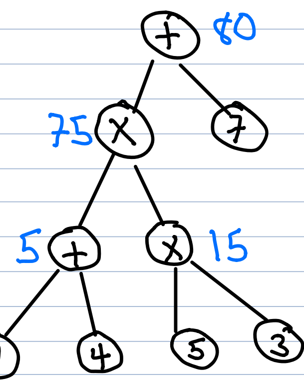
Parallel Expression Evaluation

15-750
5/1/19

Eg. Input



Possible outputs Root value, all subvalues

Simple Alg

1) Assign a processor to each node.
2) While tree not empty do
   a) If Leaf "send" value to parent & delete node [RAKE]
   else if node has values then evaluate.

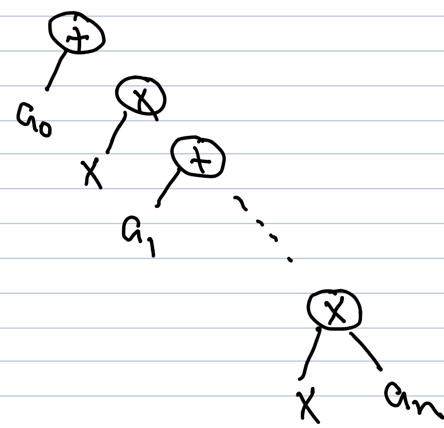
Bad Case for Simple Alg

Recall: Horner's Rule Polynomial eval

