

# COS 528

## Nearest Common Ancestors

© Robert E. Tarjan 2013

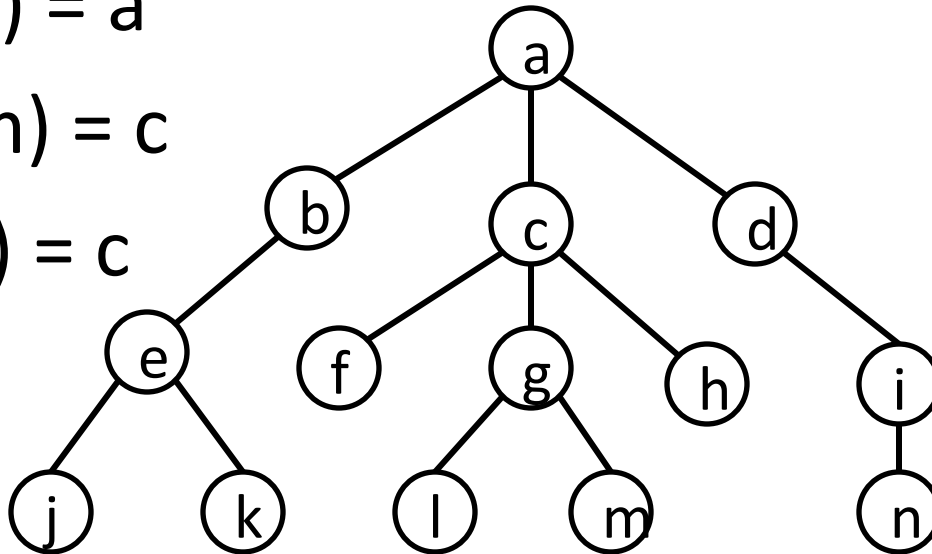
# Nearest common ancestors

Given a rooted tree  $T$  and a set  $Q$  of pairs of vertices  $(x, y)$ , find the *nearest common ancestor*  $nca(x, y)$  of each pair.

$$nca(e, h) = a$$

$$nca(f, m) = c$$

$$nca(c, l) = c$$



# Nearest common ancestors (off-line)

## Depth-first traversal using named sets

Do a depth-first traversal of the tree  $T$ . For each vertex  $x$  visited in preorder, maintain a set named  $x$ , containing  $x$  and all descendants of  $x$  so far visited in postorder. If  $(x, y)$  is a query pair with  $x$  visited second in preorder,  $nca(x, y)$  is the name of the set containing  $y$  when  $x$  is visited in preorder.

# Implementation

$C(x)$  = children of  $x$ ,  $Q(x)$  = query pairs  $(x, y)$ ,

$t$  = root of  $T$

*traverse*( $t$ ) **where** *traverse*( $x$ ) =

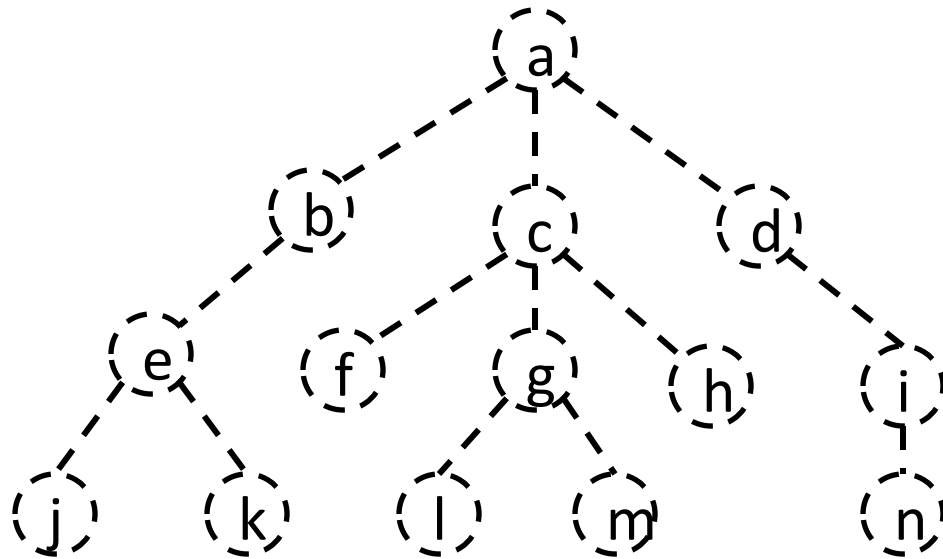
*{make-set*( $x, x$ );

**for**  $(x, y) \in Q(x)$  **do**

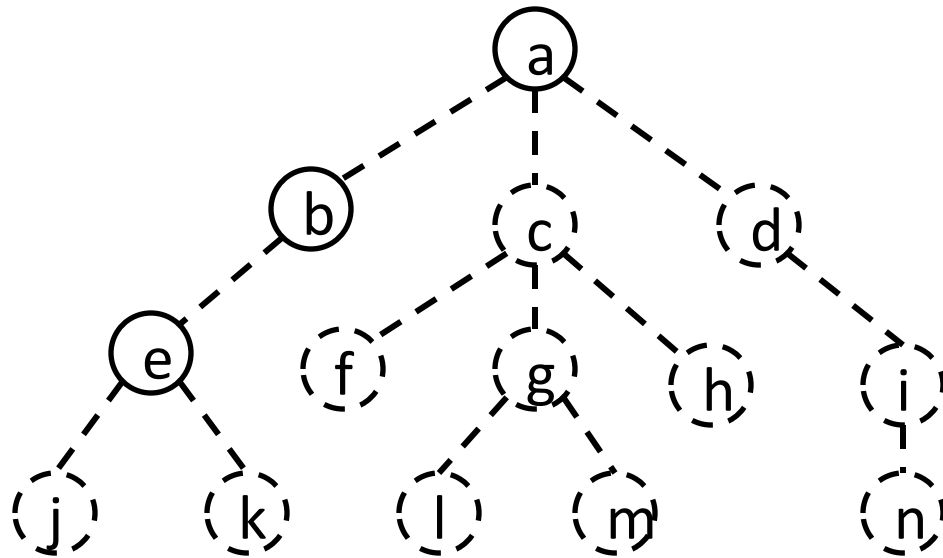
**if**  $y$  in a set **then**  $nca(x, y) \leftarrow find-name(y)$

**for**  $y \in C(x)$  **do** *{traverse*( $y$ ); *unite*( $y, x, x$ )*}*

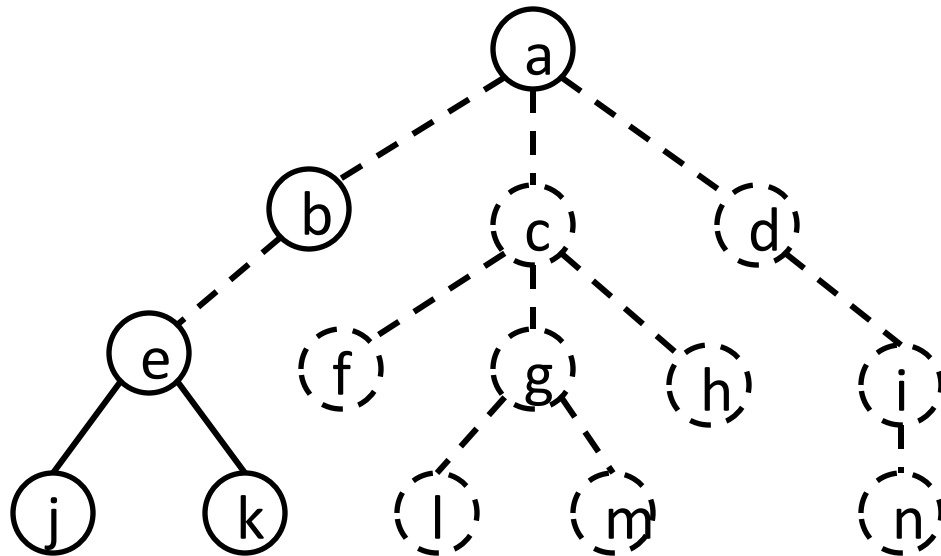
$Q = \{(e, h), (f, m), (c, l)\}$



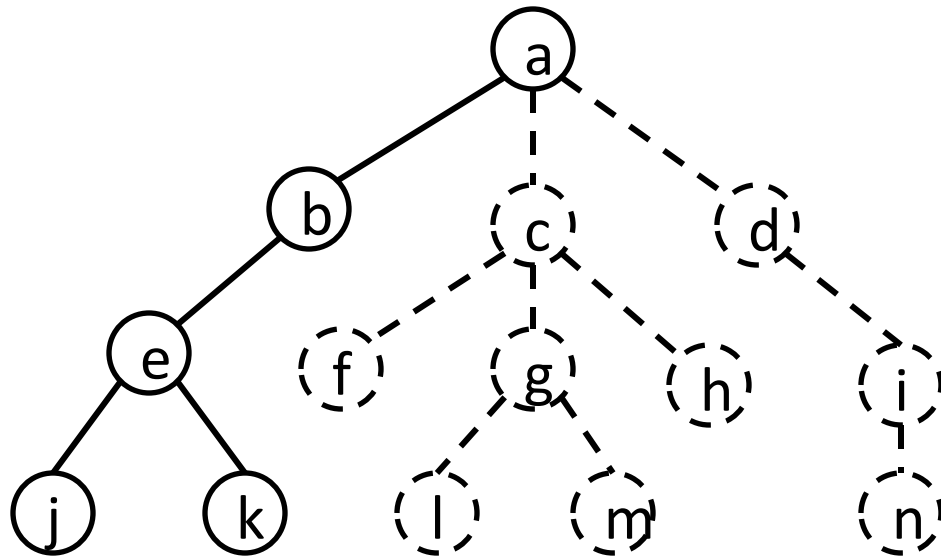
$$Q = \{(e, h), (f, m), (c, l)\}$$



$$Q = \{(e, h), (f, m), (c, l)\}$$

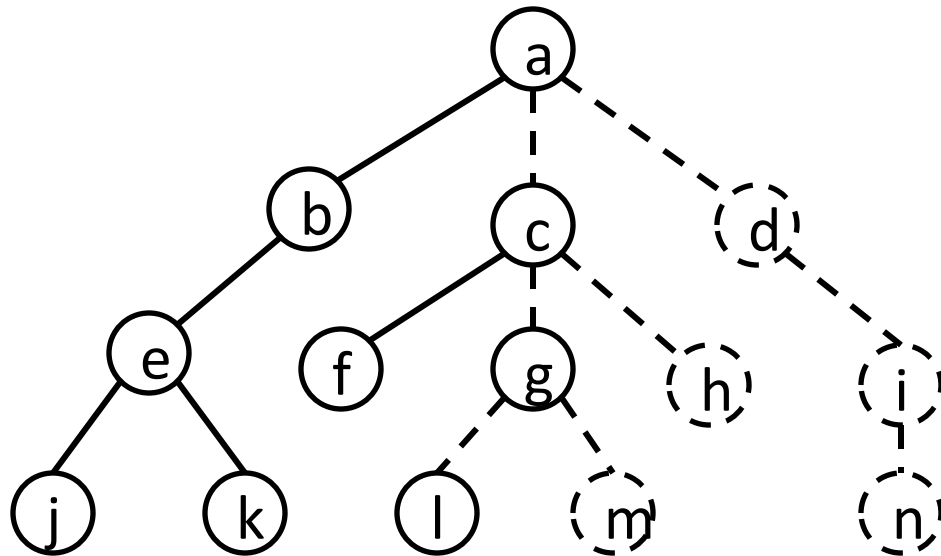


$$Q = \{(e, h), (f, m), (c, l)\}$$

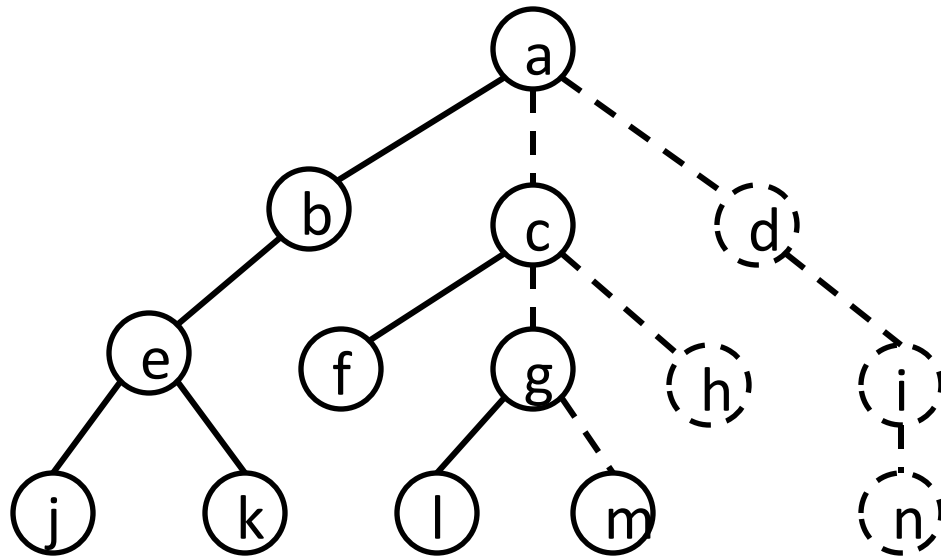




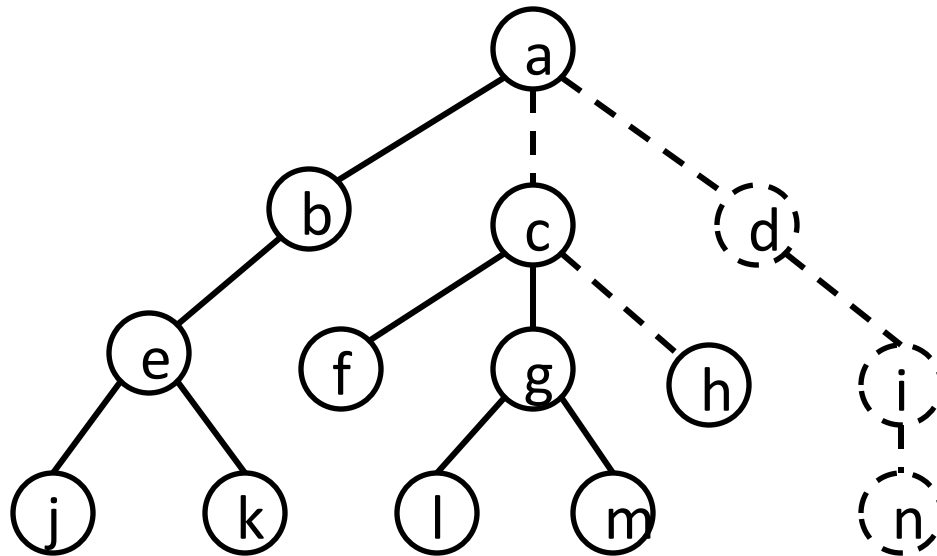
$Q = \{(e, h), (f, m), (c, l)\}$   
 $nca(c, l) = \text{find-name}(c) = c$



$Q = \{(e, h), (f, m), (c, l)\}$   
 $nca(f, m) = \text{find-name}(f) = c$



$Q = \{(e, h), (f, m), (c, l)\}$   
 $nca(e, h) = \text{find-name}(e) = a$



# Correctness of *nca* algorithm

Let  $(x, y)$  be a query pair,  $z = nca(x, y)$ . Suppose  $x$  is visited in preorder after  $y$ . All ancestors of  $y$  that are proper descendants of  $z$  have been visited in postorder by the time  $x$  is visited in preorder, so they are all in the same set as  $z$ . In particular,  $x$  is in the same set as  $z$ . When  $x$  is visited in preorder,  $z$  has not yet been visited in postorder, so  $find-name(y) = z$ .

# Harder Variants

Tree given off-line but queries given on-line

Tree and queries given on-line

- leaf addition

- root-root links

- root-node links

- links and cuts