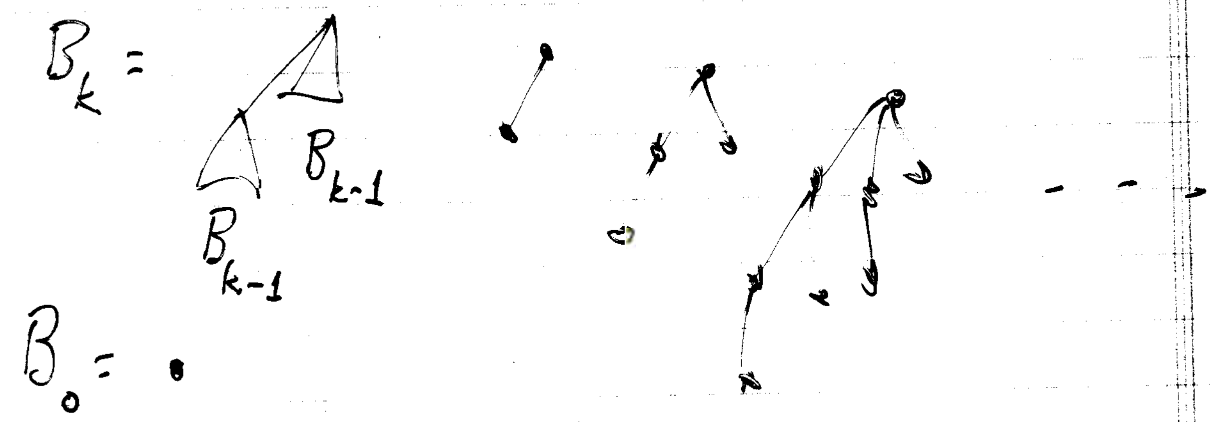
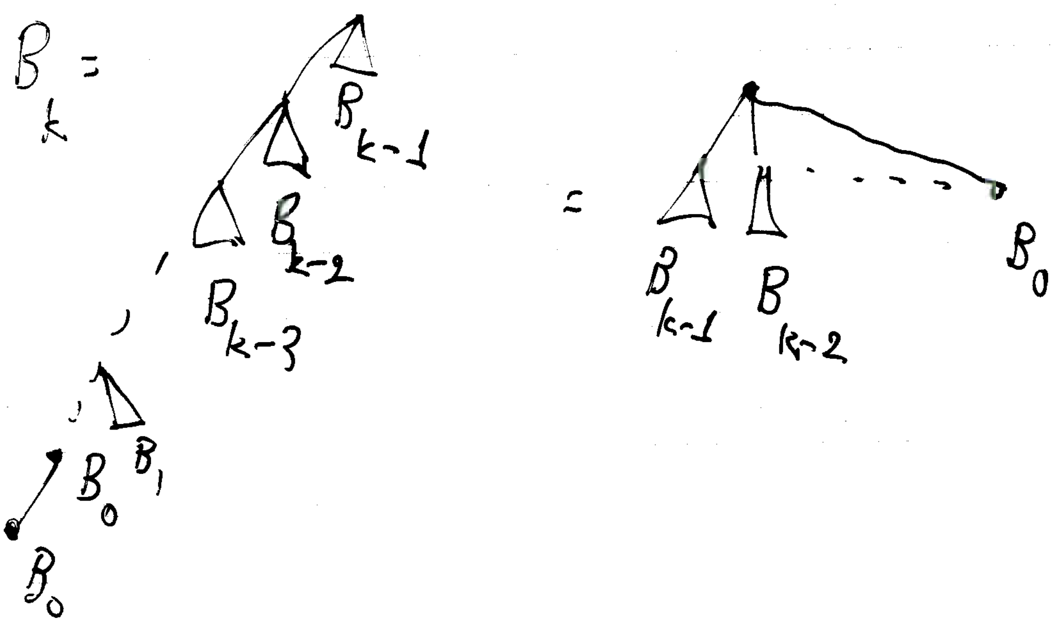
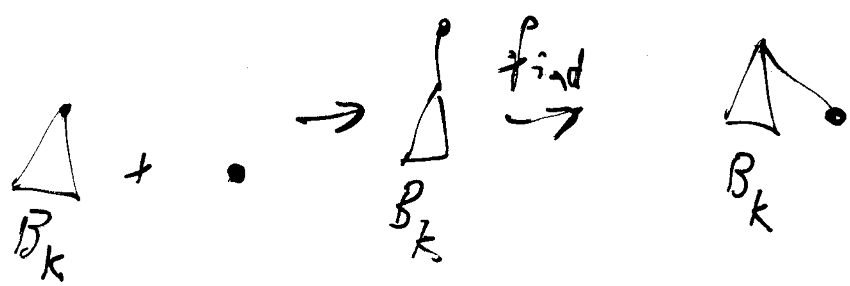
→ $O(\log n)$ per find/union