Input: polynomial $a_0 + a_1 x + \cdots a_n x^n$

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

As a tree



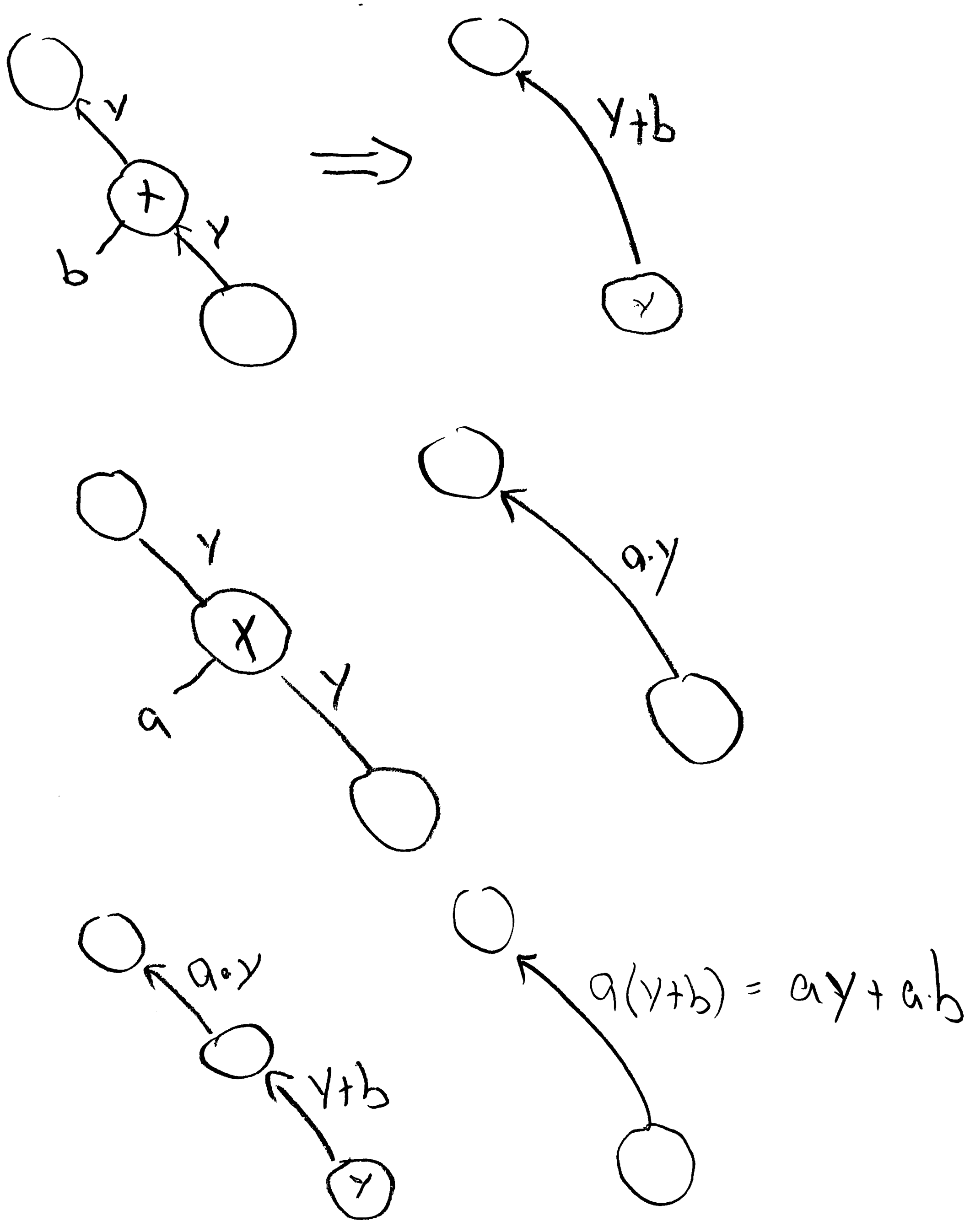
Simple Alg is $O(n)$ time

$O(n^2)$ proc. time

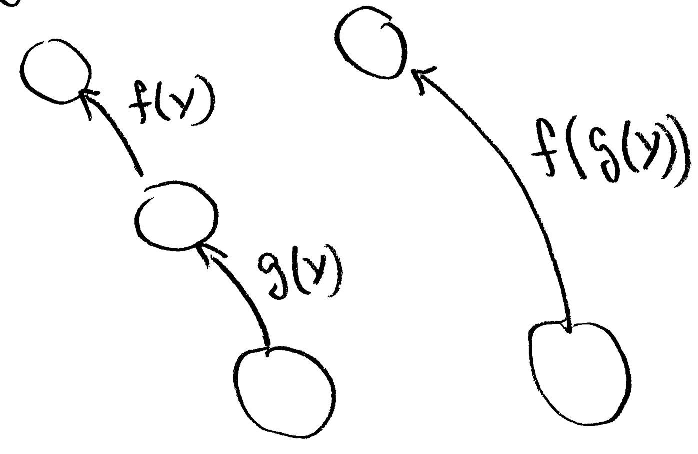
Keeping nodes with only one value busy!

Here we view the tree edges as transformers.

Init: The edge are the identity.



The general case.



