Lower Bound on External Permuting

Theorem 1 *Rearranging N elements according to a given permutation requires $\Omega(\min\{N, n\log_m n\})$ I/O-operations.*

Proof: To prove the theorem we first make a few extra model assumptions:

1. External memory is divided into N *tracks* of B elements; An I/O moves the elements in a track to/from main memory.
2. I/Os are *simple*: Transfer of elements is the only allowed operation; No new elements (or copies of existing elements) can be produced.
3. Main memory and disk is viewed as one big *extended storage array.*

The idea in the proof is now to bound the number of permutations we can produce with a given number of I/Os. A permutation is defined as the layout of elements in the extended storage array, ignoring empty entries, and the assumptions means that we can identify one unique permutation of the N elements at any given time of an algorithm. Initially we have one permutation (the original layout of the elements) and we want to be able to produce $N!$ permutations.

Consider an input operation. We can choose between N track to input. After loading one such track we can rearrange the $\leq B$ elements in the track and place these elements between the $\leq M - B$ elements in memory. Thus we produce at most $N \cdot B! \cdot \binom{M}{B}$ new permutations. If the loaded track has been touched previously we do not get the $B!$ term. The number of times we can read such a "fresh" track is at most $\frac{N}{B}$. During an output we can choose between N tracks to output, and in total we can produce at most $(B!)^{\frac{N}{B}} \cdot (\binom{M}{B} N)^t$ permutations after t I/Os.

Using the (rough) assumptions that $\log x! = x \log x$ (Stirlings formula) and $\log \binom{M}{B} = B \log \frac{M}{B}$ we then obtain the following lower bound.

$$\begin{array}{rcc}
(B!)^{\frac{N}{B}} \cdot (\binom{M}{B} N)^t & \geq & N! \\
\frac{N}{B}\log(B!) \cdot t(\log N + \log \binom{M}{B}) & \geq & \log(N!) \\
N \log B + t(\log N + B \log \frac{M}{B}) & \geq & N \log N \\
t & \geq & \frac{N \log \frac{N}{B}}{\log N + B \log \frac{M}{B}}
\end{array}
\quad
\begin{array}{c}
\Downarrow \\ \Downarrow \\ \Downarrow
\end{array}$$

If $\log N \leq B \log \frac{M}{B}$ we have $t \geq \frac{N \log \frac{N}{B}}{2B \log \frac{M}{B}} = \Omega(n \log_m n)$.

If $\log N > B \log \frac{M}{B}$ (which means that $B << \sqrt{N}$) we have $\frac{N \log \frac{N}{B}}{2 \log N} = \frac{N \log N - N \log B}{2 \log N} = \frac{1}{2}(N - N\frac{\log B}{\log N}) \geq \frac{1}{2}(N - \frac{1}{2}N) = \Omega(N)$.

Note that there must be at least one permutation σ that requires $\Omega(\min\{N, n\log_m n\})$ I/Os. We can remove the assumption that I/Os are simple since an algorithm producing σ using non-simple I/Os can be transformed into an algorithm using only simple I/Os by removing all elements that are not present in the final permutation. The track assumption can also be removed since we can simulate a track crossing I/O with $O(1)$ I/Os respecting track boundaries.

$\square$

Lower Bound on External Sorting

Theorem 2 *External sorting requires $\Omega(n \log_m n)$ I/Os in the comparison I/O model (comparisons only allowed operations in internal memory).*

Proof: We have N records to sort, therefore there are $N!$ possibilities for the correct ordering that are consistent with the information we have from the start (which is none). The idea is now to see how much we can narrow down this number using one input operation and whatever number of comparisons we want, under the assumption that an adversary chooses the worst possible outcome of the comparisons we perform. Needless to say, output operations cannot contribute to narrowing down the possibilities because any information we can get after the output we could have obtained before the output.

Consider an input of B records into internal memory. Assuming that we know the order of the records already in internal memory, but not the order of the B newly read records, there are at most $\binom{M}{B}(B!)$ possible orderings of the records in internal memory. If S denotes the number of possible orderings before the input, there exists at least one of the $\binom{M}{B}(B!)$ orderings of records in internal memory, such that the number of total orderings (of the initial N records) consistent with this ordering, is at least $\frac{S}{\binom{M}{B}(B!)}$. The adversary always chooses one such ordering. It follows that after t inputs, the number of possible orderings is at least $\frac{N!}{\left(\binom{M}{B}(B!)\right)^t}$.

The above was under the assumption that we did not know the order of the B records read into internal memory. This is not the case if the B records have been together in internal memory previously, because we always determine the order of the records in internal memory after an input. The number of times we can read B records that have not previously been together in internal memory cannot exceed $\frac{N}{B}$. It follows that after t input operations there are at least $\frac{N!}{\binom{M}{B}^t (B!)^{\frac{n}{B}}}$ orderings consistent with the information obtained from the adversary.

We want to narrow the possible orderings down to 1, and the number of I/O-operations needed to do this must therefore be the least t such that $\frac{N!}{\binom{M}{B}^t (B!)^{\frac{N}{B}}} \leq 1$. Using the (rough) assumptions that $\log x! = x \log x$ (Stirlings formula) and $\log \binom{M}{B} = B \log \frac{M}{B}$ we then get the following.

$$
\begin{array}{rcll}
\frac{N!}{\binom{M}{B}^t (B!)^{\frac{N}{B}}} & \leq & 1 & \Downarrow \\
\binom{M}{B}^t (B!)^{\frac{N}{B}} & \geq & N! & \Downarrow \\
t \log \binom{M}{B} + \frac{N}{B} \log(B!) & \geq & \log(N!) & \Downarrow \\
tB \log(\frac{M}{B}) + \frac{N}{B} B \log B & \geq & N \log N & \Downarrow \\
tB \log(\frac{M}{B}) & \geq & N \log(\frac{N}{B}) & \Downarrow \\
t & \geq & \frac{N}{B} \frac{\log(\frac{N}{B})}{\log(\frac{M}{B})} & \Downarrow \\
t & \geq & n \log_m n &
\end{array}
$$

□