# CSL 356, July 30

Can't sort *n numbers* faster than $\Omega(n \log n)$

comparisons — not div, mult.

Sort $n$ numbers in the range $[1 .. n]$

Input $x_1, x_2 \ldots x_n$

$x_i \in [1 \ldots n]$

Can sort using "count sort"

count the #1's, #2's ... #n's

counters for $i$, $1 \le i \le n$

increment the counter corresponding to $x_i$

| | 1 | 2 | 3 | 4 | . | $i$ | $n$ |
|---|---|---|---|---|---|---|---|
| count | 5 | 2 | 0 | | | | |

Output: $\underbrace{11111}_{5} \mid \underbrace{22}_{2} \mid 44 \mid \quad \underbrace{i}$

count[1] , count[1] + count[2] ,
count[1] + count[2] + count[3]

Suppose inputs are $y_1, y_2, y_3 \ldots y_n$

① $y_1$

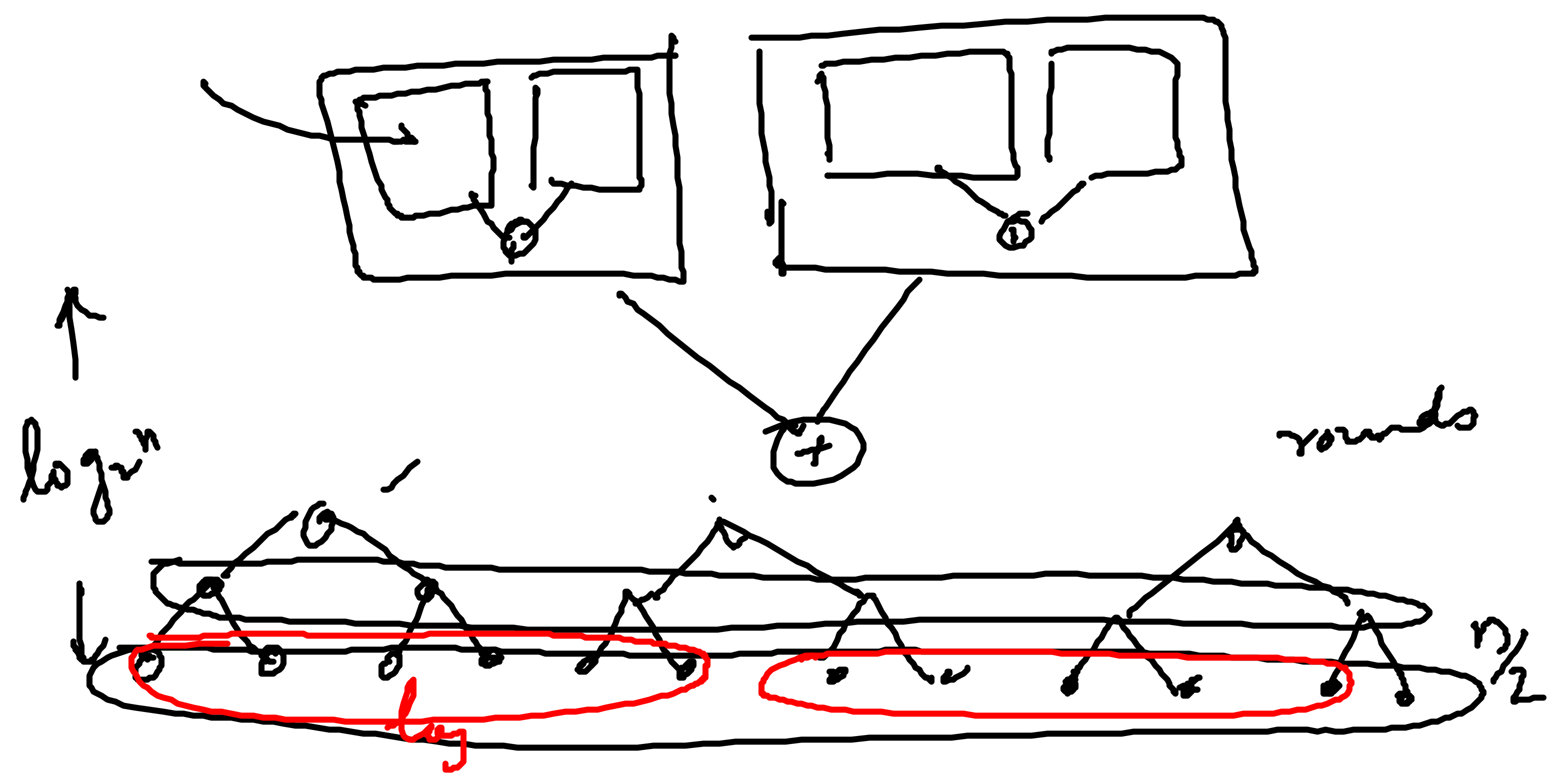
② $y_1 + y_2$ partial sums

$\sum_{j=1}^{i} y_j$ ✓

⋮

ⓝ $y_1 + y_2 + \cdots y_n$ ✓

## Parallel computation of partial sums



Using $\frac{n}{2}$ processors, we can sum $n$ numbers in $\log_2 n$ rounds

**Total effort**: #processors × #rounds

Total work $O(n \log n)$

Always compare with Total sequential time.