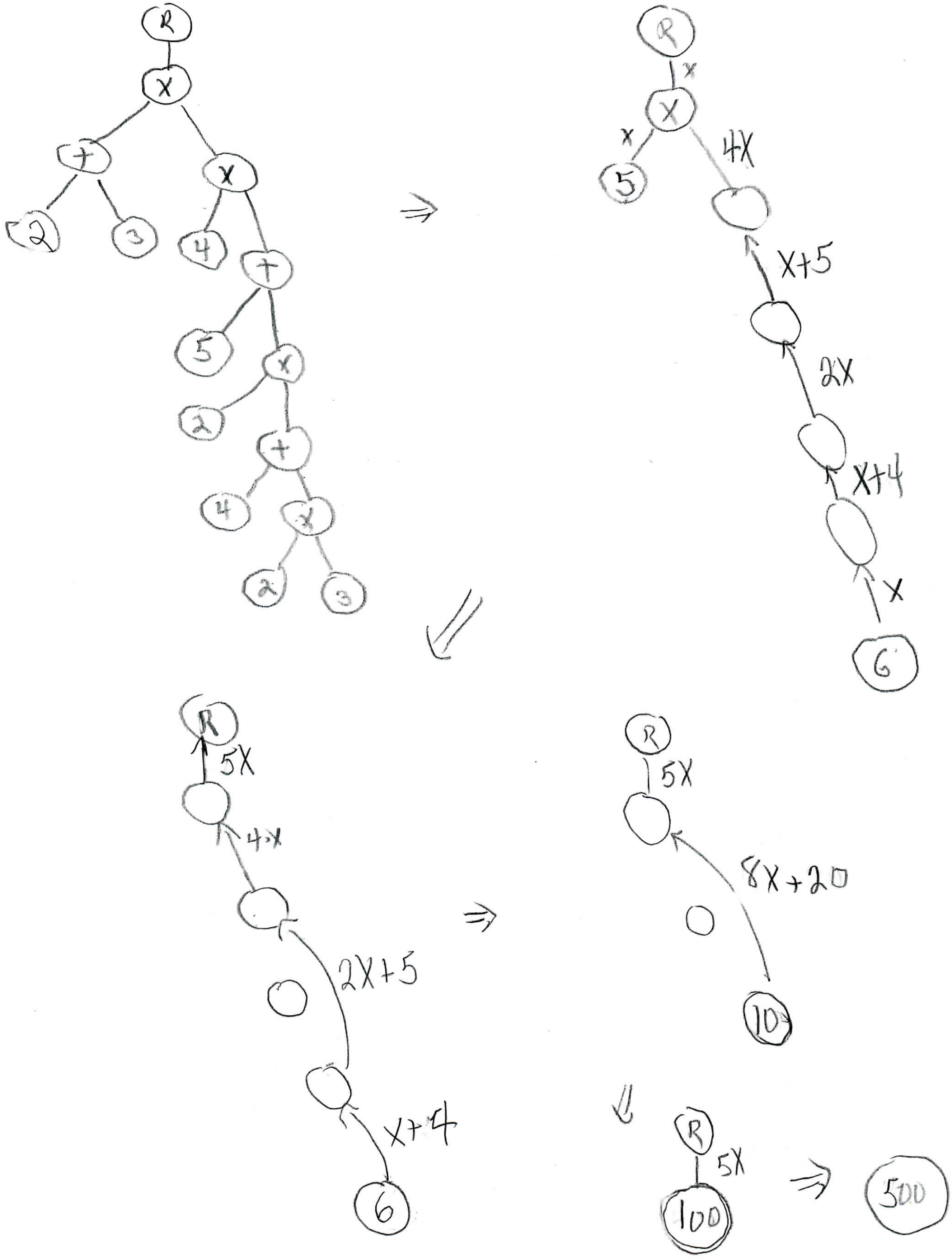
$$f(y) = ay + b \qquad g(y) = cy + d$$

$$f(g(y)) = a(cy + d) + b = \overline{a \cdot c}\, y + (ad + b)$$

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

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

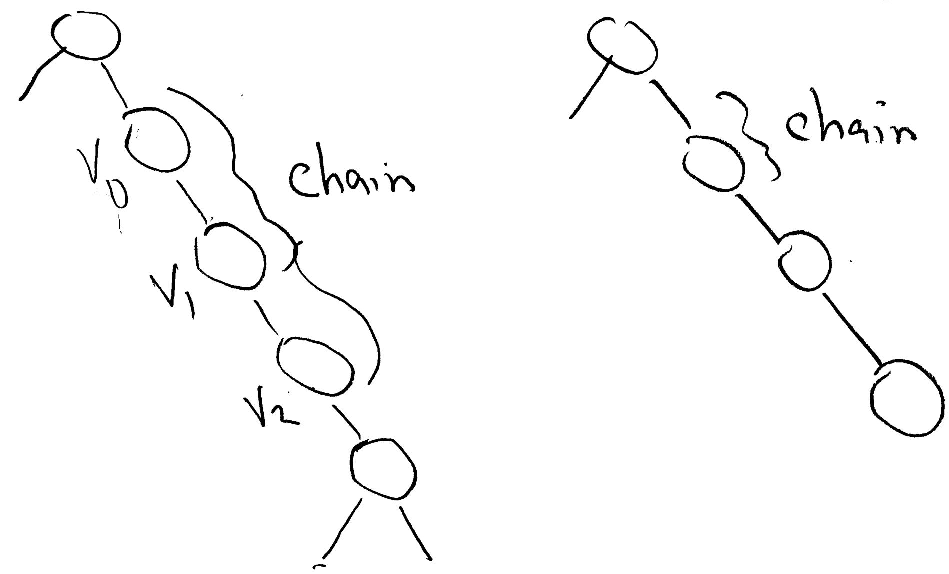
Very similiar to pivoting in Gaussian Elim!


