CS3114 Spring 2011 Homework Assignment 2
Due Thursday, March 24 at 11pm
40 points

1. Using the weighted union rule and path compression, show the array for the parent pointer implementation that results from the following series of equivalences on a set of objects indexed by the values 0 through 15. Initially, each element in the set should be in a separate equivalence class. When two trees to be merged are the same size, make the root with greater index value be the child of the root with lesser index value.

(0, 2) (1, 2) (3, 4) (3, 1) (3, 5) (9, 11) (12, 14) (3, 9) (4, 14) (6, 7) (8, 10) (8, 7) (7, 0) (10, 15) (10, 13)

2. A full K-ary tree is a tree where every internal node has exactly K children. Use mathematical induction to prove that the number of leaves in a non-empty full K-ary tree is $(K-1)n+1$, where n is the number of internal nodes.

3. Write an Insertion Sort algorithm for integer key values. However, here's the catch: The input is a stack (*not* an array), and the only variables that your algorithm may use are a fixed number of integers and a fixed number of stacks. The algorithm should return a stack containing the records in sorted order (with the least value being at the top of the stack). Your algorithm should be $\Theta(n^2)$ in the worst case.

4. One can optimize Quicksort to eliminate the cost of function calls by replacing the recursion with a stack to track the sub-arrays that should be processed "recursively."

(a) How many sub-arrays can be put on the stack in the worst case?

(b) Quicksort makes two recursive calls. The algorithm could be changed to make these two calls in a specific order. In what order should the two calls be made so as to minimize the size needed for the stack, and how does this affect how large the stack can become?