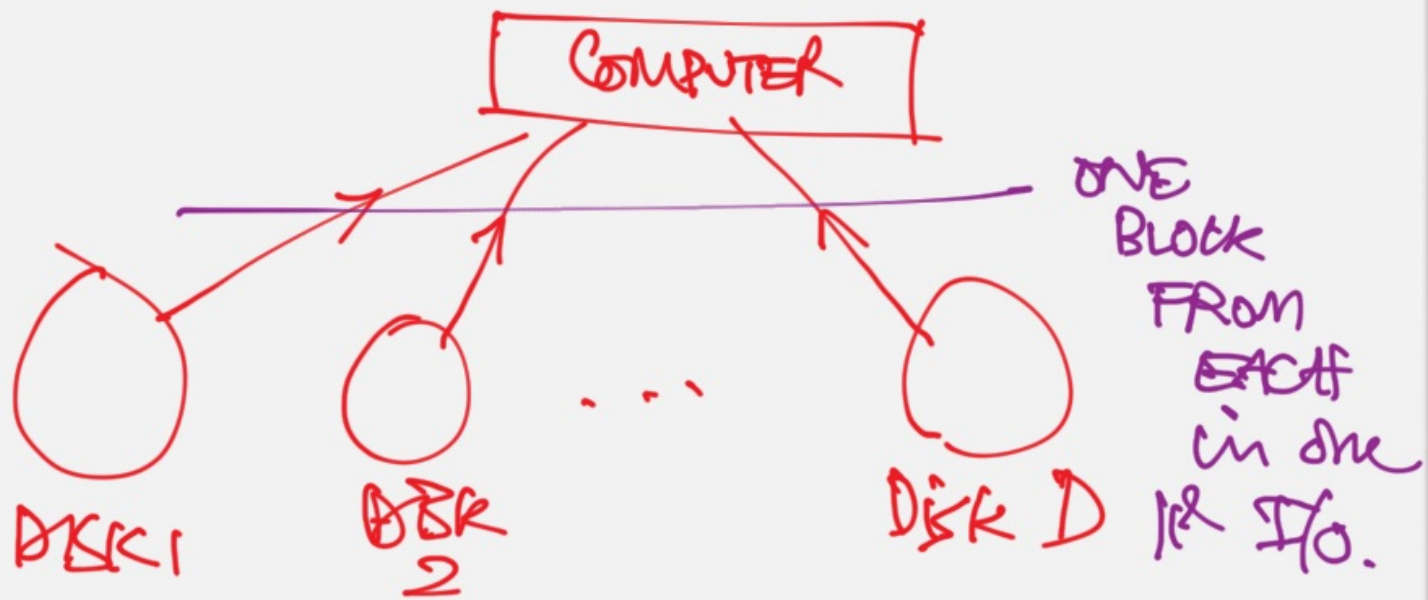
CSE 4502/5717

BIG DATA ANALYTICS

LECTURE ON 9-29-22

HW1 STAT: $\mu = 87.26$; $\sigma = 13.13$.

PARALLEL DISKS MODEL (PDM):



ONE BLOCK FROM EACH in one IO.

IN ONE PARALLEL I/O we BRING BD elements FROM the disks to the Computer.

ONE PASS → $\frac{n}{BD}$ PARALLEL I/O OPERATIONS.

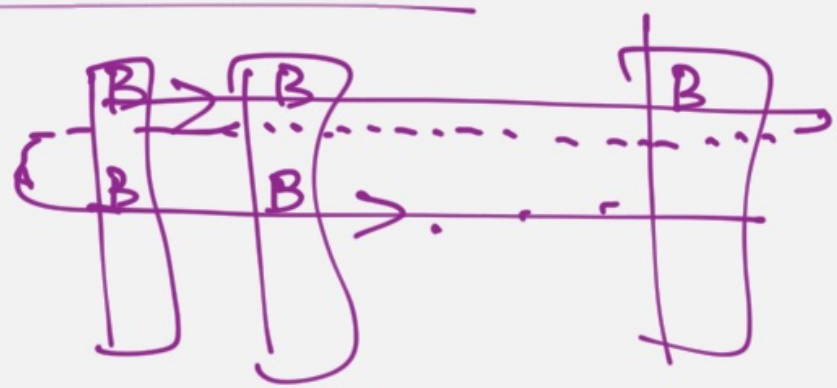
PROBLEM: SORTING n elements.

