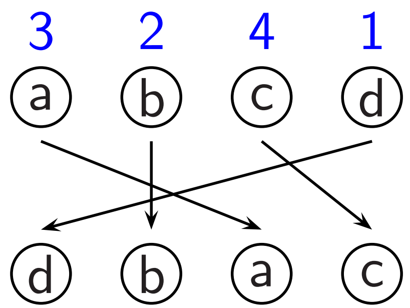
Permuting

Upper and Lower bounds

[Aggarwal, Vitter, 88]

Upper Bound

Assume instance is specified by each element knowing its final position:



Algorithm	Internal Cost	I/O Cost
1) Place each element directly	$\Theta(N)$	$\Theta(N)$
2) Sort on final position	$\Theta(N \log N)$	$\Theta(N/B \log_{M/B}(N/B))$

Upper Bound

Internally, 1) always best.

Externally, 2) best when $1/B \log_{M/B}(N/B) \leq 1$.

Note: This is almost always the case practice. Example:

$$B = 10^3,\ M = 10^6,\ N = 10^{30}$$

$$\Downarrow$$

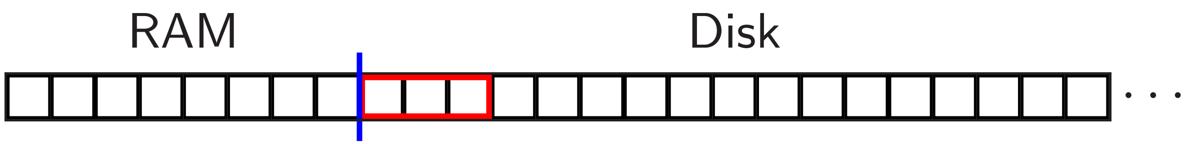
$$1/B \log_{M/B}(N/B) = 9/10^3 << 1$$

External Permuting:

$$\boxed{O(\min\{N/B \log_{M/B}(N/B), N\}) = O(\min\{\text{sort}(N), N\})}$$

Lower Bound Model

Model of memory:



- Elements are indivisible: May be moved, copied, destroyed, but newer broken up in parts.
- Assume $M \geq 2B$.
- May assume I/Os are block-aligned, and that at start [end], input [output] is in lowest contiguous positions on disk.

Lower Bound

We may assume that elements are only **moved**, not copied or destroyed.

Reason: For any sequence of I/Os performing a permutation, exactly one copy of each element exists at end. Change all I/Os performed to only deal with these copies. Result: same number of I/Os, same permutation at end, but now I/Os only move elements.

Consequence:

Memory always contains a permutation of the input

Define:

$$S_t = \text{number of permutations possible to reach with } t \text{ I/Os.}$$

If new X choices to make during I/O: $S_{t+1} \leq X \cdot S_t$.

Bounds on Value of X

Type of I/O	Read untouched block	Read touched block	Write
X	$\frac{N}{B}\binom{M}{B}B!$	$N\binom{M}{B}$	$N\binom{M}{B}$

Note: At each I/O, we may have all three types of I/O available. There is at most N/B reads of untouched blocks in any I/O-sequence.

Hence, from $S_0 = 1$ and $S_{t+1} \leq X \cdot S_t$ we get

$$S_t \leq \left(3\binom{M}{B}N\right)^t (B!)^{N/B}$$

To be able to reach every possible permutation, we need $N! \leq S_t$. Thus,

$$N! \leq \left(3\binom{M}{B}N\right)^t (B!)^{N/B}$$

is necessary for any permutation algorithm with a worst case complexity of t I/Os.

Lower Bound Computation

$$\left(3\binom{M}{B}N\right)^t (B!)^{N/B} \geq N!$$

$$t(\log\binom{M}{B} + \log N + \log 3) + (N/B)\log(B!) \geq \log(N!)$$

$$t(3B\log(M/B) + \log N + \log 3) + N\log B \geq N(\log N - 1/\ln 2)$$

$$t \geq \frac{N(\log N - 1/\ln 2 - \log B)}{3B\log(M/B) + \log N + \log 3}$$

$$t = \Omega(\frac{N\log(N/B)}{B\log(M/B) + \log N})$$

Using Lemma:

a) $\log(x!) \geq x(\log x - 1/\ln 2)$

b) $\log(x!) \leq x\log x$

c) $\log\binom{x}{y} \leq 3y\log(x/y)$ when $x \geq 2y$

Lower Bound

$$\Omega(\frac{N\log(N/B)}{B\log(M/B)+\log N})$$
$$=\Omega(\min\{\frac{N\log(N/B)}{B\log(M/B)},\frac{N\log(N/B)}{\log N}\})$$
$$=\Omega(\min\{Z_1,Z_2\})$$

Note 1: $Z_1 = \mathrm{sort}(N)$

Note 2: $Z_2 < Z_1 \Leftrightarrow B\log(M/B) < \log N \Rightarrow B < \log N \Rightarrow$

$$Z_2 = \frac{N\log(N/B)}{\log N} = \frac{N(\log N - \log B)}{\log N} = \Theta(N)$$

Note 3: $Z_2 \leq N$ always

By Note 2 and 3, it is OK to substitute N for Z_2 inside $\min$.

The I/O Complexity of Permuting

We have proven:

$$\boxed{\Theta(\min\{\mathrm{sort}(N), N\})}$$