External memory (EM) permuting

Given two input arrays A and π, where A contains N elements and π contains a permutation of $\{1, ..., N\}$, describe and analyze an optimal external-memory algorithm for producing an output array C of N elements such that $C[i] = A[\pi[i]]$ for $1 \leq i \leq N$.

SOLUTION

The algorithm is divided in 5 parts:

Pairs$_\pi$ In this phase we create couples $(\pi[i], i)$ for each $0 \leq i \leq N-1$ and we put them in an array called P. To do so we need to read and write all the elements, therefore we pay $O(\frac{N}{B})$ IOs (actually it should be more or less $4\frac{N}{B} + o(1)$).

Sort$_P$ In this phase we sort P in lexicographic order, using a K-way merge sort. This costs $O(sort(N))$ IOs, that is $O(\frac{N}{B} log_B \frac{N}{B})$.

So, let P' be the (sorted) array projected on the second component only. You see this way $\forall i. P'[i] = \pi^{-1}(i)$, that is P' maps the reverse permutation of $\pi : \forall\, 0 \leq i \leq N-1. i \xrightarrow{\pi(i)} x \xrightarrow{P'(x)} i$ (note i and x are indices, not values).

Pairs$_A$ In this phase we create couples $(P'[i], A[i])$ for each $0 \leq i \leq N-1$ and we put them in an array called B. Again, this costs $O(\frac{N}{B})$.

Sort$_B$ In this phase we sort B in lexicographic order. Again, with a K-way merge sort this costs $O(\frac{N}{B} log_B \frac{N}{B})$.

So, let B' be the (sorted) array projected on the second component only. You see this way $\forall\, 0 \leq i \leq N-1. B'[i] = A[\pi[i]]$, that is to say $B' = C$.

Therefore the cost is dominated by (twice) sorting pairs. For completeness we have a simple example:

$$
\begin{aligned}
A &= [1, 3, 4, 0, 2] \\
\pi &= [3, 4, 0, 2, 1] \\
P &= [(3, 0), (4, 1), (0, 2), (2, 3), (1, 4)] \\
P' &= [(0, 2), (1, 4), (2, 3), (3, 0), (4, 1)] \\
B &= [(2, 1), (4, 3), (3, 4), (0, 0), (1, 2))] \\
B' &= [(0, 0), (1, 2), (2, 1), (3, 4), (4, 3)] \\
C &= [0, 2, 1, 4, 3]
\end{aligned}
$$