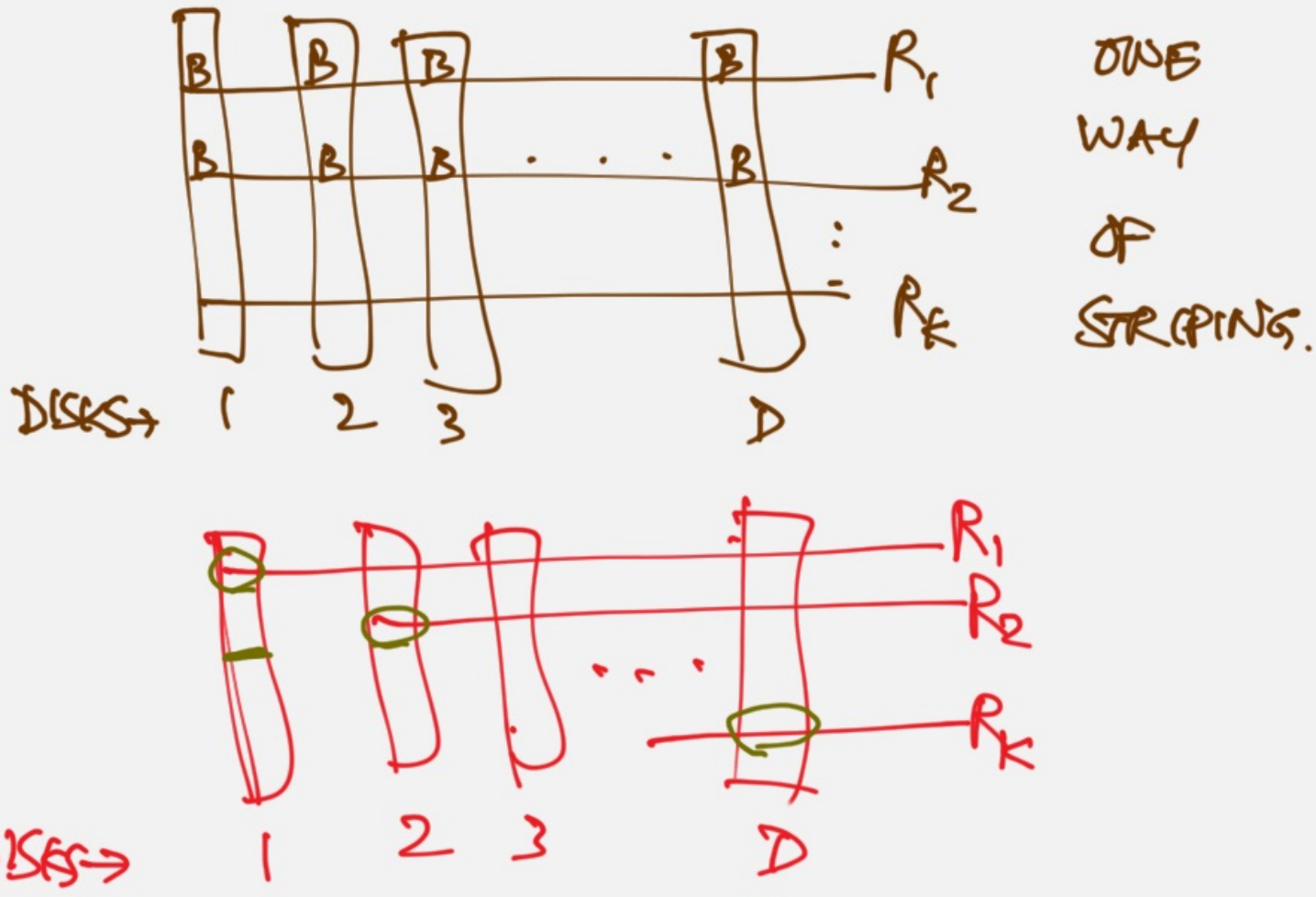
There will be $\frac{n}{D}$ input elements in EACH NODE.

IN the output, the smallest $\frac{n}{D}$ elements should be in the FIRST DISK.

The next $\frac{n}{D}$ smallest elements should be in the second disk, etc.

Another Convention: STRIPING DATA.





ONE WAY OF STRIPING.

