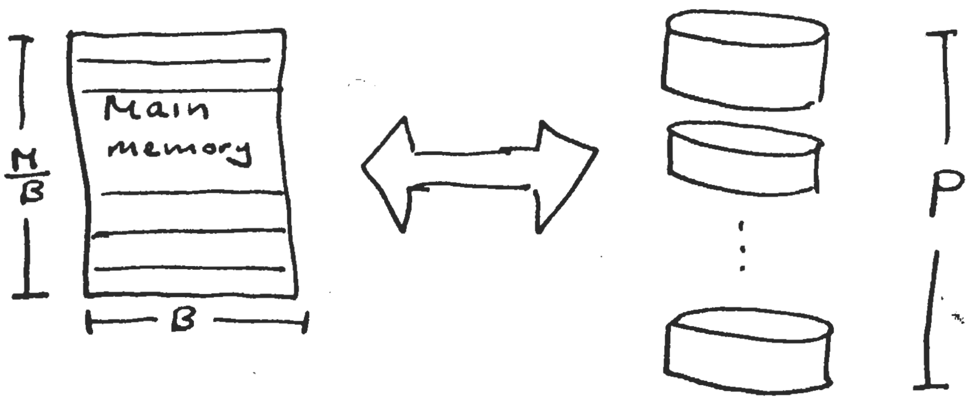
Sorting and Permuting on Sequential and Parallel Disks.

Notation: N = # records to sort
M = # records that fit in internal memory
B = # records in a transfer block
P = # blocks transferred concurrently.



DAM Model (Aggarwal, Vitter 88)

Cost of block transfer = 1
P blocks transfered concurrently, explicity manage memory.
goal: minimize # memory transfers

Compare With Cache-Oblivious Model:

$P = 1$, other parameters unknown
System manages memory

shoulders burden of memory management

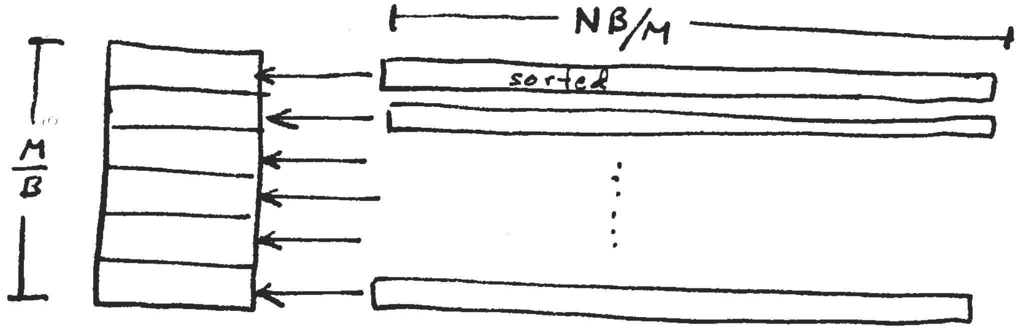
Results

Theorem: The average-case and worst-case cost to sort N records is $\Theta\left(\frac{N}{PB} \frac{\log(1+N/B)}{\log(1+M/B)}\right)$.

Theorem: The average-case and worst-case cost to permute N records is

$$\Theta\left(\min\left\{\frac{N}{P}, \frac{N}{PB} \frac{\log(1+N/B)}{\log(1+M/B)}\right\}\right).$$

Parallel Mergesort for $P = 1$.

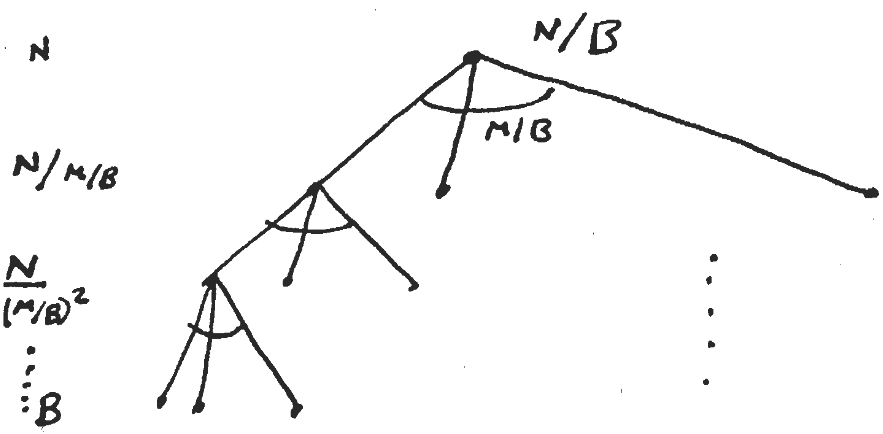


Total # Mem transfers:

$$T(N) = \frac{N}{B} + \frac{M}{B}\, T\left(\frac{N}{M/B}\right)$$

$$T(B) = 1$$

Solution :



$$\frac{\left(\frac{N}{M/B}\right)}{B} \cdot \frac{M}{B} = \frac{N}{B}$$

#levels
height $= \log_{M/B} N - \log_{M/B} B = \log_{M/B} N/B.$

cost per level $= N/B$

$$T(N) = \frac{N}{B} \log_{M/B} \frac{N}{B}.$$

Note: For simplicity I'm removing "1+".

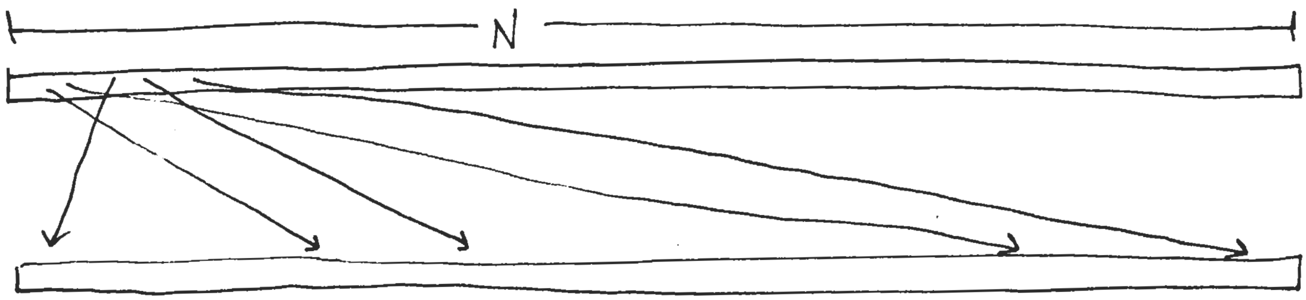
Note: The parallel mergesort doesn't immediately work for a nonconstant P, but can be made to work...

Permuting for $P=1$:

2 choices:

1) sort $\Rightarrow$ same bounds as before

2) put each element directly in its destination
$\Rightarrow$ 1 memory transfer per element



Reminder: Sorting LB.

$n!$ permutations ~~[illegible]~~ consistent with info.

each comparison rules out at most half.

need $\log(n!) = \Omega(n \log n)$ compares.

Lower Bound on Sorting for $P=1$

<u>Thm:</u> External sorting requires $\Omega(N/B \log_{M/B} N/B)$ I/O's in comparison-I/O model (comparison is only allowed op in internal memory)

Proof: Information-theoretic argument.

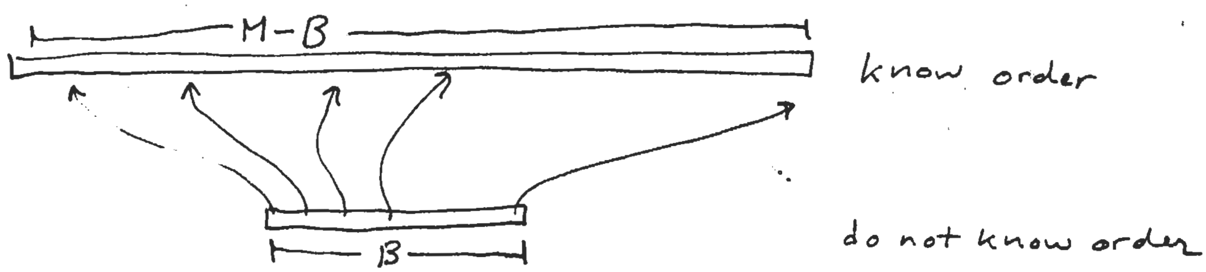
At beginning of computation, $N!$ possibilities available for correct ordering based on available information (none).

