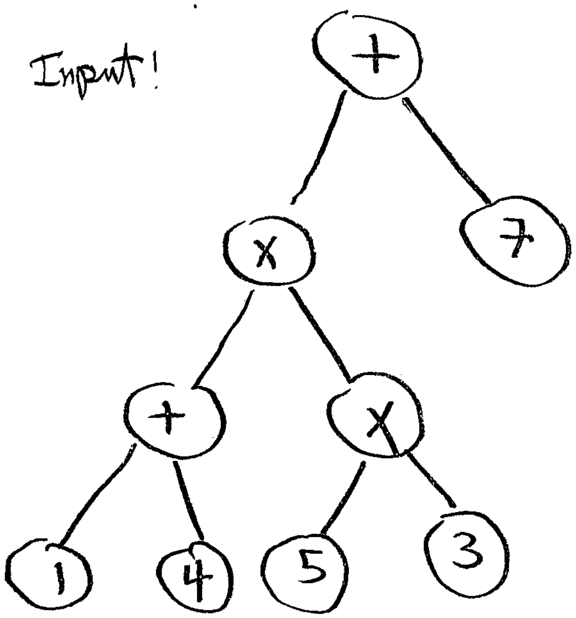
15-750
4/4/16

Parallel Expression Evaluation

Eg Input:



Output: Value, all subvalues

Goal: Parallel Alg

Simple Alg

1) Assign a processor to each node.

While tree non empty **do**

2) **if** Leaf "send" value to parent
delete node [RAKE]

3) if node has 2 values then evalute.

