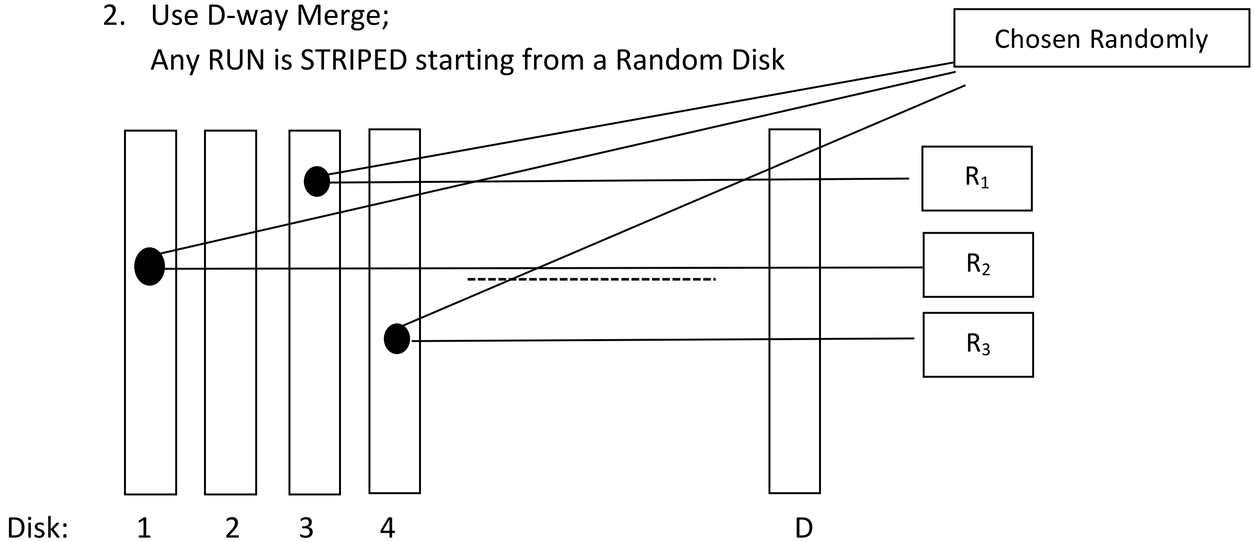
CSE4502/5717: Big Data Analytics

Prof. Sanguthevar Rajasekaran

Lecture 8– 02/19/2018; Notes by Saumya Gupta

- **Simple Randomized Merge Sort (SRM)**
  It is one of the sorting algorithms used in Parallel Disk Model.
  1. Form Runs of length M each.
  2. Use D-way Merge;
     Any RUN is STRIPED starting from a Random Disk



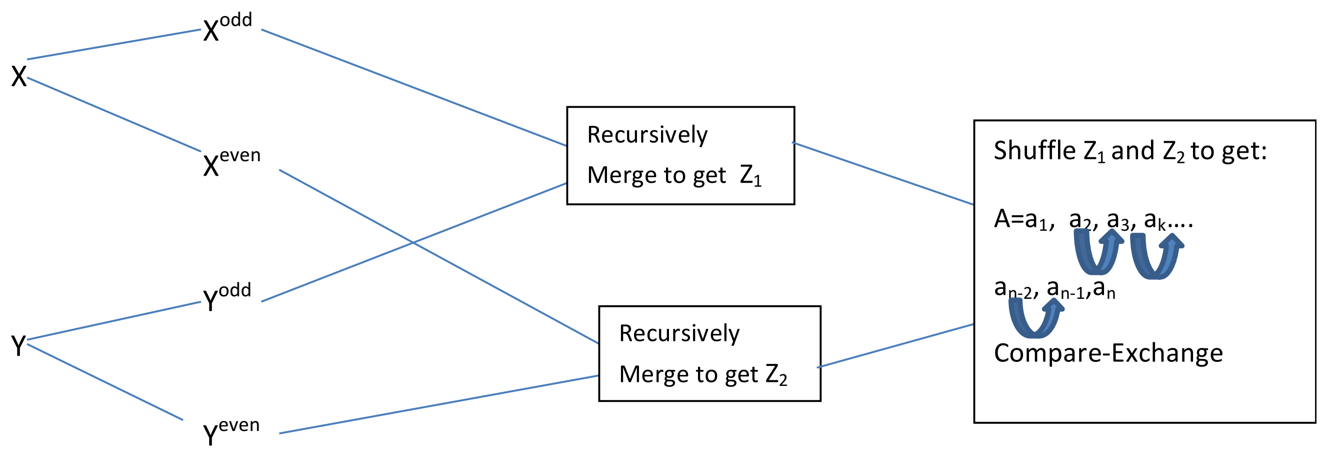
FACT: SRM is asymptotically optimal in expectation if

