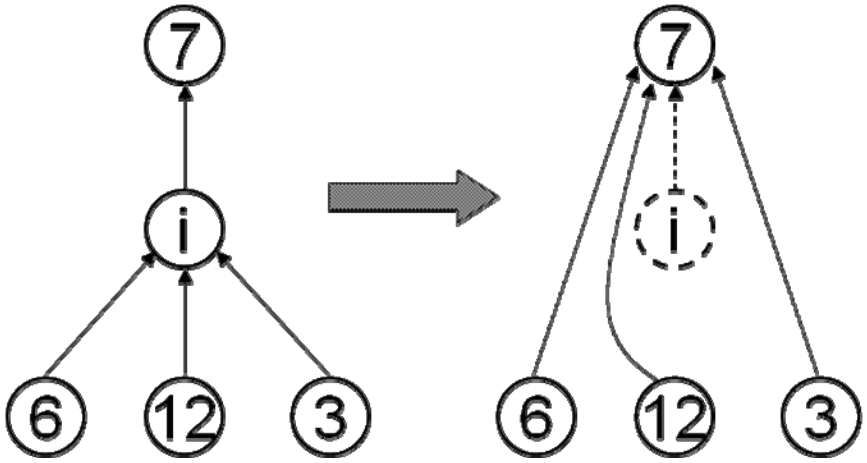
CS 410/586: Quiz 4, 20 April 2009 Name:_____KEY_____

No books or notes. Work individually.

Prof. Mumble claims he has a method to support DELETE's in the tree-structured UNION-FIND algorithm in a way that allows a sequence of n operations (UNION, FIND and DELETE) in $O(n\,G(n))$ time. His method is described by the following pseudo-code and figure.

DELETE(i)
for each child c of the node for i
    Parent(c) ← Parent(i)
remove the node for i.



You point out two problems with Prof. Mumble's method.

Question 1 (4 points): DELETE(i) won't work correctly for all items. Why?
If node i happens to be the root, then the children of i won't be linked correctly.

Question 2 (6 points): A sequence of n operations might actually take $O(n^2)$ time. Why?
Observe that in the tree structure for UNION-FIND, each node knows its parent, but not its children. So to find all the children of node i, we actually have to iterate through every node, checking its parent. So DELETE is actually O(n) if there are n elements in the universe. (There are also arguments that can be made relative to a certain shape of tree, and that one node can have O(n) children that have to be updated on a DELETE.)

Note that if you wanted to include DELETE and keep the same time complexity, you could leave the node in place, but just have a 1-bit flag to mark the node as "dead".