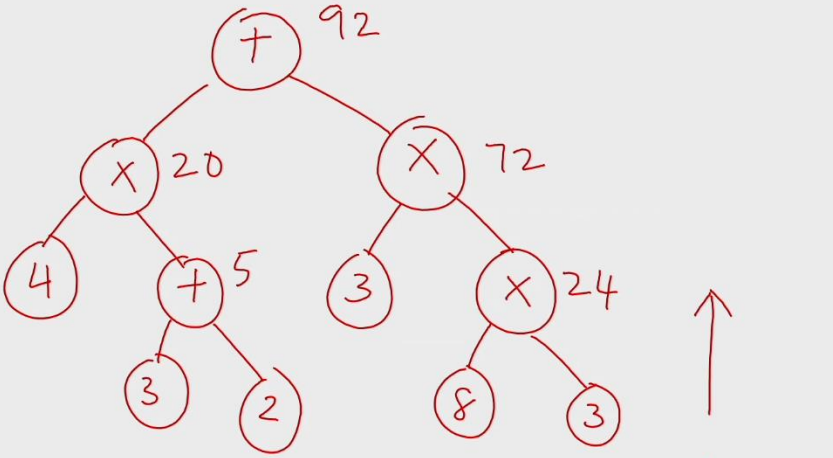
Expression tree

internal nodes with operations $\{+, \times\}$

leaf nodes have integer

evaluate the root



$O(n)$ time sequentially

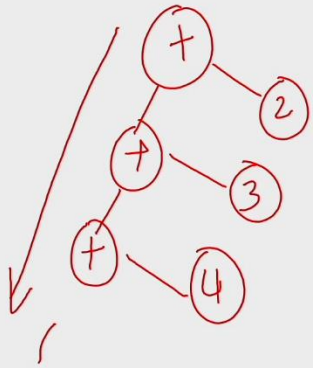
Bottom up

parallel solution

O(depth) time

could be O(n)

not satisfactory



Tree Contraction

every vertex in the expression tree is given a label of (1,0) initially

ordered pair

node v (a_v, b_v)

value sent up by v

is $(a_v X + b_v) = X$

$a_v = 1$, $b_v = 0$

Perform several steps

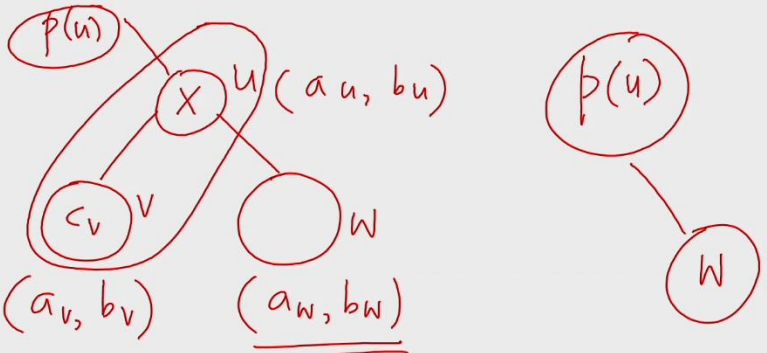
each step executes

independent rake ops

Labels of the remaining nodes

change, so that the tree

still gives the same val



$(a_v c_v + b_v)$: v sends to u.

w sends a value of $a_w X + b_w$ to u.

$$a_u \left[(a_v c_v + b_v) * (a_w X + b_w) \right] + b_u$$

$$= a_u (a_v c_v + b_v) a_w (X) + a_u (a_v c_v + b_v) b_w + b_u$$

$\Big\}$ $a'_w X + b_w$

w sends a value of
$a_w X + b_w$ to u.

$$a_u\left[(a_v c_v + b_v) \overset{+}{*} (a_w X + b_w)\right] + b_u$$

$$= a_u (a_v c_v + b_v) a_w (X) + \left.\begin{array}{l} \\ a_u (a_v c_v + b_v) b_w + b_u \end{array}\right\} \begin{array}{l} a'_w X \\ + \\ b_w \end{array}$$