R
X
+
2
3
X
4
+
5
X
2
+
4
X
2
3
R
x
X
x
4X
5
X+5
2X
X+4
X
6
R
5X
4·X
2X+5
X+4
6
R
5X
8X+20
10
R
5X
100
500

<u>Def</u> $V_0 \cdots V_k$ is a <u>chain</u> 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:



<u>The Independent Set</u>

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 $= \{$RAKE, COMPRESS$\}$

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

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> $R_a = V_0 \cup V_2 \cup C$

$Com = V_1 - R_a$

$RAKE(R_a) \subseteq V_2 \cup C \Rightarrow |RAKE(R_a)| \leq \frac{2}{3}|R_a|$

<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

Max Value in Subtree

Input: Rooted binary tree T with node weights.

Output: Max value in each subtree.

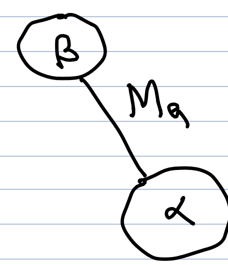
Unary fcn: $M_a(x) = \max\{a, x\}$

Alg

For each edge set edge a tranformer to $M_{-\infty}(x)$

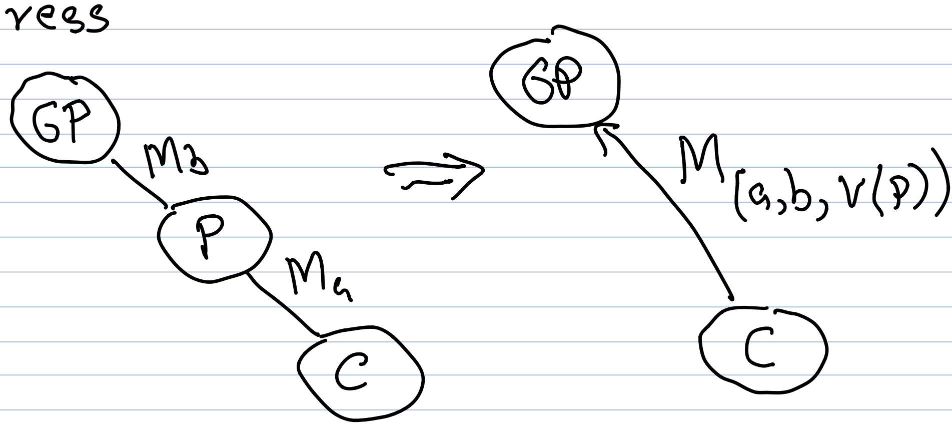
Contraction Phase:

