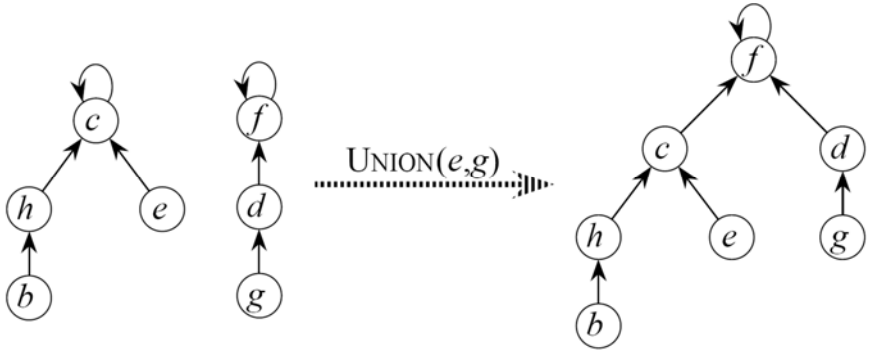
Outline

- Disjoint-Set Operations
- Linked-List Representation of Disjoint Sets
- ***Disjoint-Set Forests***

Forest of Trees (1/2)

- 1 tree per set.
- Root is representative.
- Each node points only to its parent.



Forest of Trees (2/2)

- MAKE-SET: make a single-node tree.
- UNION: make one root a child of the other.
- FIND-SET: follow pointers to the root.

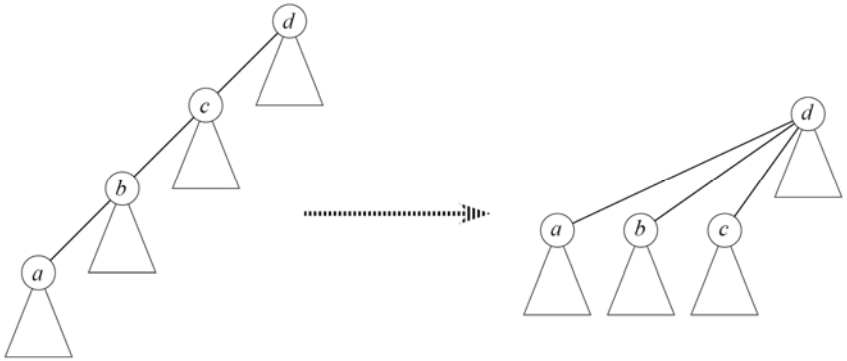
➡ Not so good—could get a linear chain of nodes.

Great Heuristics

- Union by rank: make the root of the smaller tree (fewer nodes) a child of the root of the larger tree.
  - Don't actually use size.
  - Use rank, which is an upper bound on height of node.
  - Make the root with the smaller rank into a child of the root with the larger rank.
- Path compression: Find path = nodes visited during FIND-SET on the trip to the root. Make all nodes on the find path direct children of root.

Path Compression (1/2)

- Each node has two attributes, *p* (parent) and *rank*.



Path Compression (2/2)

MAKE-SET(x)
$x.p = x$
$x.rank = 0$

FIND-SET(x)
if $x \neq x.p$
  $x.p$ = FIND-SET($x.p$)
return $x.p$

LINK(x, y)
if $x.rank > y.rank$
  $y.p = x$
else $x.p = y$
  // If equal ranks, choose y as parent and increment its rank.
  if $x.rank == y.rank$
    $y.rank = y.rank + 1$

FIND-SET makes a pass up to find the root, and a pass down as recursion unwinds to update each node on find path to point directly to root.

UNION(x, y)
LINK(FIND-SET(x), FIND-SET(y))

Homework Assignment #4

Problem 21-2: Depth Determination

- Please implement MAKE-TREE(v), FIND-DEPTH(v), and GRAFT(r, v).
- TAs will announce the detailed Input/Output format in Moodle.
- Please submit your program to e-Tutor.
- Please submit your README document to Moodle.
- **Due Date: 10 May 2017**.