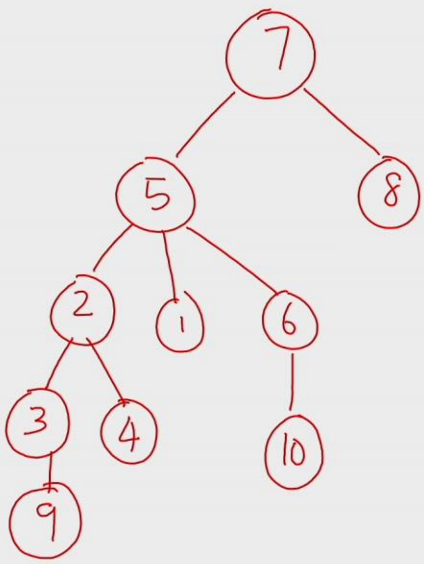
Preorder Traversal

of a rooted tree T is a listing of the vertices of T in which every vertex is listed before its children



Preorder Traversal

7
5
2
3
9
4
1
6
10
8

parent - child edges : 1
count up

child - parent edges do not
count : 0

Euler ckt starting at the root
assign weights of
1 to p-c edges
0 to c-p edges

Prefix sums on the Euler ckt

75	1	1
52	1	2
23	1	3
39	1	4
93	0	4
32	0	4

24	1	5
42	0	5
25	0	5
51	1	6
15	0	6
56	1	7
6 10	1	8
10 6	0	8

65	0	8
57	0	8
78	1	9
87	0	9

7	5	2	3	9	4	1	6	10	8
0	1	2	3	4	5	6	7	8	9

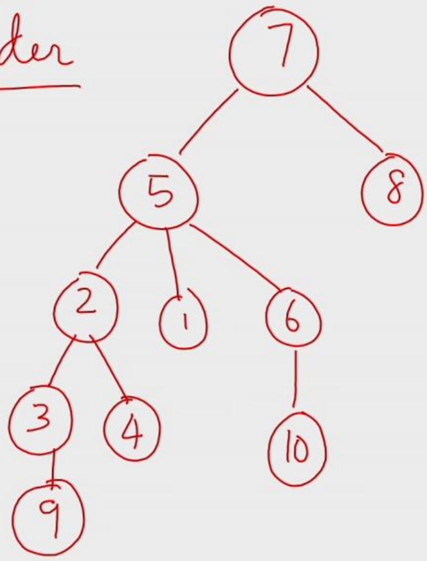
Preorder Traversal
can be computed in
$O(\log n)$ time $\frac{n}{\log n}$ processors
on an EREW PRAM

Post Order Traversal

of T, every vertex v is listed after all its children.

Post order

9	6
3	5
4	8
2	7
1	
10	



Preorder Traversal

7	1
5	6
2	10
3	8
9	
4	

Assign weights to the edge

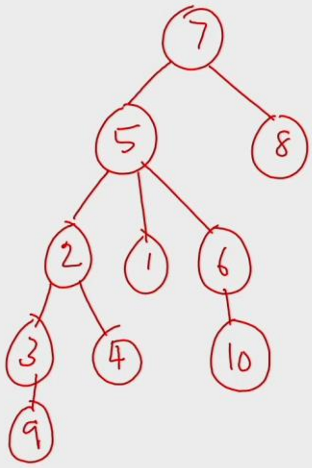
p–c edge : 0

c–p edge : 1

$O(\log n)$ time using $n/\log n$ pr.

EREW PRAM.

Number of descendants of node



a node along with its
children
g. children
g.g. children & so on
form the set of
its descendants

for each parent-child ~~node~~
edge $[i, j]$

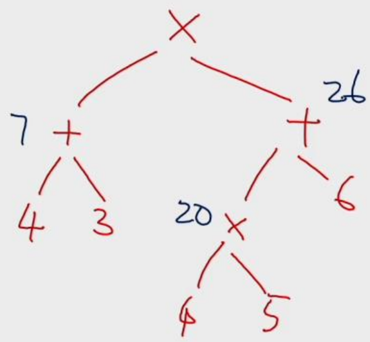
$\text{rank}([j, i]) - \text{rank}([i, j])$
$= \#$ proper descendants of j

$O(\log n)$ time $\quad n/\cancel{n}\log n$ processors
on EREW PRAM

Arithmetic expression

$\times \quad +$ values are integers

$(4+3) \,\underline{\underline{\times}}\, ((\underline{4\times 5)+6})$



<u>expression tree</u>

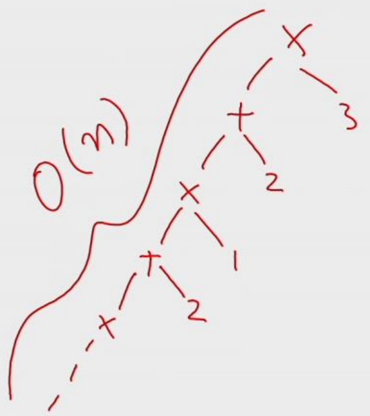
An expression tree can be evaluated bottom up.