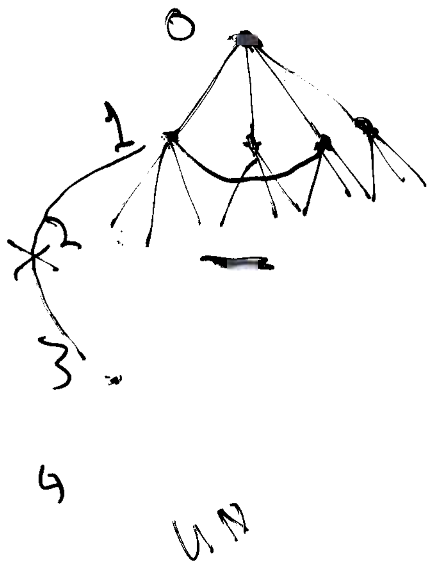
$O(\alpha(n))$ " " "

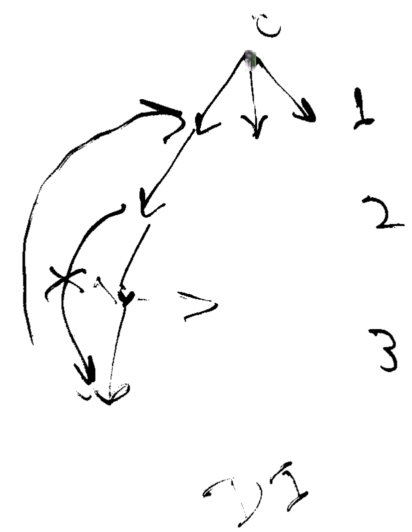


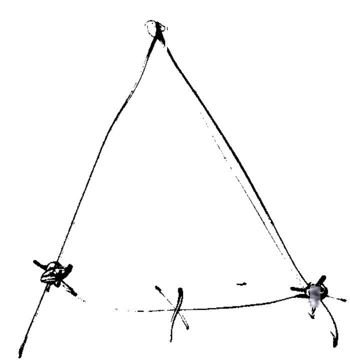




BFS







Properties of Depth-First Search

First edge into each vertex = tree edge

Tree edges define spanning forest; trees rooted at search start vertices

Previsit order = preorder on tree (discovery order)

Postvisit order = postorder on tree (finishing order)

v an ancestor of w iff $pre(v) \le pre(w)$
& $post(v) \ge post(w)$

Undirected graph: direct edges along search direction:

each nontree edge leads from a descendant to an ancestor

Digraph: (v, w) an edge implies v and w are related in depth-first spanning forest
or $pre(v) > pre(w)$ (& $post(v) > post(w)$)

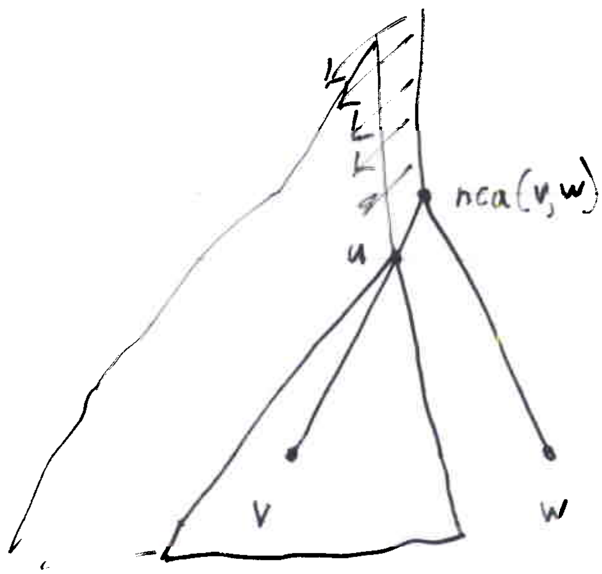
would be traversed from v
before w is reached,
hence a tree edge

