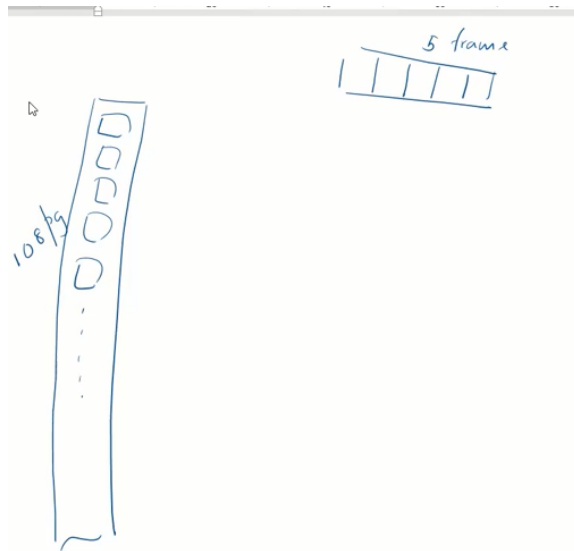


General External Merge Sort

With this algorithm we want to sort the data set. This data set (records or tuples) are in the hard disk / secondary memory. In order to sort them you need to take the data set into the memory. But the problem is, you can just take all the tuples into memory. We'll see how to solve this problem with the External Merge Sort algorithm.

Example

Let's say we have a database file with 108 pages. We need to sort this. We have a buffer in the memory. Buffer size is 5. (5 frames)



Pass 0 ->

Here we're going to get 5 pages from the file and put it into the buffer. Now we can sort that 5 pages and create a file. Again we take the 5 pages and sort it and create another sorted file with those 5 pages. And we continue until all are sorted.

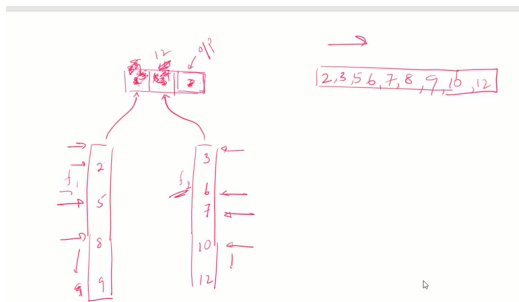
After pass 0 is ended we have $\text{ceil}(108/5) = 22$ sorted files. Each file has 5 pages.

$$\text{Pass-0} \quad \left\lceil \frac{108}{5} \right\rceil = 22 \text{ files}$$

Pass 1 ->

Now we want to merge these files together and create one file.

Note: If we want to merge 2 sorted files we want a buffer with at least 3 frames. We can add one file to one frame, add the other file to another frame. And output a sorted 1 file with the other frame.



Now we have 22 files and a buffer with 5 frames. We are going to put 4 files (Even though we have 5 frames in the buffer we have to keep one frame to get the output, so we only can use 4 frames for merging) into the buffer at a time and create $\text{ceil}(22/4) = 6$ files with $5 \times 4 = 20$ pages. We call this process B-1 merging.

Pass-1 (B-1 way merging)

$$\left\lceil \frac{22}{(5-1)} \right\rceil = 6 \text{ files}$$

Pass 2 -> We still have 6 files. But we want one file. So we continue the B-1 merging.

Pass-2

$$\left\lceil \frac{6}{(5-1)} \right\rceil = 2$$

Pass 3 -> We still have 2 files. So we continue B-1 merging. To merge 2 files, we only need 3 frames. So we don't need to take all the frames.

Pass-3

$$\left\lceil \frac{2}{3} \right\rceil = 1$$

Cost of External Merge sort - if we have N number of pages in a file. We have to read all of the page and sort. After sorting we have to write the same number of pages into secondary memory even its as separate files. So we do this process for P number of passes.

So the cost is,

$$\text{Cost} = 2 \times N \times P$$

Sort merge join ganak with R join S table



$$\begin{aligned}
 \text{Cost}_{SMJ} &= \text{Sorting Cost} + \text{Merging} \\
 &= \text{Sorting Cost}(R) + \text{Sorting Cost}(S) + \text{Merging Cost} \\
 &= 2M \frac{\# \text{pass}}{\uparrow} + 2N \# \text{passes} + (N + M) \\
 &> \underline{\underline{\hspace{2cm}}}
 \end{aligned}$$