Parallel? Balanced? efficient

Not Balanced:

$O(\text{depth})$.

Depth could be $O(n)$
when n is the size of the tree



$\underline{O(n) \text{ time}}$

will not do

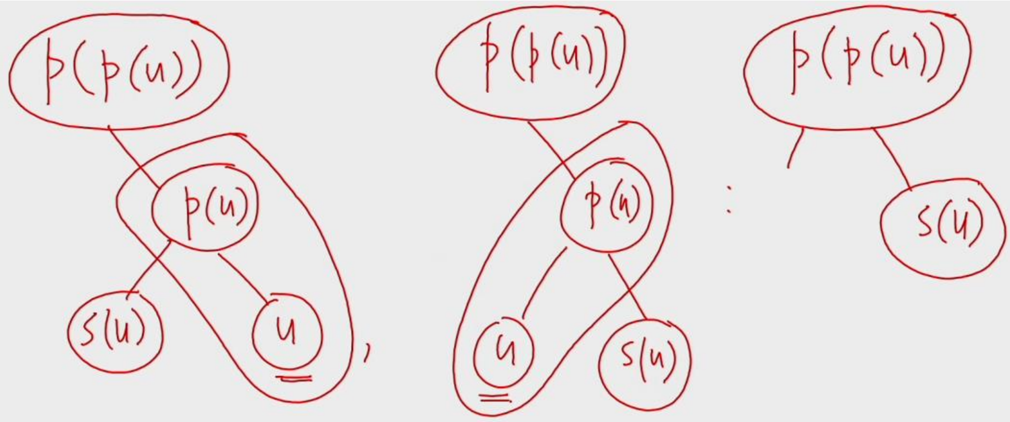
Tree Contraction Technique

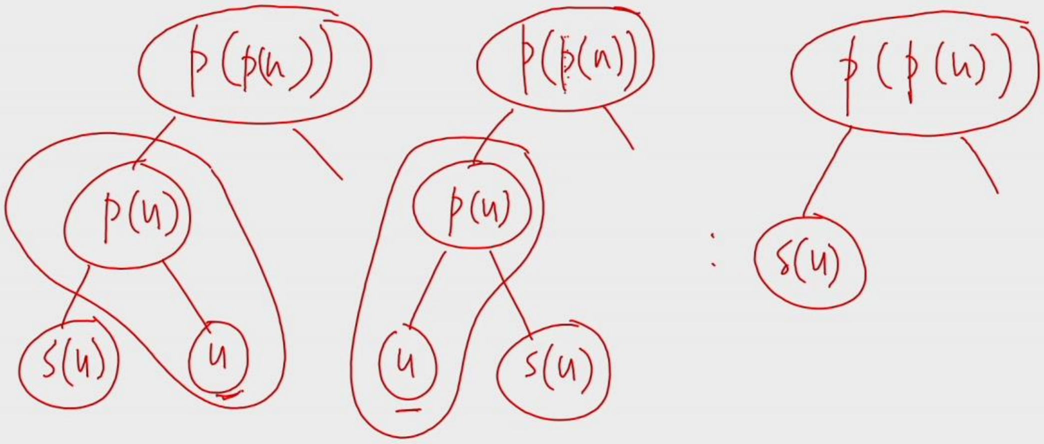
every internal node has 2 children

Rake operation

A leaf u s.t $p(u) \neq$ root

remove u & $p(u)$ and make the sibling (u) a child of $p(p(u))$





shrinking of a tree to a single vertex by repeated rake ops and one final reduction of a tree of the form 

T s.t T is a rooted binary tree

each ~~vertex~~ nonleaf has exactly 2 children.

each vertex has $p(\cdot)$

$s(\cdot)$

1. Get the adjacency list repn of T
2. Find an Euler ckt
3. Break the ckt open by deleting an incoming edge of r
4. Find a preorder traversal

5. Copy the traversal into an array

6. Mark all the leaf nodes

7. Compact the leaf nodes.

Now the leaves are in a L to R order. Let A denote the array of leaves

A_{odd} : the seq. of the odd elements in A

A_{even} : ______________ even

____

a b c d e f

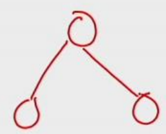
| × | × | ×

for $\lceil \log(n+1) \rceil$ iterations

- apply rake on all A_{odd} elements that are left children
- apply rake on the remaining elements of A_{odd}.
- $A = A_{even}$

we have at most $\lfloor m/2 \rfloor$ leaves, if we began with m leaves before the iteration

leaves is 2



n — 1

$n/2$ — 1

$n/4$ — 1

$\vdots$

$O(n) : O(\log n)$

Brent's SP

$O(\log n)$ time

$n/\log n$ processo: