Topic: Generalized External Sorting

Assigned reading: Sections 13.3 and 13.4.2 from Raghu Ramakrishnan's book

In the previous note, we read about external merge sort that can use only three buffer pages. However, if more buffer pages (say B pages) are available then we will like our sorting algorithm to utilize them and sort faster. Here is how we can achieve this goal.

Pass 0: Create CEILING(N/B) runs. Each run is of size B pages. The last run can be of a smaller size.

In each successive pass, we merge consecutive (B-1) runs. We assign one buffer page for each participating run. One page is reserved to store temporary output. In each pass, we read each data page and write it back to the disk. Thus I/O cost of each pass is still 2N. However total number of passes have nowew reduced to ($\mathrm{Log}_{(B-1)}N + 1$).

Double buffering is a technique that aims to reduce the total time of sorting by multiplexing I/O with computation. However, this method results in more number of passes. We will assign 2.K pages for each participating run and temporary output. Thus we can merge only FLOOR((B-2K)/2K) runs at a time. I/O cost of each pass will remain same. However number of passes will increase to ($\mathrm{Log}_{(Y)}N + 1$) where Y= FLOOR((B-2K)/2K).