$$M = \Omega(BD \log D).$$

- **Recap From Last class:**

  We saw ODD-EVEN Merge Algorithm :

  If X and Y are two sorted sequences, with |X|=|Y|=n, we can merge them as follows:



To prove correctness of this algorithm, we will apply 0-1 lemma .

- **PROOF OF CORRECTNESS (USING 0-1 LEMMA)**

  0-1 Lemma states that: If an oblivious comparison-based sorting algorithm correctly sorts every sequence of length n that has only zeros and ones, then, it correctly sorts every sequence of length n of arbitrary elements.

Proof of correctness of the odd-even merge algorithm (using the zero-one lemma):

Let $X=x_1,x_2,x_3,x_4,\ldots\ldots.x_n$ where $x_i=0$ or 1 for all i

Let $Y=y_1,y_2,y_3,y_4,\ldots\ldots.y_n$ where $y_i=0$ or 1 for all i

Let n_1 be the number of zeros in X.

Let n_2 be the number of zeros in Y.

Number of zeros in $Z_1 = \left\lceil \frac{n_1}{2} \right\rceil + \left\lceil \frac{n_2}{2} \right\rceil$

Number of zeros in $Z_2 = \left\lfloor \frac{n_1}{2} \right\rfloor + \left\lfloor \frac{n_2}{2} \right\rfloor$

These two numbers differ by at most 2.

Case 1: Number of zeroes in Z_1 equals the number of zeroes in Z_2.

Z_1 =000 000 … 0011 … 11

Z_2=000 000 … 0011 … 11

Shuffle:

Q= 000000000000 … 00001111 … 1111

(compare-exchange operation not needed - already sorted)

Case 2: These two numbers differ by 1

Z_1 =000 000 … 0001 … 11

Z_2=000 000 … 0011 … 11

Shuffle:

Q= 0000 … 00011111 … 1111

(compare-exchange operation not needed - already sorted)

Case 3: These two numbers differ by 2

Z_1 =000 000 … 0001 … 11

Z_2=000 000 … 0111 … 11

Dirty Sequence

Length of the dirty sequence is at most 2 here

Shuffle:

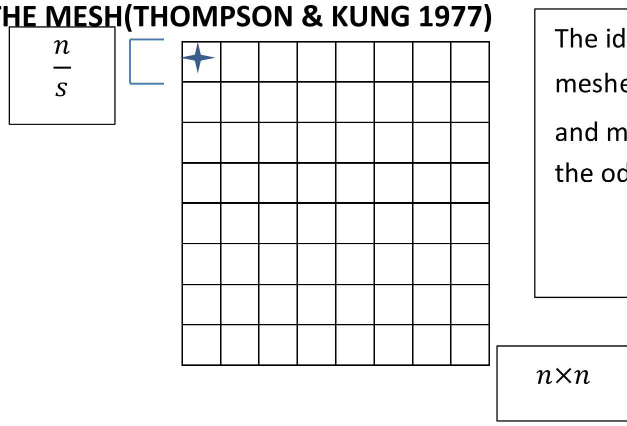
Q= 0000 … 000 10 111….11111

(compare-exchange cleans the boxed 'dirty sequence')

