Permuting

Upper and Lower bounds

[Aggarwal, Vitter, 88]
Upper Bound

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

\[
\begin{array}{cccc}
3 & 2 & 4 & 1 \\
a & b & c & d \\
d & b & a & c \\
\end{array}
\]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Internal Cost</th>
<th>I/O Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Place each element directly</td>
<td>( \Theta(N) )</td>
<td>( \Theta(N) )</td>
</tr>
<tr>
<td>2) Sort on final position</td>
<td>( \Theta(N \log N) )</td>
<td>( \Theta(N/B \log_{M/B}(N/B)) )</td>
</tr>
</tbody>
</table>
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:

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

<table>
<thead>
<tr>
<th>Type of I/O</th>
<th>Read untouched block</th>
<th>Read touched block</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>( \frac{N}{B} \binom{M}{B} B! )</td>
<td>( N \binom{M}{B} )</td>
<td>( N )</td>
</tr>
</tbody>
</table>

Note: at most \( N/B \) I/0s on untouched blocks.

From \( S_0 = 1 \) and \( S_{t+1} \leq X \cdot S_t \) we get

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

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

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

is necessary for any permutation algorithm with a worst case complexity of \( t \) I/0s.
Lower Bound Computation

\[
\begin{align*}
&\left( \left( \frac{M}{B} \right)^N \right)^t (B!)^{N/B} \geq N! \\
t(\log \left( \frac{M}{B} \right) + \log N) + (N/B) \log(B!) \geq \log(N!) \\
t(3B \log(M/B) + \log N) + N \log B \geq N(\log N - 1/\ln 2) \\
t \geq \frac{N(\log N - 1/\ln 2 - \log B)}{3B \log(M/B) + \log N} \\
t = \Omega\left( \frac{N \log(N/B)}{B \log(M/B) + \log N} \right)
\end{align*}
\]

Using Lemma:

a) \( \log(x!) \geq x(\log x - 1/\ln 2) \)
b) \( \log(x!) \leq x \log x \)
c) \( \log \left( \frac{x}{y} \right) \leq 3y \log(x/y) \) when \( x \geq 2y \)
Lower Bound

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

\[ = \Omega\left( \min\left\{ \frac{N \log(N/B)}{B \log(M/B)}, \frac{N \log(N/B)}{\log N} \right\} \right) \]

\[ = \Omega\left( \min\{Z_1, Z_2\} \right) \]

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

Note 2: \( Z_2 < Z_1 \iff B \log(M/B) < \log N \iff B < \log N \iff \)

\[ Z_2 = \frac{N \log(N/B)}{\log N} = \frac{N(\log N - \log B)}{\log N} = \Theta(N) \]
The I/O Complexity of Permuting

We have proven:

$$\Theta(\min\{\text{sort}(N), N\})$$