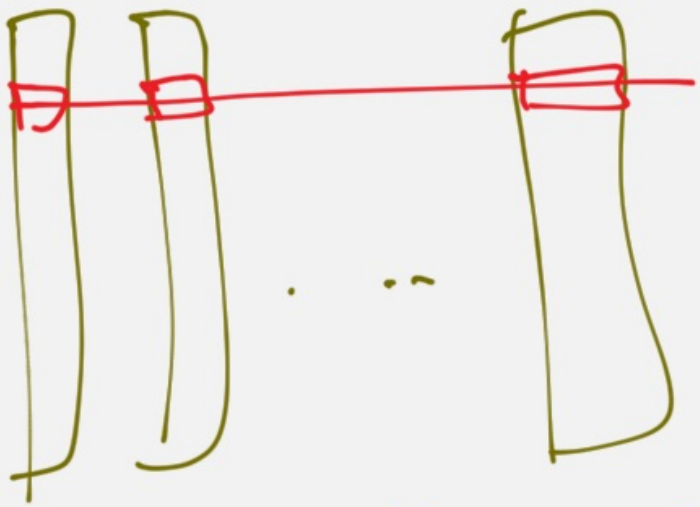
PROBLEM: SORTING. INPUT: $X = k_1, k_2, \ldots, k_n$.

THEOREM: ANY SORTING ALGORITHM ON A PDM WITH D DISKS NEEDS

$$\Omega\left(\frac{n}{BD}\,\frac{\log(n/M)}{\log\left(\frac{M}{B}\right)}\right)$$

I/O OPERATIONS.

DISK STRIPED MERGE SORT (DSM SORT):



TREAT the DISKS AS A SINGLE DISK with a BLOCK SIZE of BD.

IT USES $\frac{M}{BD}$ - WAY MERGE.

STRATEGY:

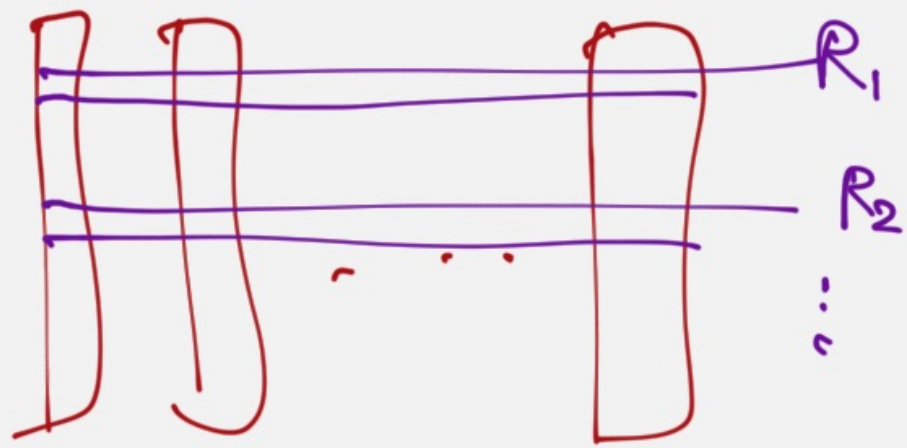
1. FORM RUNS OF length M EACH. this takes ONE PASS.

2. MERGE the $\frac{N}{M}$ RUNS USING R-WAY MERGE FOR SOME RELEVANT R.

FOR DSM, $R = \frac{M}{BD}$.

GENERAL ASSUMPTION: $M = \Theta(BD)$.

At any time BRING BD ELEMENTS FROM EACH OF the RUNS to the CORE & START MERGING.



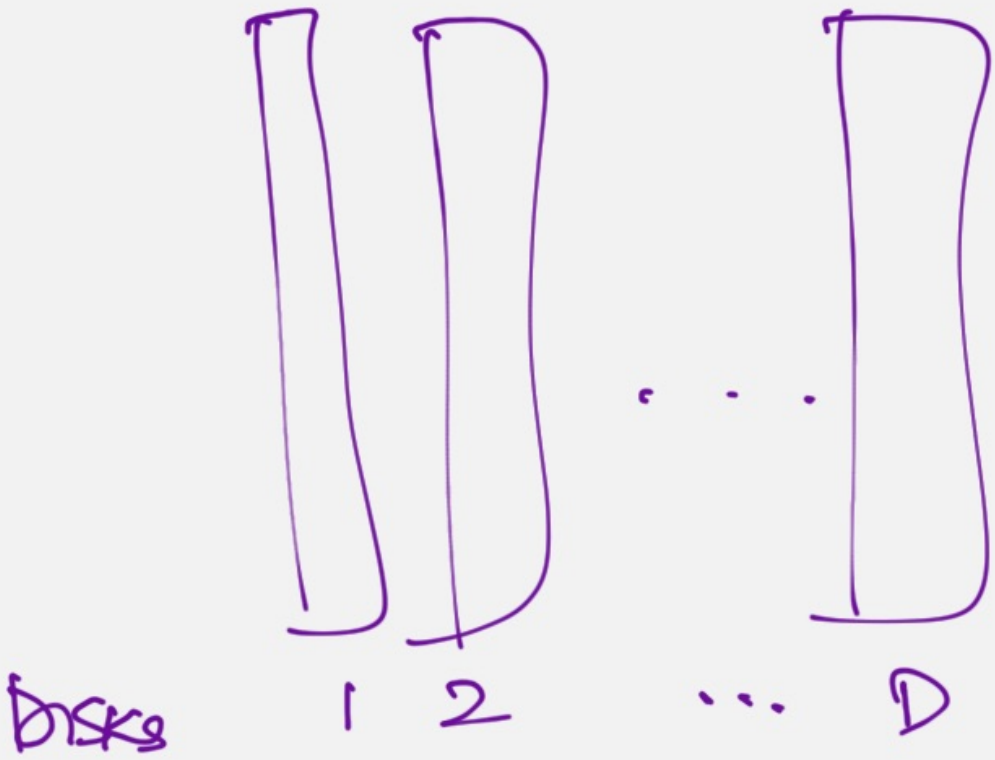
#8 IR IO OPERATIONS

$$= O\left[\frac{n}{BD}\left(\frac{\log\left(\frac{n}{M}\right)}{\log\left(\frac{M}{BD}\right)} + 1\right)\right]$$

(R. BARVE & J. VITTER 1996):

SIMPLE RANDOMIZED MERGE SORT (SRM SORT).

USES D-WAY MERGE.



ANY RUN IS STRIPED ACROSS the DISKS STARTING FROM A RANDOM DISK.

They show that SRM is OPTIMAL ON AN AVERAGE IF $M = \Omega(BD \log D)$.

(l, m)-MERGE SORT

(RAJASEKARAN 1996).

BATCHER'S ODD-EVEN MERGE SORT:

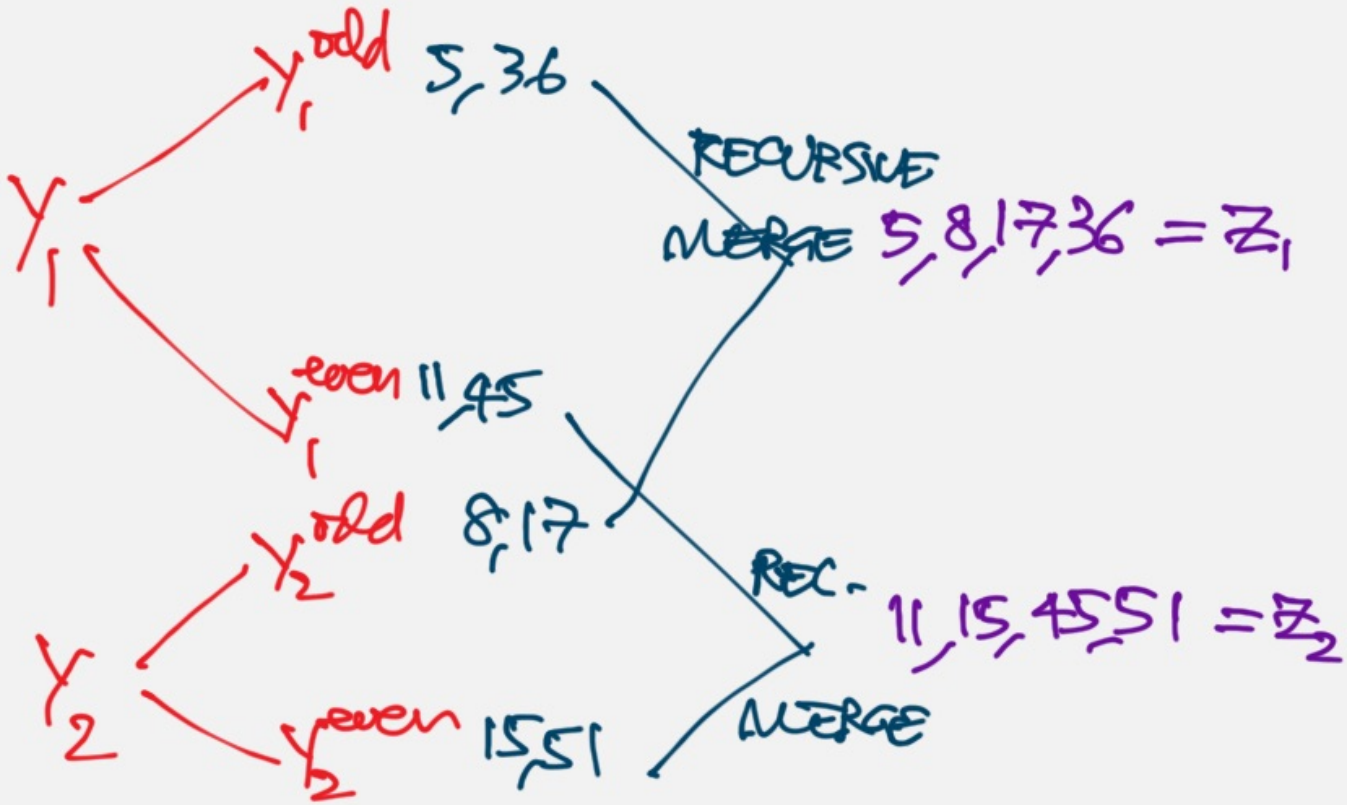
INPUT: $X = k_1, k_2, \ldots, k_n$.