Q= 0000 … 00001111 … 1111

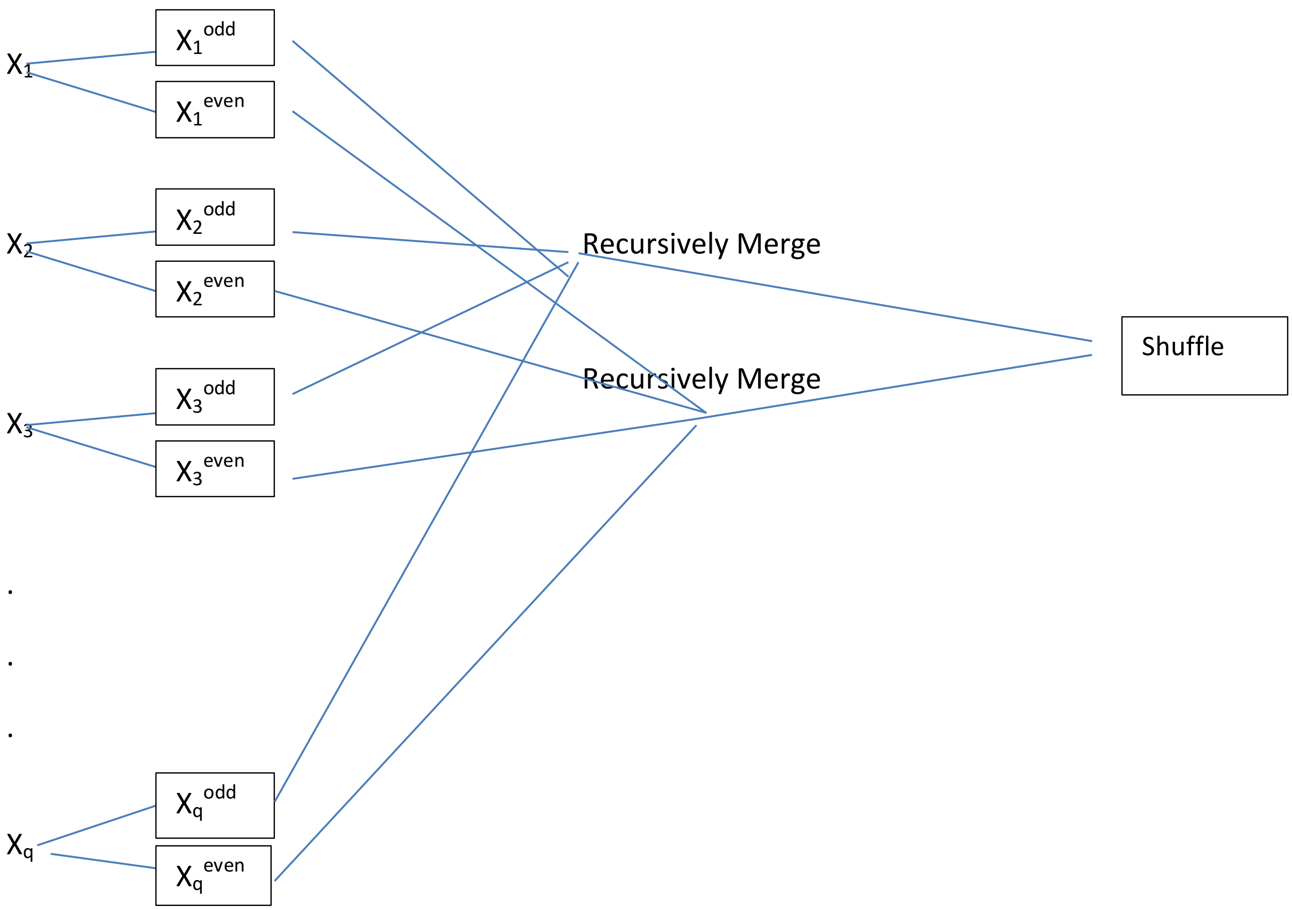
Therefore, in all the three cases, the algorithm works correctly!

- **SORTING ON THE MESH(THOMPSON & KUNG 1977)**



The idea was to partition the mesh into sub meshes of size $\frac{n}{s} \times \frac{n}{s}$, sort each sub mesh, and merge the s^2 sorted sub meshes using the odd-even merge algorithm

<u>S^2–Way MERGE SORT</u> (based on the idea of odd-even merge):



In the above figure, $q=s^2$. It can be shown that the length of the dirty sequence in the shuffled sequence is $<=2s^2$. (To clean up the dirty sequence, some local sorting is used).