Rake: $V(P) = M_a(V(\text{Leaf}))$

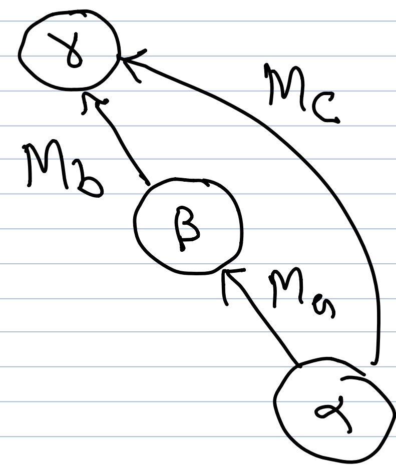


$\beta \Leftarrow M_a(\alpha)$

Compress



Expansion Phase



$\Rightarrow$ Set $\beta = M_a(\alpha)$

Prob: Find Max of Ancestors

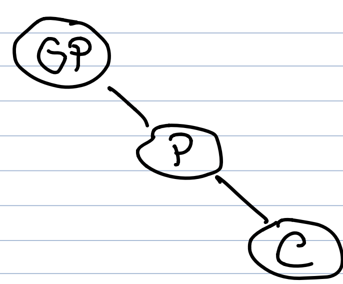
Contraction Rake $\equiv \emptyset$

Compress $V(C) = \mathrm{Max}\{V(P), V(C)\}$

Expansion

Rake $V(C) = \mathrm{Max}\{V(P), V(C)\}$

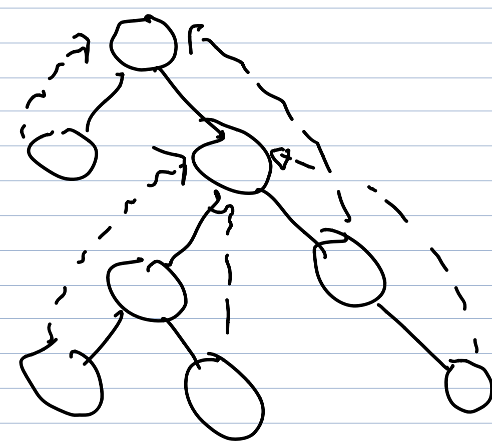
Compress $V(P) = \mathrm{Max}\{V(GP), V(P)\}$



Computing Low Point Numbers

Input: DFS tree of undirected graph.

Output: Low Point Numbers

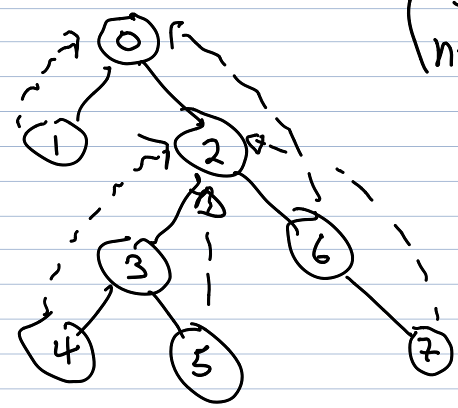


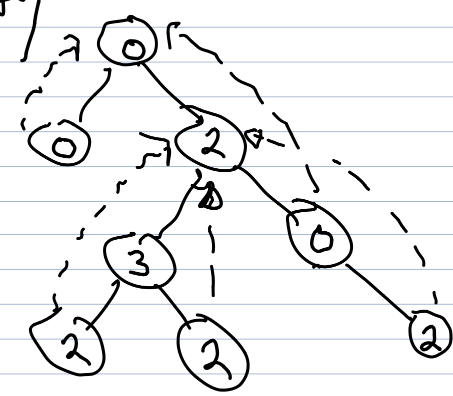
Alg: T

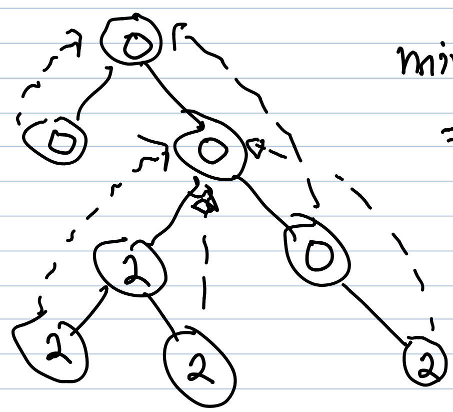
1) Compute preorder numbers
2) Replace preorder numbers with beckedge numbers
3) Compute minvalue in each subtree

Inorder

(backedge numbers)







minvalue in subtree

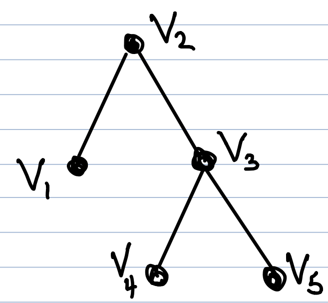
= low-point.

Lowest Common Ancestor Prob LCA

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

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

eg



$LCA(V_1, V_5) = V_2$

$LCA(V_3, V_3) = V_3$

$LCA(V_3, V_5) = V_3$

Note After an Euler Tour we get a constant work ancestor test.