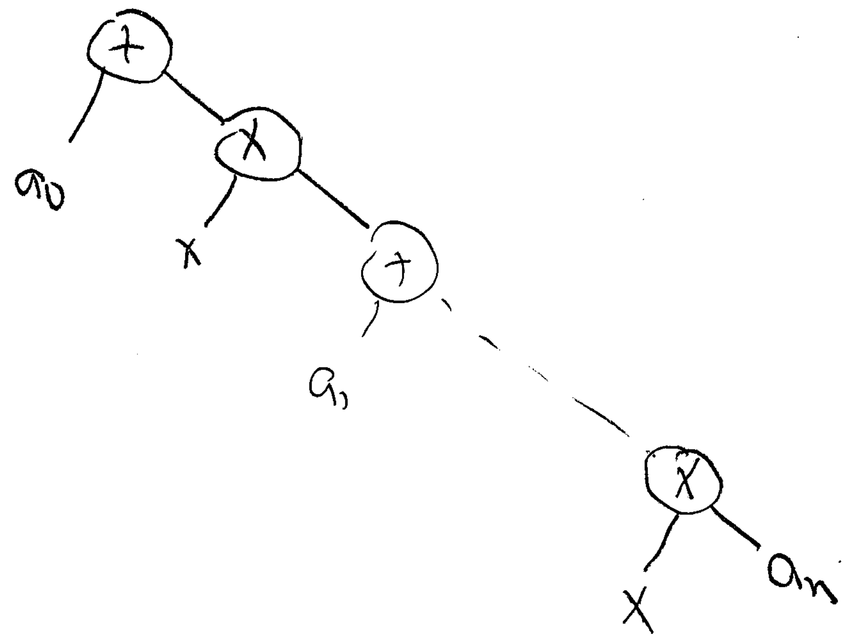
Worst Case for simple Alg

Recall: Horner's Rule

Input: polynomial $a_0 + a_1 X + a_2 X^2 + \cdots + a_n X^n$

Alg: $a_0 + X(a_1 + X(\cdots + X(a_{n-1} + X(a_n)\cdots)$

As a tree



Simple Alg

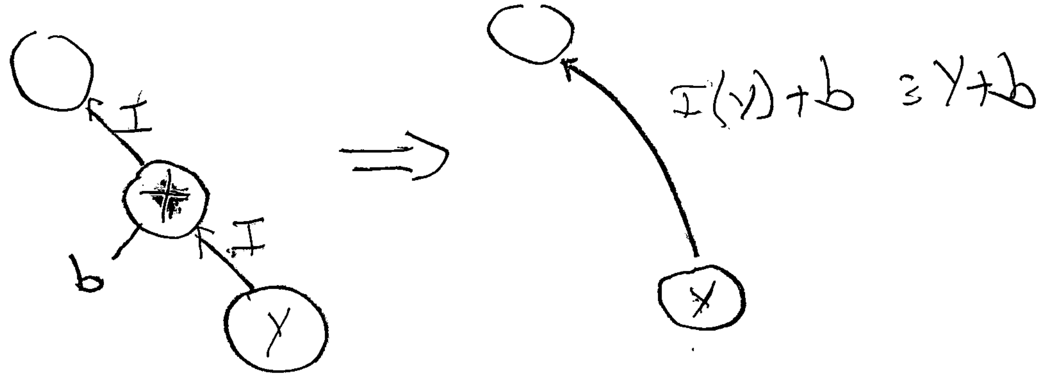
$O(n)$ Time

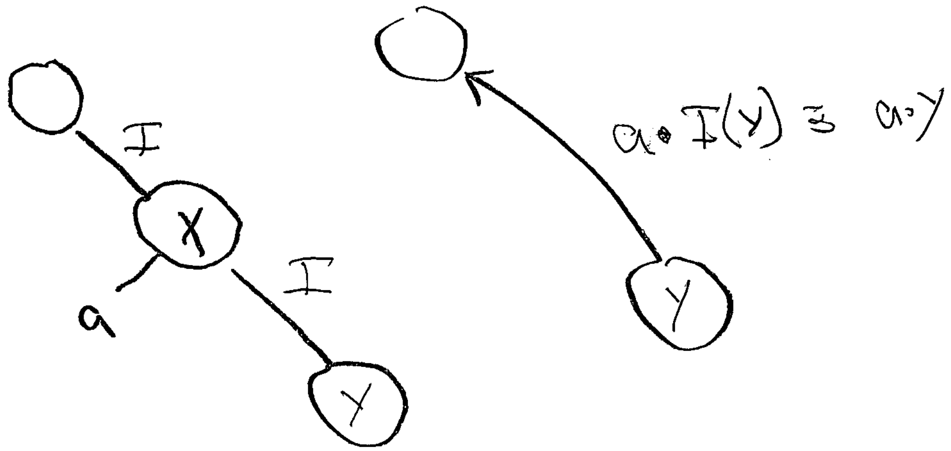
$O(n^2)$ P.T.

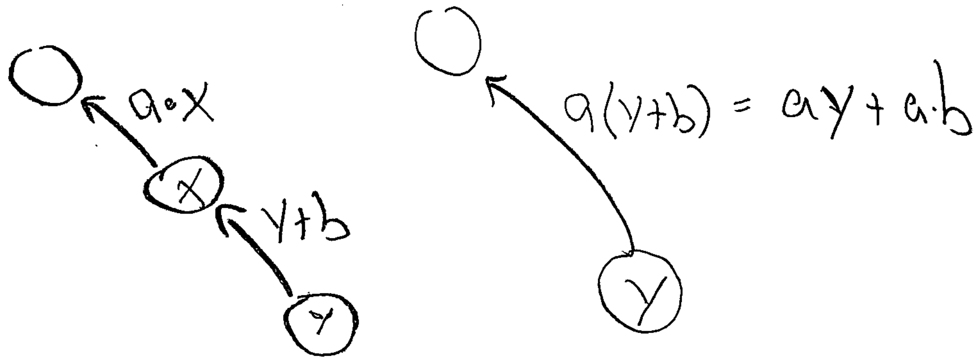
Keeping nodes with only one value busy!

Here we view the tree edges as transformers.

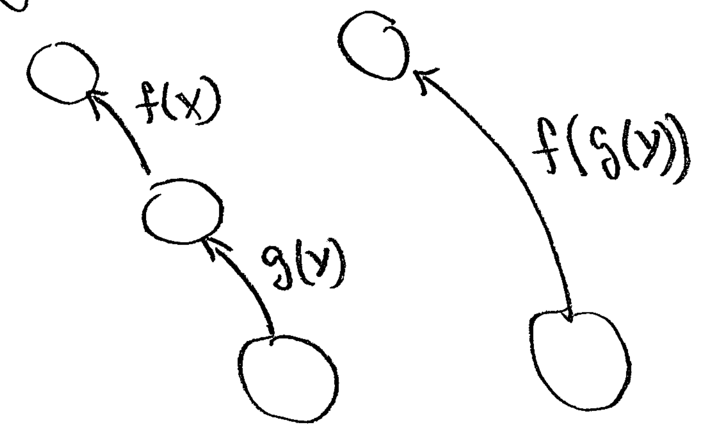
Init: The edge are the identity.