- **(l, m) MERGE SORT (Rajasekaran 1999)**

  Input : X = $k_1, k_2, \ldots, k_n$

  Output : Sorted X

  Algorithm :

  1. Partition X into $X_1, X_2, \ldots X_l$ such that $|X_i| = \frac{n}{l}$;

  2. for $1 \leq i \leq l$ do

     Sort each X_i recursively, to get Y_i ;

  3. Merge $Y_1, Y_2, \ldots Y_l$ using the (l, m) Merge Algorithm.

<u>Algorithm (l, m)-Merge</u>

Input : Sorted sequences $Y_1, Y_2, \ldots Y_l$

Output : Merge of $Y_1, Y_2, \ldots Y_l$

Algorithm :

Let $Y_i = y_i^1, y_i^2, \ldots y_i^r$ $1 \leq i \leq l$

for $1 \leq i \leq l$ do

Unshuffle Y_i into m parts: $Y_i^1, Y_i^2, \ldots Y_i^m$

if $Y_i = y_i^1, y_i^2, \ldots y_i^r$

then $Y_i^1 = y_i^1, y_i^{m+1}, y_i^{2m+1}, \ldots$

$Y_i^2 = y_i^2, y_i^{m+2}, y_i^{2m+2}, \ldots$

.

.

$Y_i^m = y_i^m, y_i^{2m}, y_i^{3m}, \ldots$

1. for $1 \leq i \leq m$ do

   Recursively merge $Y_1^i\ Y_2^i, \ldots, Y_l^i$ to get $Z_i = Z_1, Z_2, \ldots$

   Merge $Y_1^1, Y_2^1, \ldots Y_l^1$; to get Z_1

   Merge $Y_1^2, Y_2^2, \ldots Y_l^2$; to get Z_2

   .

   .

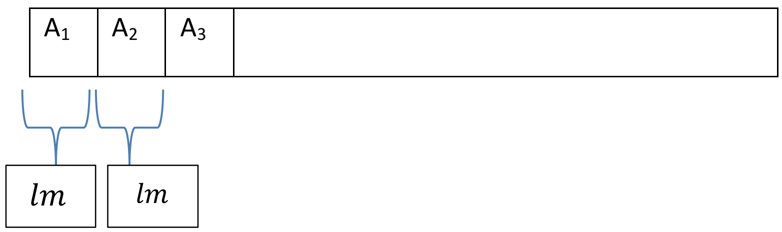
   Merge $Y_1^m, Y_2^m, \ldots Y_l^m$; to get Z_m

$$| Z_i | = \frac{lr}{m} = \frac{n}{m}$$

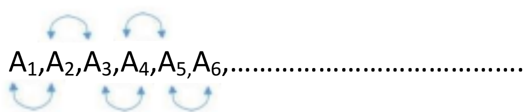
2. Shuffle $Z_1, Z_2, \ldots Z_m$ to get