Use $\frac{n}{\log n}$ processors.

Initially each processor adds up $\log n$ numbers each sequentially

Total time $O(\log n)$

Subsequently we have $\frac{n}{\log n}$ no.s and $\frac{n}{\log n}$ processors

Use the tree based computation

$\Rightarrow \log\left(\frac{n}{\log n}\right)$ rounds $\quad \frac{n}{\log n}$ processors

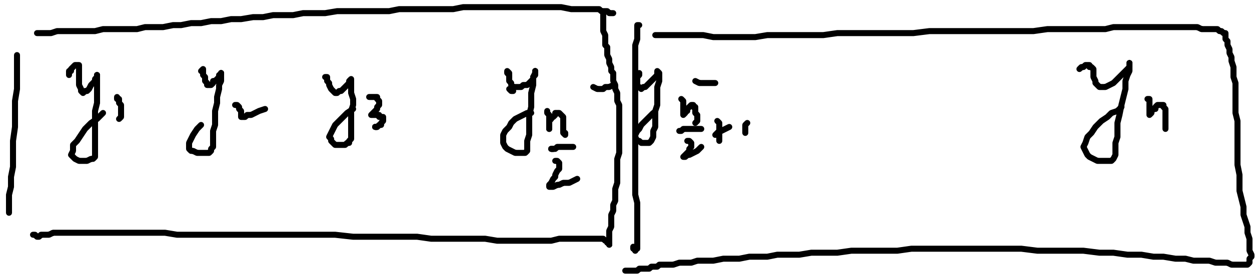
$O(\log n) \times \frac{n}{\log n} = O(n)$