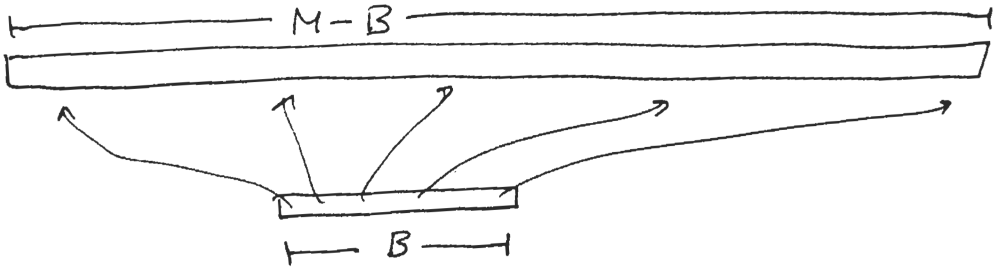
After each input we learn through comparisons, narrowing down possible number of orders.

Show that need $t = \Omega(N/B \log_{M/B} N/B)$ inputs to learn enough (narrow down possibilities) that only one consistent order left.

Two cases:

Case 1: We know order of elements in internal memory but not order of block B being input.





\# possible orderings in memory

$$\leq (B!)\left(\frac{M-B+B}{(M-B)!\;B!}\right)$$

order in block — # interleavings (∗s and ▭s)

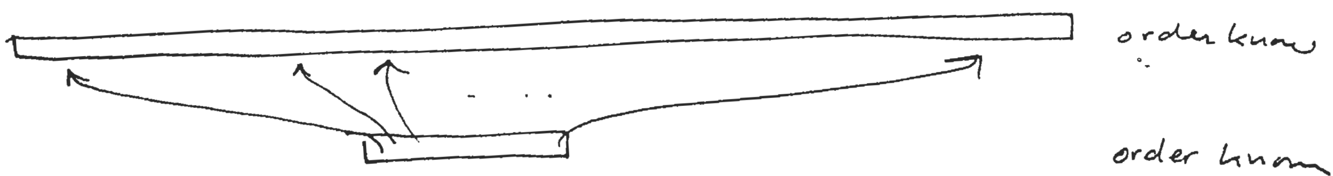
$$\leq (B!)\binom{M}{B}.$$

If S denotes # possible orderings of N elmts before input, $\exists$ one of $(B!)\binom{M}{B}$ orderings within memory, such that # remaining orderings still consistent is

$$\geq \frac{S}{(B!)\binom{M}{B}}.$$

After t inputs of case 1: # remaining orderings $\geq \dfrac{S}{\left(B!\binom{M}{B}\right)^t}$.

CASE 2: Order of records in both main memory and input block already known (e.g. input block was output previously).



\# possible orderings in memory

$$\leq \binom{M}{B}.$$

Claim: # times we can read a block of B records that have not been together in memory: N/B.

Lemma: After t input operations, at least

$$\frac{N!}{\binom{M}{B}^t (B!)^{N/B}}$$

orderings are consistent with available information.

Goal: Narrow down possible orderings to 1:

Thm: # I/O's, t, must satisfy

$$\frac{N!}{\binom{M}{B}^t (B!)^{N/B}} \leq 1.$$

Useful formulae:

- $\log(x!) = \Theta(x \log x)$ (stirling) $n! \approx \sqrt{2n\pi}\left(\frac{n}{e}\right)^n$
- $\log\binom{M}{B} = \Theta\left(B \log \frac{M}{B}\right)$

  Pf. $\left(\frac{M}{B}\right)^B \leq \binom{M}{B} \leq \left(\frac{eM}{B}\right)^B$.

Solve for t:

$$\frac{N!}{\left(\frac{M}{B}\right)^t (B!)^{N/B}} \leq 1$$

$$\left(\frac{M}{B}\right)^t (B!)^{N/B} \geq N!$$

$$t \log\left(\frac{M}{B}\right) + \frac{N}{B} \log(B!) \geq \log(N!)$$

$$t B \log\left(\frac{M}{B}\right) + \frac{N}{B} B \log B \geq \Omega(N \log N)$$

$$t B \log\left(\frac{M}{B}\right) \geq \Omega\left(N \log \frac{N}{B}\right)$$

$$t \geq \Omega\left(\frac{N}{B} \cdot \frac{\log(N/B)}{\log(M/B)}\right)$$

$$t = \Omega\left(\frac{N}{B} \log_{M/B}(N/B)\right).$$

Notation from I/O efficient algs:

$m = M/B$

$n = N/B.$

$\Rightarrow \quad \Omega(n \log_m n).$

Lower Bound on External Permuting

<u>Thm</u>: Rearranging N elements according to a given permutation requires $\Omega(\min(N, \frac{N}{B} \log_{M/B} N/B))$ I/O operations.

