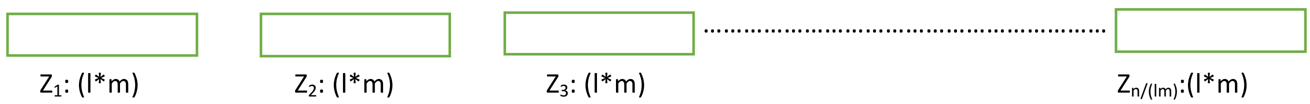
Lecture - 9 (02/21/2018)

Big Data by Prof. Raj.

Documented by: Vinayak Gupta

Professor discussed about the doubts faced by students in the Homework 1.

In the (l,m)-merge algorithm, we shuffle the recursively merged sequences to get a sequence Z. The length of the dirty sequence in Z is no more than lm. We can clean up the dirty sequence as follows:



Z_1 denotes the sequence of the first lm elements of Z; Z_2 denotes the next lm elements of Z; and so on. Thus Z is partitioned into Z_1, Z_2, ………,$Z_{n/(lm)}$.

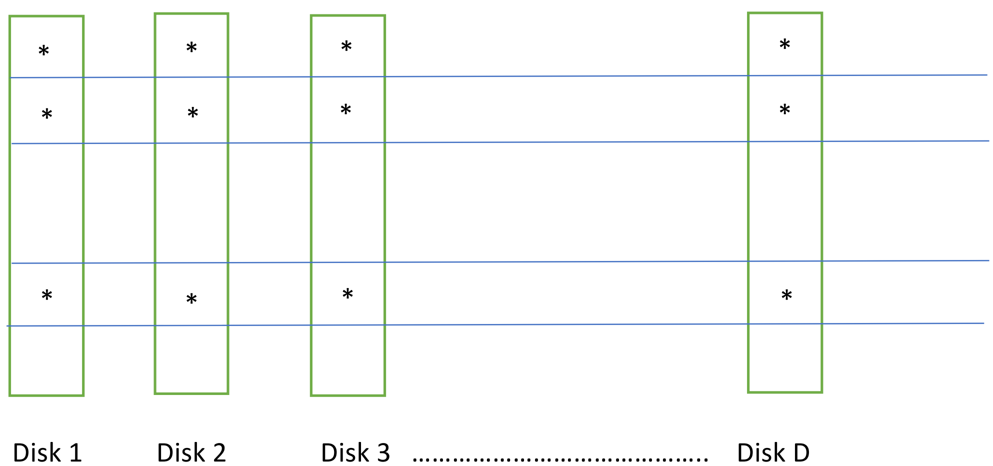
Sort Z_1 and Z_2, send the first lm elements to the disk.

Similarly, sort Z_2 and Z_3, send the first lm elements to the disk; and so on.

Example:

N = $M\sqrt{M}$; D = B = $\sqrt{M}$

Let X = K_1, K_2, K_3,……………… K_n. Input is given across the D disks:



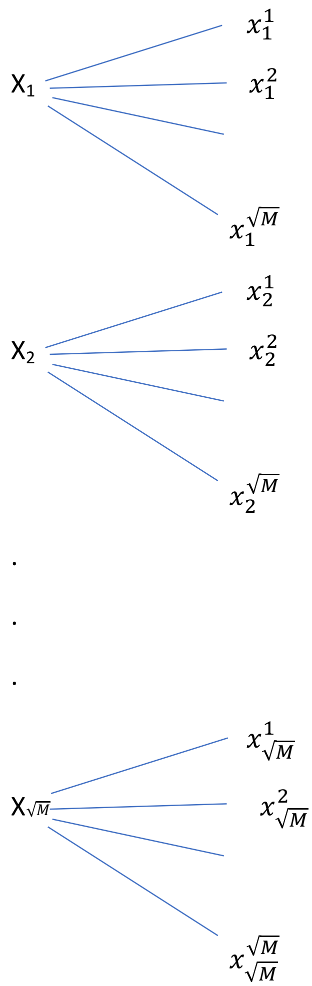
<u>Steps Involved in the algorithm.</u>

<u>Use:</u> l=$\sqrt{M}$; m=$\sqrt{M}$

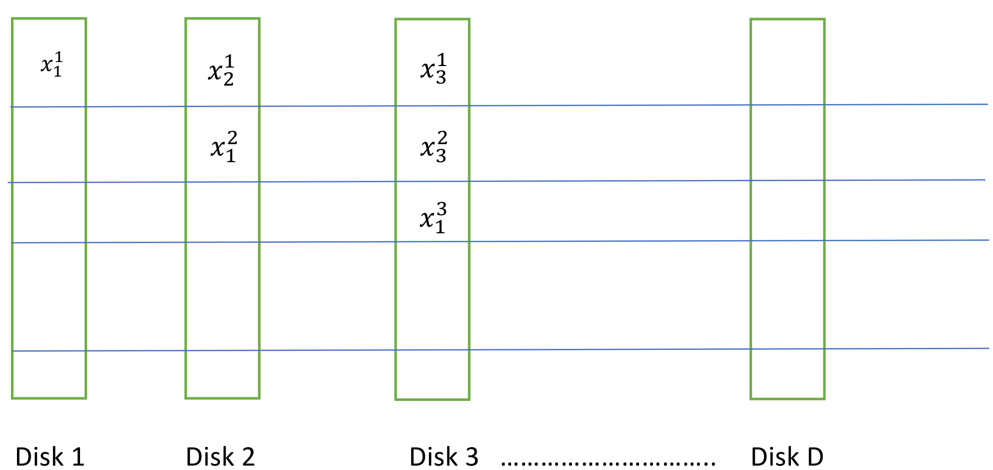
Step 1: Form runs of length M each.

Let these runs be X_1, X_2, X_3 ………………. $X_{\sqrt{M}}$.

Step 2: Unshuffle the above runs into $\sqrt{M}$ parts:



How will we write the unshuffled sequences in the disk? Use the following strategy:



Step 1 and Step 2 can be done in one pass through the data. This is because we can bring M elements to the core memory, sort them, and unshuffle them.

Step 3:

For $1 \leq i \leq \sqrt{M}$ do

Merge $x_1^i,\ x_2^i,\ x_3^i,\ x_4^i \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots . x_{\sqrt{M}}^i$ to get Y_i

This takes one pass through the data.

Step 4:

Shuffle Y_1, Y_2, Y_3…………………. $Y_{\sqrt{M}}$ to get Z;

Step 5:

Clean up the dirty sequence. The length of the dirty sequence ≤ M.

Assume that the core memory is of size 2DB.

Step 4 and Step 5 can be done in one pass together.

Thus the total number of passes = 3.

<u>Chaudhry and Cormen 2002:</u>

We can sort $\frac{M.\sqrt{M}}{2}$ elements in 3 passes through the data when B = D = $\sqrt{M}$.

They have used the column sorting algorithm (of Leighton).

Exercise:

Let T(i, j) be the number of passes needed to merge i sequences of length j each.

Problem 1:

Show that T($\sqrt{M}$,M) = 3 when $\frac{M}{B} \geq \sqrt{M}$.   (Hint : use l = m = $\sqrt{M}$).

Problem 2:

Show that T($\frac{M}{B}, M$) = 3 when $\frac{M}{B} < \sqrt{M}$.   (Hint: Use l = m = $\frac{M}{B}$).

General Algorithm:

Step 1: In one pass through the data form runs of length M each.

Step 2: We have to merge $\frac{N}{M}$ runs of length M each.

What is T($\frac{N}{M}, M$)?

Case 1:

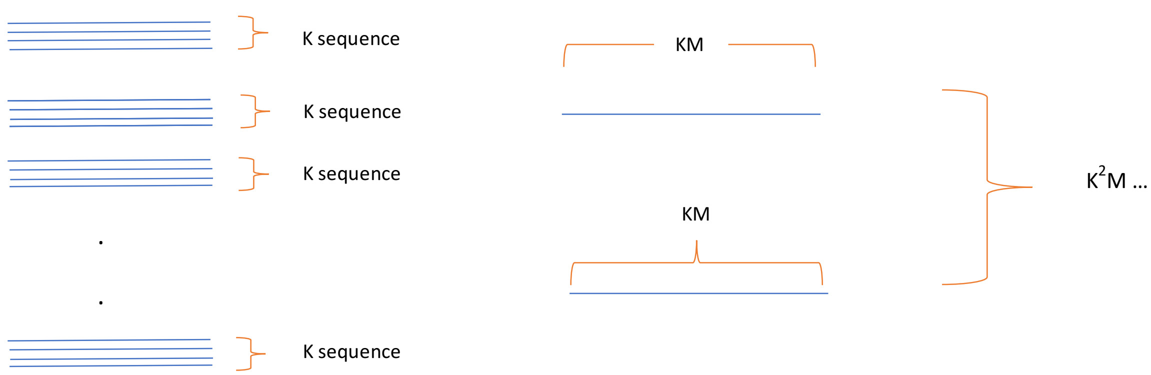
$\frac{M}{B} \geq \sqrt{M}$, we will use l = m = $\sqrt{M}$.

Let K= $\sqrt{M}$, and $\frac{N}{M}$ = K^{2c} = M^{c}

$$c.\log M = \log\left(\frac{N}{M}\right)$$

And $c = \frac{\log(\frac{N}{M})}{\log M}$

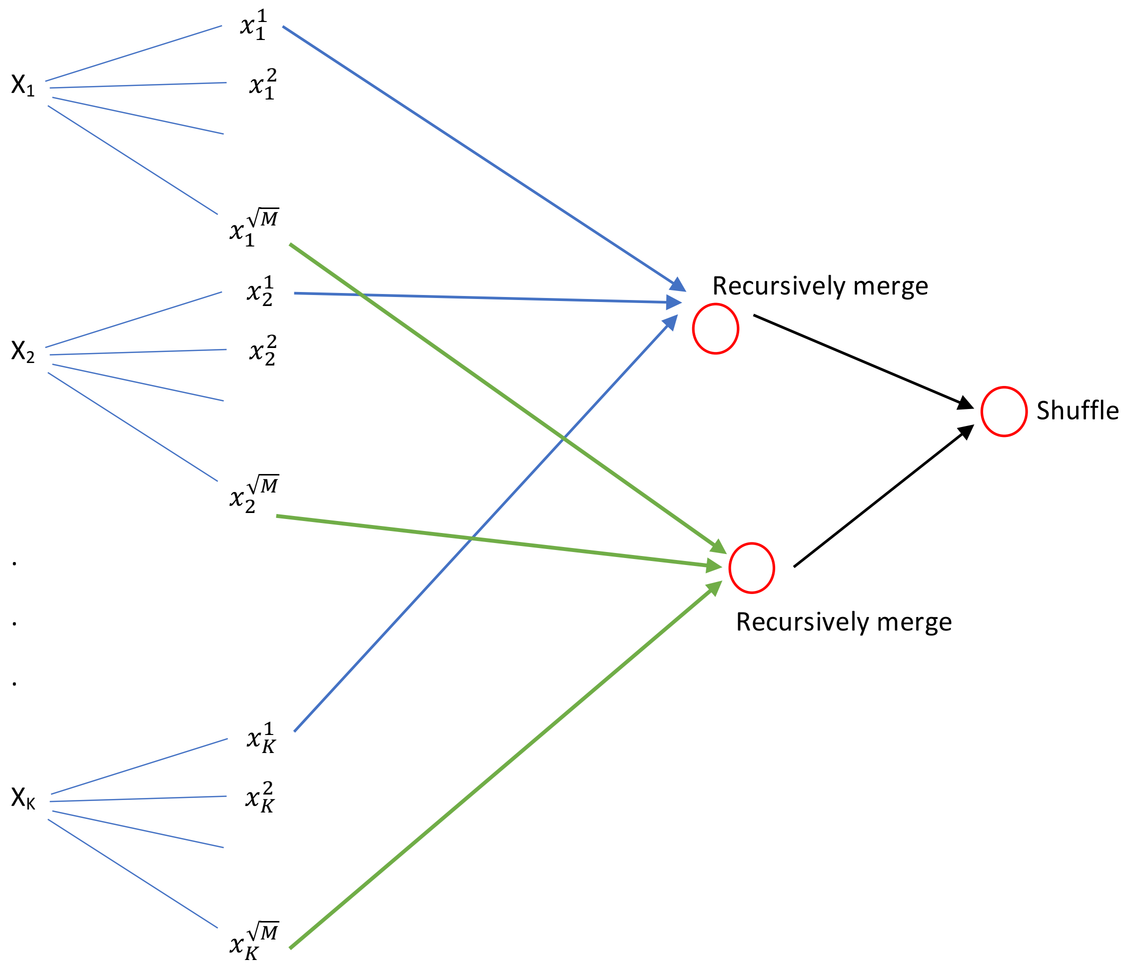
We use K-way merge to merge K^{2c} sequences of length M each. Specifically, we merge K sequences at a time:



We see that:

$T(K^{2c},M)= T(K,M) + T(K,KM) + T(K,K^2M)$…………………………………+ $T(K,K^{2c-1}M)$ Equation(1)

To Compute T(K,K^iM) (for any value of i):



At the last (i.e., the rightmost) node we need to shuffle and clean up the sequence obtained by shuffling.

Note that $|X_p^j|=k^{(i-1)}$ M, for $1 \leq p, j \leq K$.

The unshuffling step requires one pass.

The recursive merging step takes T(K, $K^{(i-1)}$M) passes.

Shuffling and cleaning up can be done in one pass through the data. As a result, we get:

T(K, K^iM)= T(K, $K^{(i-1)}$M) + 2

.

.

= 2i+ T(K, M)= 2i + 3 ——————— Equation (2)

Substitute Equation (2) in Equation (1) to get:

$T(K^{2c}, M) = \sum_{i=0}^{2c-1}(2i + 3)$

$= 2\sum_{i=0}^{2c-1}(i) + 6c$

$= \frac{2*(2c-1)*(2c)}{2} + 6c$

$= 4c^2+4c.$

We'll consider the case of $\frac{M}{B} < \sqrt{M}$ in the next lecture.