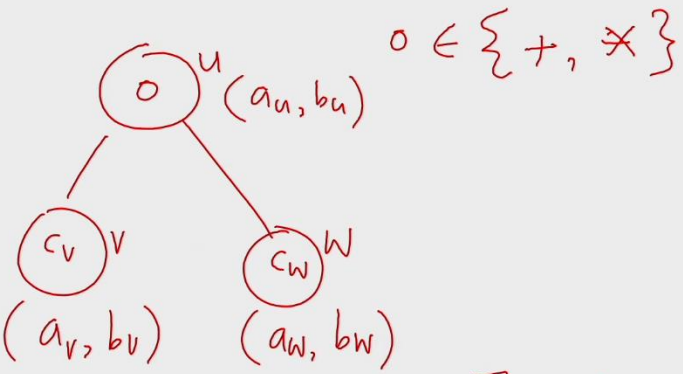
exactly one rake operation with relabelling as suggested ensures the same val at the root

We have many independent rake ops.

each iteration

① Add elements that are left children

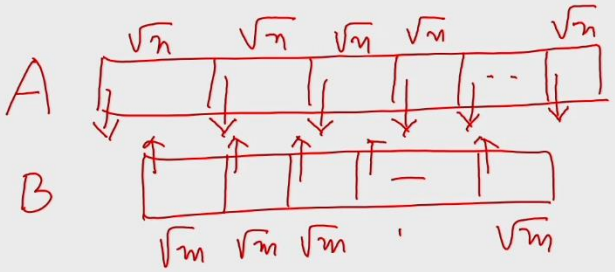
② All the remaining Add elements



$$a_u\left[(a_v c_v + b_v) \circ (a_w c_w + b_w)\right] + b_u$$

$O(\log n)$ time
$n/\log n$ processors
on an EREW PRAM

Merge Algorithm
that runs in
$O(\log\log n)$ time



from $A \longrightarrow A'$ array of leaders

$|A'| : \sqrt{n}$

$B \longrightarrow B' \quad |B'| : \sqrt{m}$

$n+m$ processors

$$\sqrt{n}\sqrt{m} < n+m$$

✓ $2nm < (n+m)^2$

$2nm < n^2 + 2nm + m^2$

✓ $0 < n^2 + m^2$

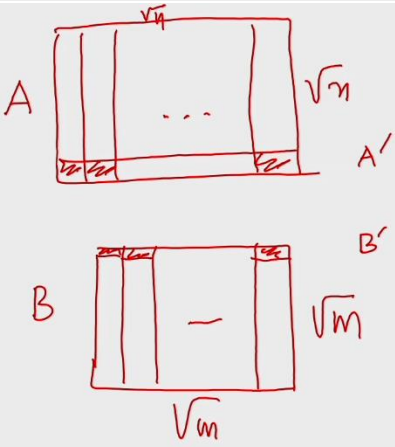
$nm < (n+m)^2$

$2\sqrt{n}\sqrt{m}$ processors with us

$|A'| = \sqrt{n}$ $\quad$ $|B'| = \sqrt{m}$

↓ assign $\sqrt{m}$ processors/leader $\quad$ ↓ assign $\sqrt{n}$ processors/leader

$\sqrt{n}\sqrt{m}$ $\quad$ $\sqrt{n}\sqrt{m}$



$x \in A'$

x has $\sqrt{m}$ procrs ✓

search (B', x)

$O(1)$ time

$|B'| = \sqrt{m}$

A' and B' are now cross ranked.