Pf:

Model Assumptions:

1) External memory comprised of N <u>blocks</u> of size B (size NB). An I/O moves a single block.

2) I/O's are simple. Transfer of elements only allowed operation — no new elements or duplicates.

(3) Main memory + Disk viewed as big extended array.

Def. Permutation = order of elements in extended array
(ignore spaces)

Claim: Assumptions $\Rightarrow$ exactly one permutation at all ~~ti~~

Idea: Bound # permutations for t I/Os.

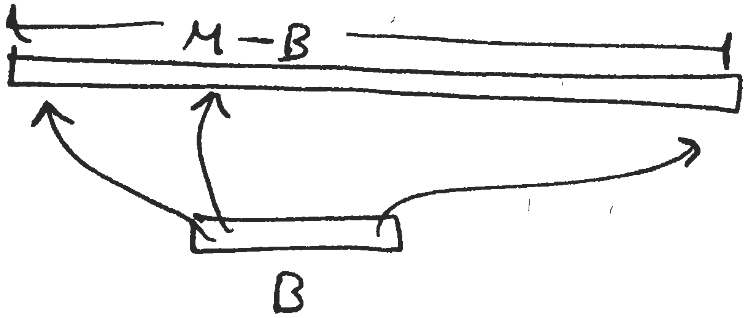
Initially: 1 permutation

Require: N! permutations.

Input:

- choice of N blocks to input
- after loading one block, can put $\le B$ elmts between $\le M-B$ locations in memory.

$\Rightarrow$



2 cases:

1) Virgin block, new $\quad N(B!)\binom{M}{B}\times$ (# permutations already)

2) already-read block: $N\binom{M}{B}\times$ (# perms already)

Claim: case (1) can happen $\le N/B$ times.

<u>output</u>:

N target blocks to output. B elements to pick.

<u>Claim</u>: After t I/Os $\le$

$$(B!)^{N/B}\left(N\binom{M}{B}\right)^t$$

perms attainable.

$$(B!)^{N/B}\left[\left(\frac{M}{B}\right)N\right]^t \geq N!$$

$$\frac{N}{B}\log(B!) + t\left[\log\left(\frac{M}{B}\right) + \log N\right] \geq \log(N!)$$

$$N\log B + t\left[B\log(M/B) + \log N\right] \geq \Omega(N\log N)$$

$$t\left[B\log\left(\frac{M}{B}\right) + \log N\right] \geq \Omega\left(N\log(N/B)\right)$$

$$t \geq \Omega\left[\frac{N\log N/B}{B\log\left(\frac{M}{B}\right)+\log N}\right]$$

2 cases:

Case 1: $\log N \leq B \log M/B$

$$\Rightarrow \quad t \geq \Omega\left(\frac{N}{B} \log_{M/B} N/B\right)$$

Case 2: $\log N > B \log(M/B) \Rightarrow \boxed{B \ll \sqrt{N}}$

$$\frac{N \log N/B}{2 \log N} = \frac{N \log N - N \log B}{2 \log N}$$

$$= \frac{1}{2}\left[N - N \frac{\log(B)}{\log(N)}\right].$$

$$= \frac{1}{2}\left(N - \frac{1}{2} N\right)$$

$$= \Omega(N)$$

Model Justification:

nonsimple → simple: remove all I/O's not present in final perm.

~~Block size~~

size assumption → no reason to have blocks that are empty.

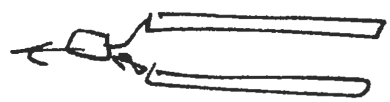
Infomertial: Cache-Oblivious Sorting

Cost of mergesort:

$$T(N) = 2T(N/2) + N/B$$

$$T(B) = 1$$

$$T(N) = \Theta\left(\frac{N}{B}\log_2 N\right).$$



Need multiway merge! But how big??