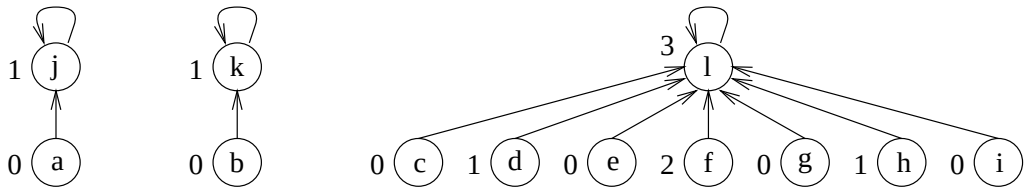
Analysis of Algorithms: Assignment 7

Due date: March 8 (Thursday)

Problem 1

Consider the disjoint-set forest below, where numbers are the ranks of elements, and suppose that you apply three successive operations to this forest: UNION(a, b), UNION(b, c), and FIND-SET(a). Give a picture of the disjoint forest after each of these operations; thus, you need to draw three different pictures.



Problem 2

Write pseudocode for MAKE-SET, FIND-SET, and UNION, using the linked-list representation of disjoint sets. Your UNION algorithm must always append the shorter list to the longer one. For every linked list, you will need to store its size, a pointer to the first element, and a pointer to the last element.

Problem 3 (bonus)

This problem is optional; if you solve it, then you will get one bonus point toward your **final grade** *for the course. You cannot submit this bonus problem after the deadline.*

Suppose that $A[1..n]$ and $B[1..m]$ are sorted arrays, and the size of A is no greater than the size of B, that is, $n \leq m$. Write an algorithm that finds the smallest common element of these arrays; for example, if $A = \langle 1, 4, 5, 7 \rangle$ and $B = \langle 2, 3, 4, 7, 8 \rangle$, then the smallest common element is 4. If the arrays have no common elements, the algorithm should return 0.

Your solution should be efficient both when A is almost as large as B and when A is much smaller than B. In particular, if A is much smaller, the complexity should be better than $\Theta(m)$.