$x \in A'$: $r_B[x]$

x falls between the $r_B[x]^{th}$ element & $r_B[x]+1^{st}$ element of B'

rank the leaders in the array on the other side

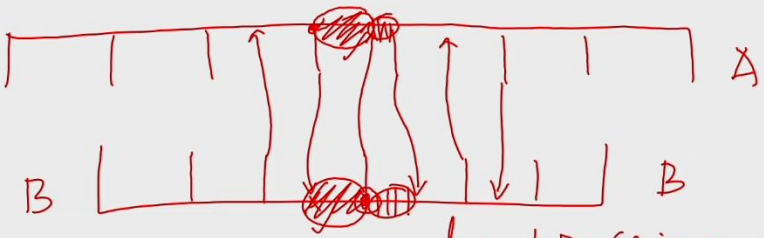
$A' \rightarrow B$ & $B' \rightarrow A$

$O(1)$ time

merge (A', B')

: the sorted ~~list~~ array of all leaders

$B' \rightarrow A$ $A' \rightarrow B$



combs pressing against each other

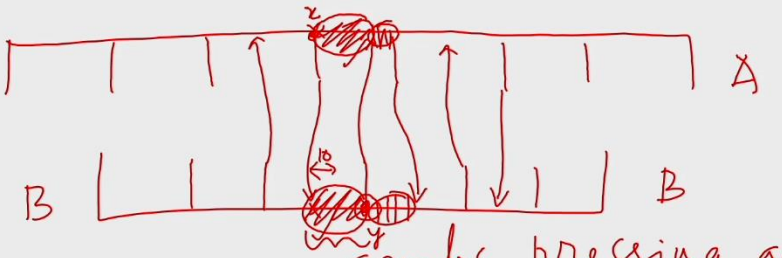
teeth break the spines

$\sqrt{n} + \sqrt{m}$

Solve the subproblems
recursively

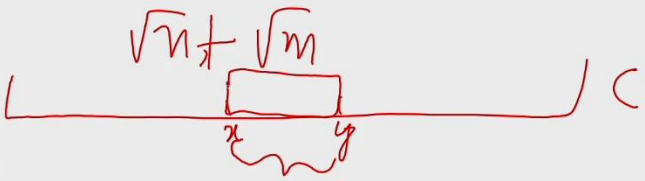
ensure that each processors
knows its offset within
its subproblem

$> \sqrt{n}\sqrt{m}$ processors



combs pressing against
each other

teeth break the spines $|C| = |A| + |B|$



Recursion is valid.

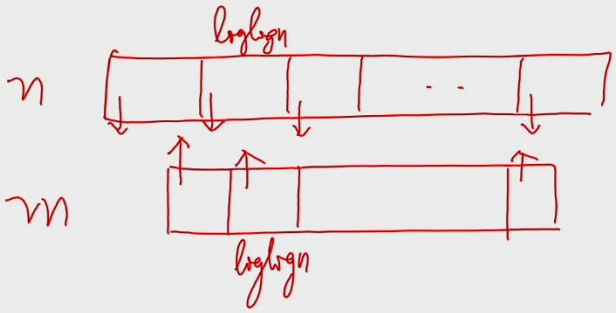
Time of the algorithm

$$T(n,m) = T(\sqrt{n},\sqrt{m}) + 1$$
$$= T(\sqrt{\sqrt{n}},\sqrt{\sqrt{m}}) + 2$$
$$= T(n^{1/8}, m^{1/8}) + 3$$

$$= T(n^{1/2^k}, m^{1/2^k}) + k$$

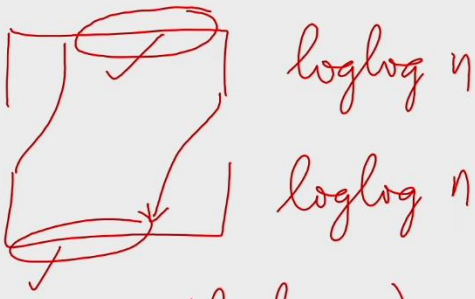
$$k = O(\log\log n)$$

CREW PRAM



$O(\log\log n)$ time

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



$O(\log\log n)$

C

$O(\log\log n)$ time

$n/\log\log n$ processors

On a CREW PRAM