The general case.



$$f(y) = ay + b \qquad g(y) = cy + d$$

$$f(g(y)) = a(cy+d) + b = ac\,y + (ad+b)$$

Note: fcns $ay+b$ are closed under compositions

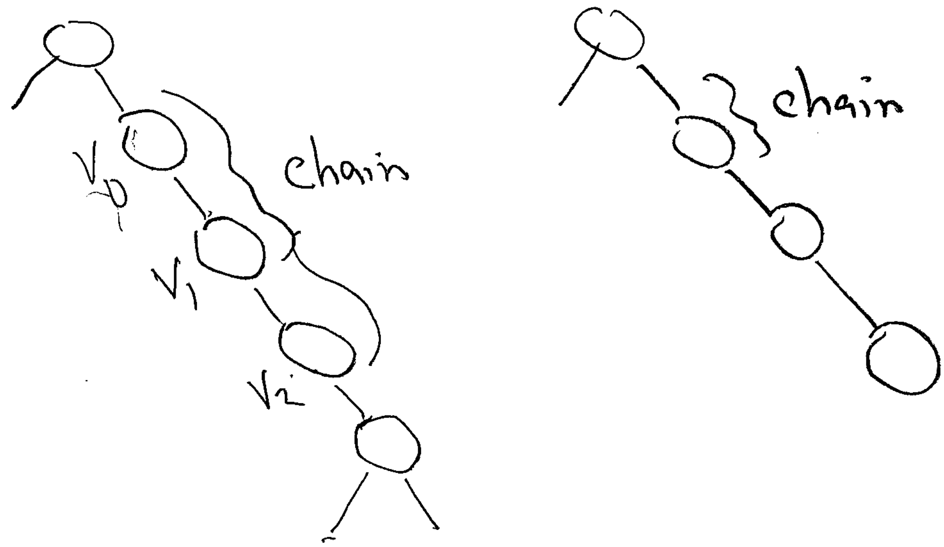
We can also remove an independent set of 1-child nodes (degree 2 nodes)

Very similiar to pivoting in Gaussian Elim!

Def $V_0 \cdots V_K$ is a chain if:

1) V_{i+1} is only child of V_i $0 \le i < K$.
2) V_K has only one child & it is not a leaf

Eg:



The Independent Set

1) All leaves
2) Max independent set from each maximal chain

Parallel Tree Contraction

RAKE $\equiv$ remove all leaves

COMPRESS $\equiv$ replace each maximal chain of length k with one of length $\lfloor k/2 \rfloor$.

CONTRACT $\equiv$ {RAKE, COMPRESS}

Thm $|\text{CONTRACT}(T)| \leq \frac{2}{3}|T|$

pf Def V_0 = leaves of T

$V_1 \subseteq V$ with 1 child

$V_2 \subseteq V$ with $2 \leq$ # children

$C \subseteq V_1$ with child in V_0

<u>Claim</u> $|V_0| > |V_2|$

<u>pf</u> induct on size of T.

Claim: $|V_0| \geq |C|$

<u>Def</u> $R_a = V_0 \cup V_2 \cup C$

$Com = V_1 - R_a$

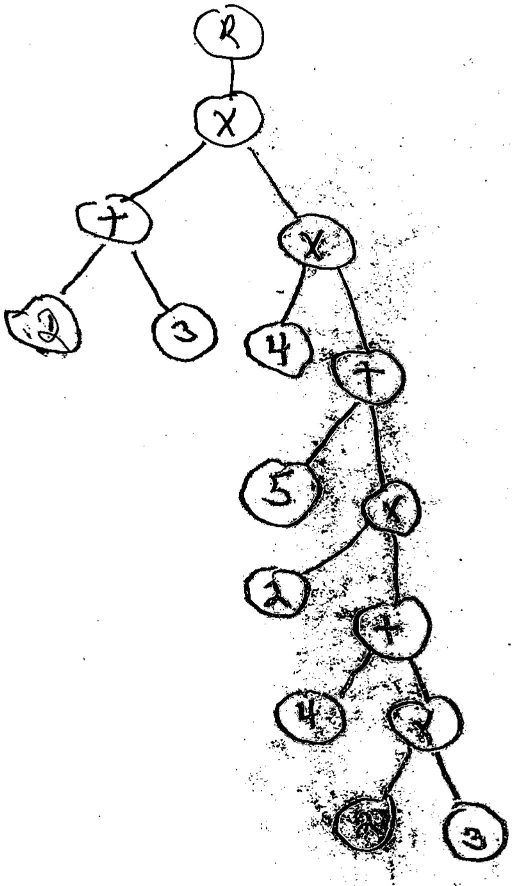
$RAKE(R_a) \subseteq V_2 \cup C \Rightarrow |RAKE(R_a)| \leq 2/3 |R_a|$