ALGORITHM:

0. Partition X into TWO:
   $X_1 = k_1, k_2, \ldots, k_{\frac{n}{2}}$; $X_2 = k_{\frac{n}{2}+1}, \ldots, k_n$
1. RECURSIVELY SORT X_1 to get Y_1;
   RECURSIVELY SORT X_2 to get Y_2;
2. MERGE Y_1 and Y_2 USING ODD-EVEN MERGE.

ODD-EVEN MERGE:

$Y_1 = 5, 11, 36, 45$; $Y_2 = 8, 15, 17, 51$



SHUFFLE Z_1 and Z_2.

5, 11, 8, 15, 17, 45, 36, 51

← ONE COMPARISON EXCHANGE

5, 8, 11, 15, 17, 36, 45, 51

GENERAL ALGORITHM: MERGE Y_1 and Y_2.

(1) Partition Y_1 into Y_1^{odd} & Y_1^{even}

Partition Y_2 into Y_2^{odd} & Y_2^{even}

(2) RECURSIVELY MERGE Y_1^{odd} & Y_2^{odd} to get Z_1;

Recursively MERGE Y_1^{even} & Y_2^{even} to get Z_2.

(3) Shuffle Z_1 and Z_2 to get Q.

(4) Do ONE COMPARISON-EXCHANGE OF NEIGHBORS STARTING FROM POSITION 2.

ZERO-ONE LEMMA:

IF A COMPARISON BASED OBLIVIOUS SORTING ALGORITHM CORRECTLY SORTS every sequence of zeros and ones of length n, then it also correctly sorts any sequence of arbitrary elements of length n.

PROOF OF CORRECTNESS (ODD_EVEN MERGE):

Assume that Both Y_1 and Y_2 have only zeros and ones. $|Y_1| = n = |Y_2|$.

Let the # of ZEROS in Y_1 be n_1 and the # of ZEROS in Y_2 be n_2.

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

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

$\Rightarrow$ # of ZEROS in Z_1 is AT MOST TWO MORE than the # of ZEROS in Z_2.

<u>CASE 1</u>: # of ZEROS in Z_1 and Z_2 are the SAME.

$Z_1 \rightarrow$ 000...0111...1

$Z_2 \rightarrow$ 000...0111...1

$\Downarrow$ SHUFFLE

000000...00111111...11

This sequence is already sorted!

CASE 2: # of ZEROS IN Z_1 = # of ZEROS in Z_2 + 1.

$Z_1 \rightarrow$ 000 ··· 00 1111 ··· 1

$Z_2 \rightarrow$ 000 ··· 0 1111 ··· 1

$\Downarrow$ Shuffle

000000 ··· 000 11111111 ··· 11

THE SEQUENCE IS ALREADY SORTED!

CASE 3: # of ZEROS IN Z_1 = # of ZEROS in Z_2 + 2.

Z_1: 000 ··· 000111 ··· 1

Z_2: 000 ··· 011111 ··· 1

⇓ Shuffle

00000 00 ··· 00 0 1 0 1111111 ··· 11

DIRTY SEQ.

THE SEQ. GETS CLEANED with a Compare-Exchange!