(same argument)

Path lemma:

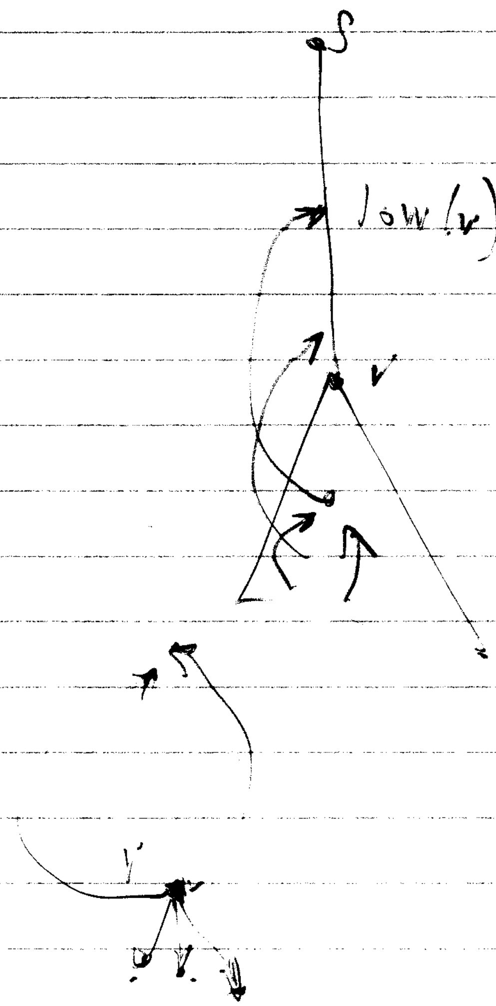
undigraph: Any path from v to w contains a common ancestor of v and w.

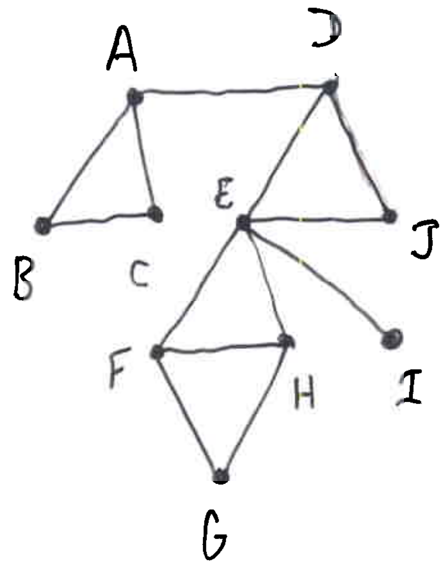
digraph: Any path from v to w with pre(v) < pre(w) (or post(v) < post(w)) contains a common ancestor of v and w.



common ancestor = first vertex on the path that is not a descendant of u

(digraph) = first vertex on the path that is ≥ u in postorder


s
low(v)
v
v'


A
D
E
J
B
C
F
H
I
G

Bipolar order: edges directed so acyclic,
s = only source
t = only sink

Bipolar order = topological order with one source, one sink

G has a bipolar order iff $G \cup \{(s,t)\}$ has no cut vertices

Bipolar order algorithm:

1. DFS from s, starting along (s,t), compute pre, low, parents in DFS tree

2. Visit vertices in preorder, constructing list.

+ = after current vertex $\qquad \bar{s}, t$
− = before current vertex

if sign(low(v)) = +, insert v after $p(v)$; sign($p(v)$) = −
else insert v before $p(v)$; sign($p(v)$) = +