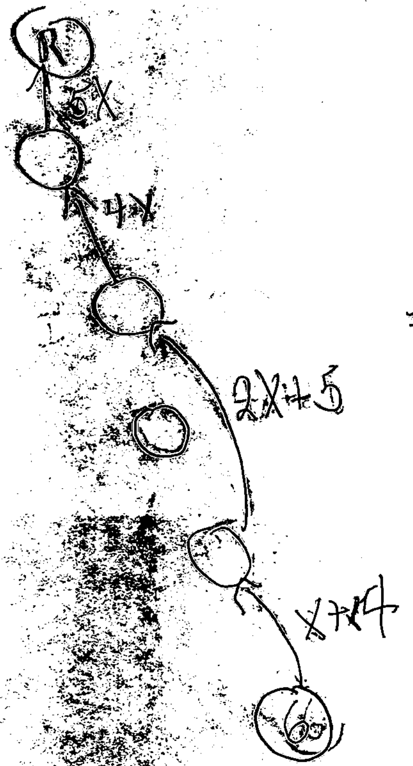
<u>Note</u> Com $\equiv$ union of maximal chains

$|COMPRESS(Com)| \leq \frac{1}{2} |Com|$

<u>Cor</u> After $\log_{3/2} n$ CONTRACTS T is empty

7A





Work and Time Efficient Tree Contraction

Idea 1 Do regular RAKE.
Use Random-Mate to COMPRESS chains.

Using Chernoff Bounds

Thm Randomized Tree Contraction runs in $O(\log n)$ with high prob.

Thus $W = O(n \log n)$ Time $= O(\log n)$.

Idea 2
1) Break tree into $n/\log n$ pieces each of size $\le \log n$
2) Contract pieces to constant size.
3) Run Rand Tree Contraction on tree of size $O(n/\log n)$

A Tree into Bridges

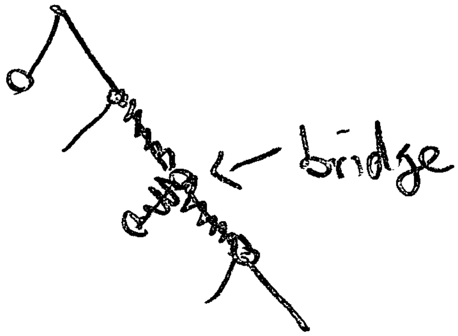
Let T be a rooted tree $T = (V, E)$

<u>Def</u> A subtree B is a **bridge** if at most 2 attachments: a root, a leaf.

<u>eg</u> 1) Single edge

2) Induced subtree

3)



<u>Thm</u> $\forall m\ \exists$ decomp of T into $O(n/m)$ bridges of size at most m.

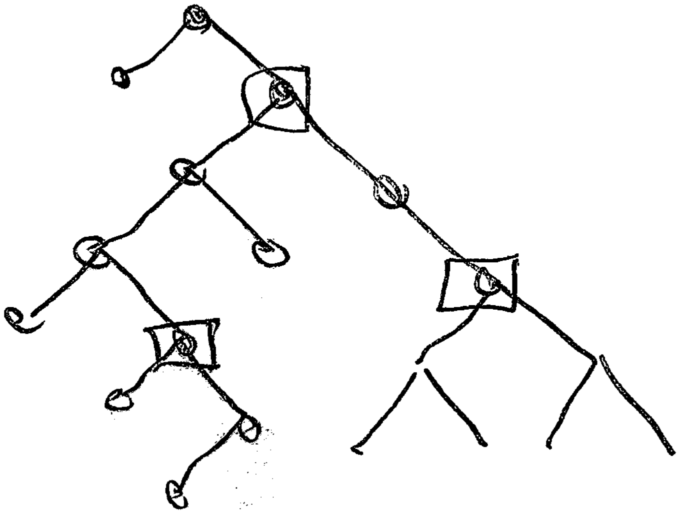
m-critial nodes

$T = (V, E)$ $W(v) \equiv$ # nodes in subtree rooted at v.