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

For each round of PTC do the following:

1) (RAKE) In $\|$ $\forall$ leaf nodes v

a) If e is selfloop at v then $LCA(e) = v$ & delete

b) else move edges to parent & delete leaf.

2) (Compress) In $\|$ for each node v to be spliced out

i) $\forall$ edge $e = (v, w)$

If v ancestor of w then

$LCA(e) = v$ & delete e.

else $e = (parent(v), w)$

Splice Out v.

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 $T = (V, E)$

Def A subtree B is a bridge if at most 2 attachments: a root, a leaf.

eg 1) Single edge

2) Induced subtree

3)

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

m-critical nodes

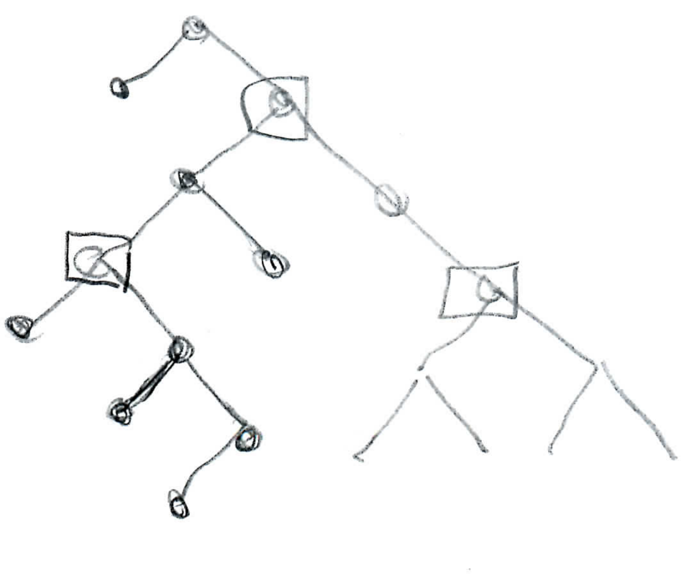
$T=(V,E)$ $\quad w(v) = $ # nodes in subtree rooted at v.

Def v is m-critical if

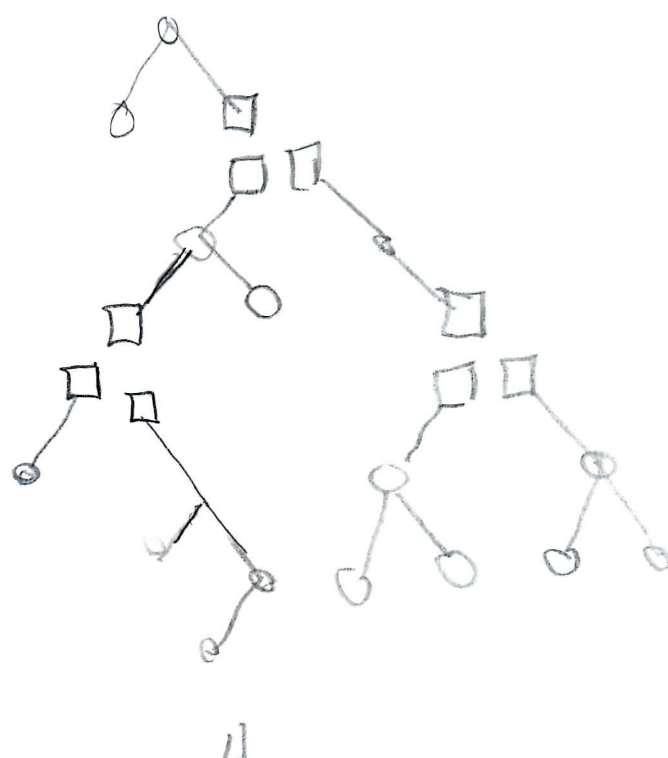
1) v not a leaf.
2) $\lceil \frac{w(v)}{m} \rceil > \lceil \frac{w(v')}{m} \rceil \quad \forall v' \in \text{children}(v)$.

---

eg 5-critical



5-bridges



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

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

Known: Det in same bounds.

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