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 $C[\pi[i]] = A[i]$ for $1 \leq i \leq N$.

SOLUTION

The algorithm is divided in 3 parts:

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

Sort In this phase we sort the couple, based on the $\pi[i]$ elements, previously created using K-way merge sort. This cost $O(sort(N))$, that is $O(\frac{N}{B} log_B \frac{N}{B})$.

Write C In this phase we write the elements $A[i]$ sorted using π in C. This will take again $O(\frac{N}{B})$ because the pairs are store sequentially in memory.

Therefore the cost is dominated by the sorting of the pairs. For completeness we have simple example:

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