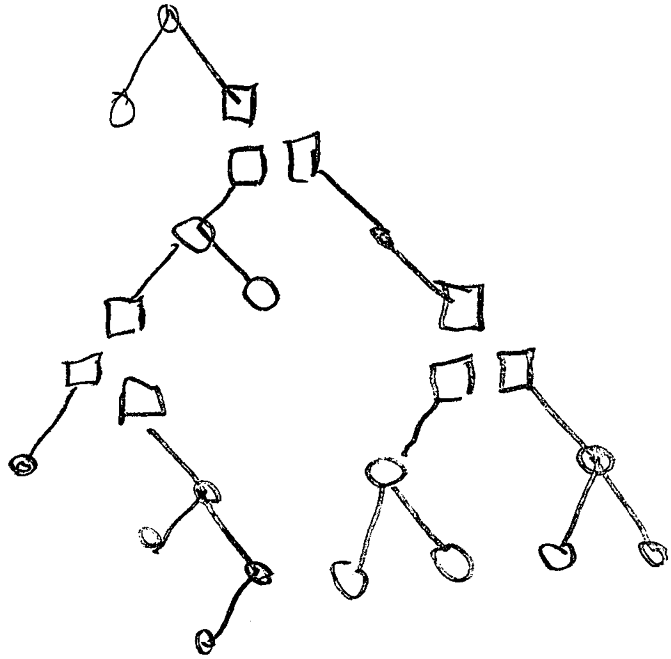
Def v is m-critical if

1) v not a leaf.
2) $\left\lceil \frac{W(v)}{m} \right\rceil > \left\lceil \frac{W(v')}{m} \right\rceil$ $\forall v' \in \text{children}(v)$.

eg 5-critical



5-bridges



Claim (see chap 3) (m-1)-bridges proves thm.

<u>Thm</u> Tree Contraction can be done in
$O(n)$ work(PT) $O(\log n)$ time with high prob.

<u>Known:</u> Det in same bounds.

<u>Alg</u>

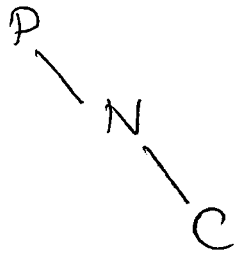
1) Compute $\log n$-critical nodes using Euler tour
2) Contract bridges
3) Contract $n/\log n$ tree using random mate.
4) Expand.

Simple Application of PTC

Rooted binary tree T with node weights

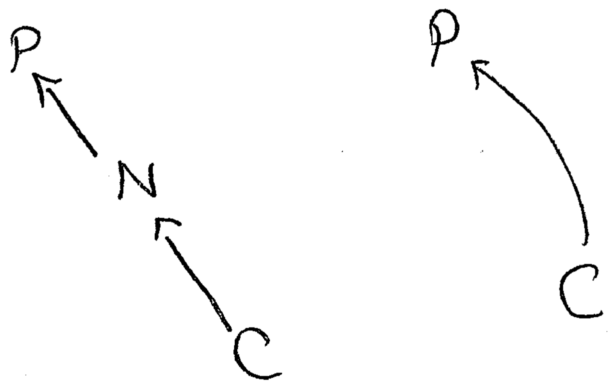
Goal: For each node compute max value in subtree

	Contract	Expand
Rake	$V(P) = \mathrm{Max}\{V(P), V(L)\}$	$\emptyset$
Compress	$V(P) = \mathrm{Max}\{V(P), V(N)\}$	$V(N) = \mathrm{Max}\{V(N), V(C)\}$

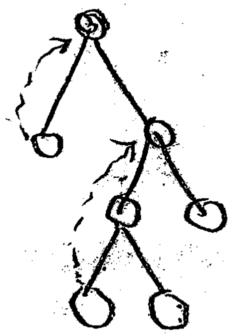


Goal: Max of ancestors

	Contract	Expand
R	$\emptyset$	$V(L) = \text{Max}\{V(L), V(P)\}$
Comp	$V(C) = \text{Max}\{V(C), V(N)\}$	$V(N) = \text{Max}\{V(N), V(P)\}$



Low Point Numbers



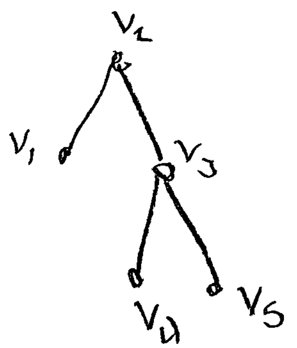
After computing min value of backedges per node, the problem reduces to min value of each subtree

Lowest Common Ancestor Prob LCA

Input: Rooted tree T and nontree edges $e_1 \cdots e_m$

Output: For each edge $e_i = (v, w)$ the LCA of v & w

eg



$LCA(v_1, v_5) = v_2$

$LCA(v_3, v_5) = v_3$

$LCA(v_4, v_4) = v_4$

$LCA(T, e_1, \cdots, e_m)$

Run PTC on T for each round do

In|| for each edge $e = (e_v, e_w)$

In|| for each $(u, u') \in \{(v,w), (w,v)\}$

If u ancestor u' then $LCA(e) = u$

else $e_u = \text{parent}(u)$ & $e_{u'} = \text{parent}(u')$