   $A = a_1, a_2, \ldots a_n$

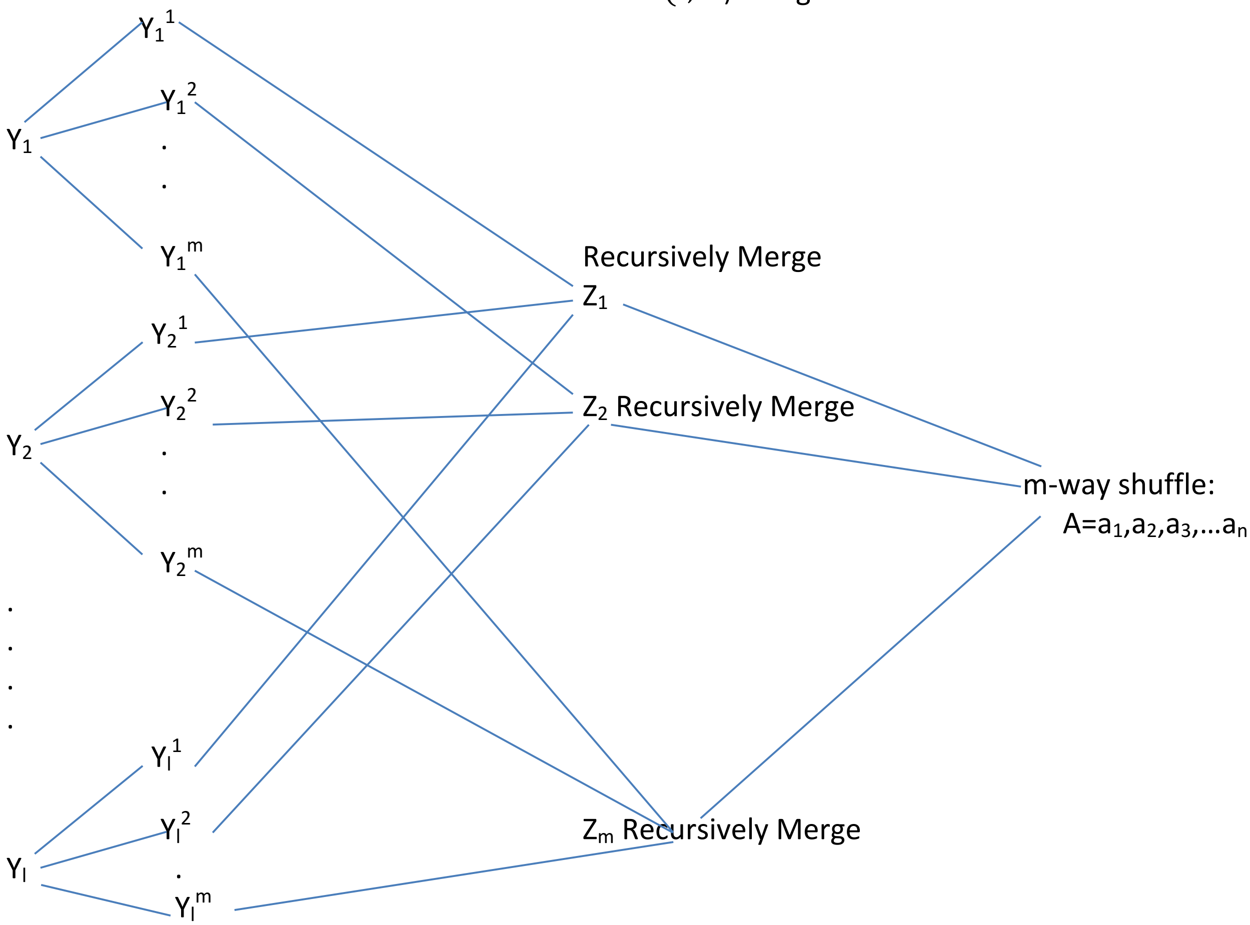
3. Partition A into blocks of size lm each



- Sort and Merge A_1 & A_2 ; A_3 & A_4;…………..
- Sort and Merge A_2 & A_3; A_4 & A_5;……………



Demonstration of (l, m)-Merge:



<u>NOTE:</u> The length of the Dirty Sequence is $\leq lm$.

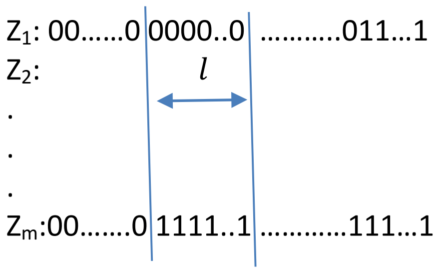
<u>To Prove correctness of (l, m)-merge algorithm, we can apply 0-1 lemma:</u>

Let n_i be the number of zeroes in Y_i , $1 \leq i \leq l$

Number of zeroes in Z_1: $\sum_{i=1}^{l} \left\lceil \frac{n_i}{m} \right\rceil$

Number of zeroes in Z_m: $\sum_{i=1}^{l} \left\lfloor \frac{n_i}{m} \right\rfloor$

These two numbers differ by at most l.



The number of columns that contribute to the dirty sequence is at most l, and they contain at most lm elements.

Therefore, the length of dirty sequence is $\leq l$m.

Next, we will apply the <u>(l, m)-merge sort algorithm</u> to the Parallel Disk Model and calculate the number of I/O operations.