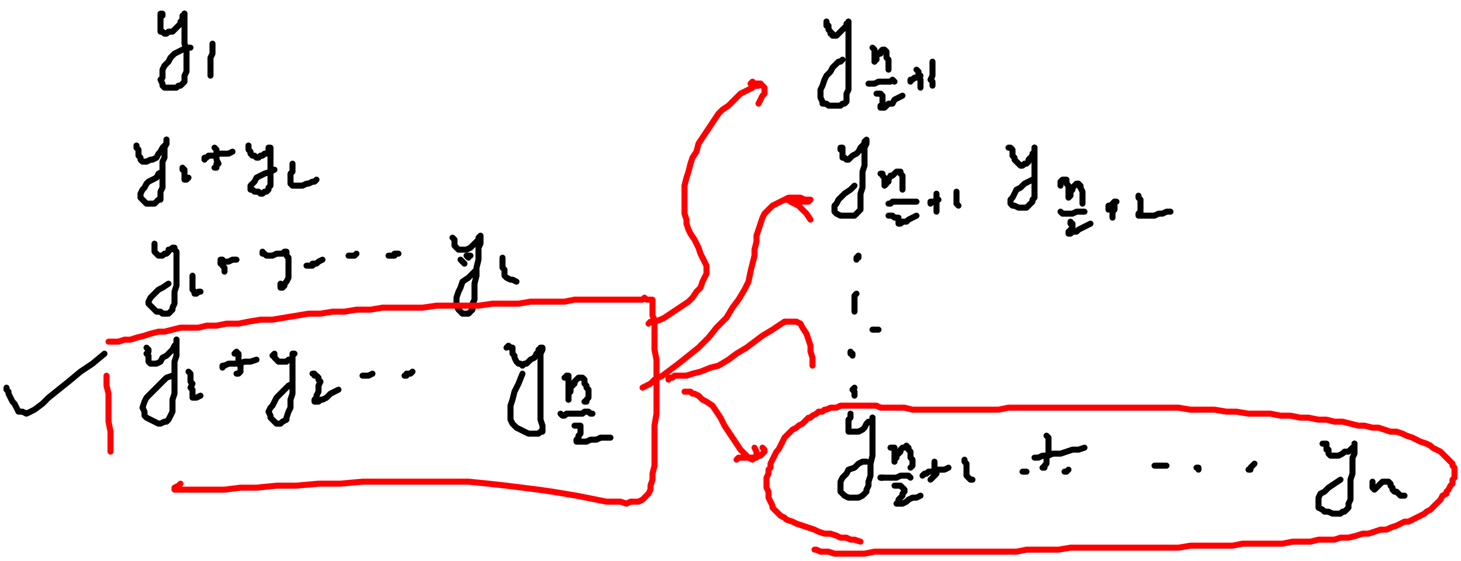
2 log n parallel steps

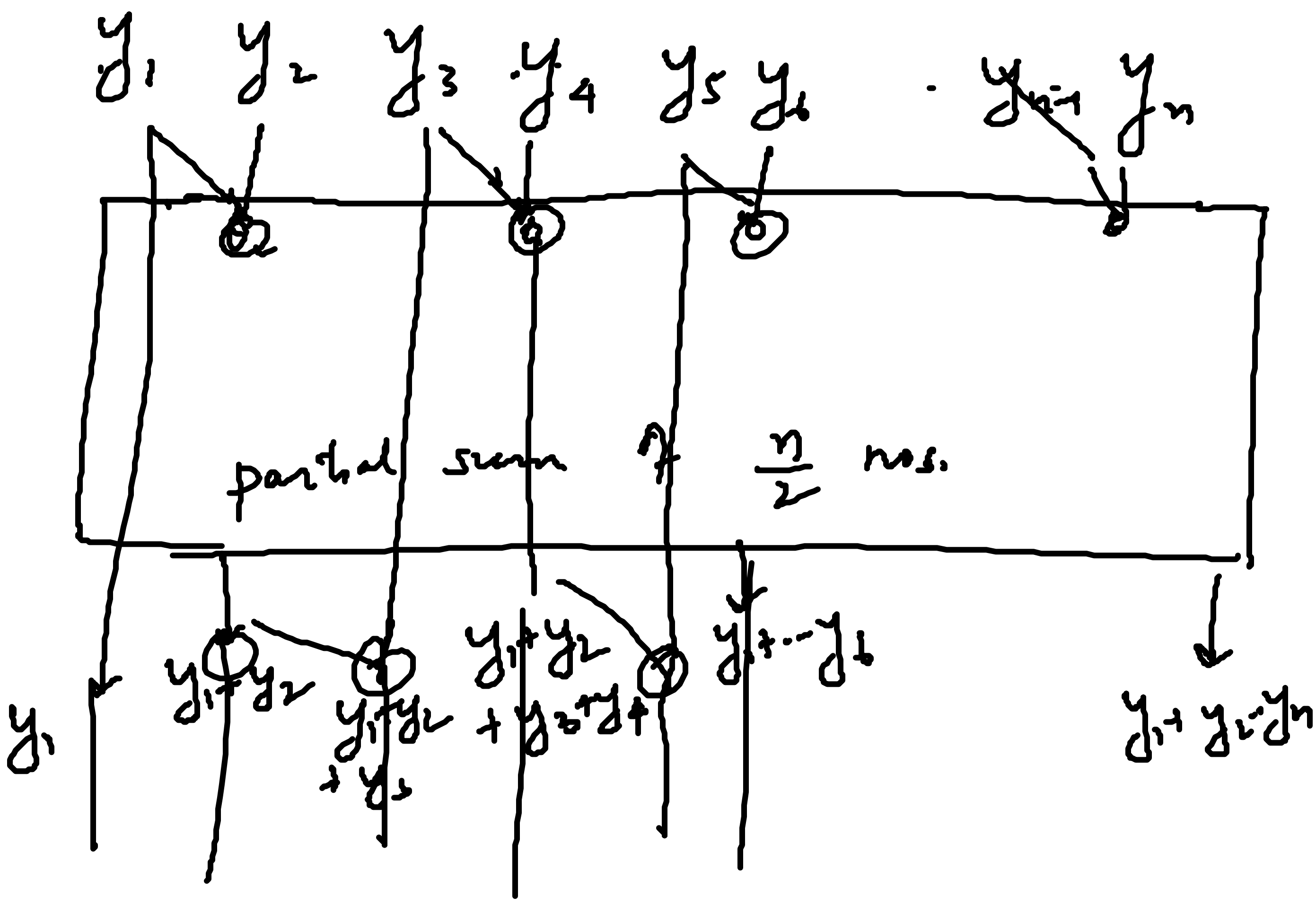
For partial sums, if we employ different sets of processors for each term, then $\Rightarrow \frac{n}{\log n} + \frac{n-1}{\log n} \cdots$

$$= O\left(\frac{n^2}{\log n}\right)$$

Total work $\quad \frac{n^2}{\log n} \times \log n = O(n^2)$


y1 y2 y3 yn/2
yn/2+1 yn


y1
y1+y2
y1+y2 ... y
y1+y2 ... yn/2
yn/2+1
yn/2+1 yn/2+2
yn/2+1 + ... yn


y1 y2 y3 y4 y5 y6 . yn-1 yn
partial sum of n/2 nos.
y1
y1+y2
y1+y2 + y3
y1+y2
y1+y2 + y3+y4
y1+...y6
y1+y2...yn

What is time

What is the # addition operations
( adder circuits )

$$T^{\parallel}(n) = T^{\parallel}(n/2) + 2$$

$$\Rightarrow T^{\parallel}(n) = 2 \log n$$

$$S(n) = S\left(\frac{n}{2}\right) + n$$

$$\Rightarrow S(n) \leq 2n \text{ ie. } O(n)$$

parallel prefix / scan operation

It generalises to any "associative" operation

$y_1 \odot y_2 \odot_k$

associative