Spring 2013 **CMSC 351: Homework 4** Clyde Kruskal

Due Thursday, February 28. Write your NAME and SECTION on the top of your solution.

Problem 1. Consider an array of size nine with the numbers in the following order

$$40, 10, 20, 30, 90, 80, 70, 60, 50.$$

(a) Form the heap using the standard (Williams) algorithm. Show the heap as a tree. Show the heap as an array. Exactly how many comparisons did heap creation use?

(b) Start with the heap created in Part (a). Show the *array* after each element sifts down *after heap creation*. How many comparisons does each sift use? What is the total number of comparisons *after heap creation*?

Problem 2. We are going to repeat Problem (1) using Floyd's version for sifting. Consider an array of size nine with the numbers in the following order

$$40, 10, 20, 30, 90, 80, 70, 60, 50.$$

(a) Form the heap using the Floyd's sifting algorithm. Show the heap as an array. Exactly how many comparisons did heap creation use?

(b) Start with the heap created in Part (a). Show the *array* after each element sifts down using Floyd's sifting algorithm *after heap creation*. How many comparisons does each sift use? What is the total number of comparisons *after heap creation*?

Problem 3. Assume that we visit the nodes in a binary tree in the order that they are numbered in heapsort, but you may only move directly to nodes connected by tree edges. You want to start at the root and end at the root, and visit all nodes up to and including node n. You will have to visit intermediate nodes multiple times.

For example, assume that you want to visit up to node 3 then you will visit the nodes in order 1, 2, 1, 3, 1. If you want to end at node 7 you will visit the nodes in order 1, 2, 1, 3, 1, 2, 4, 2, 5, 2, 1, 3, 6, 3, 7, 3, 1. In the first example you will cross edges 4 times, in the second example you will cross edges 16 times.

We want to calculate how many edge crossings there are. To keep things simple we will only consider visiting up to the last node of a level. Then we can calculate by levels rather than node numbers. There are a number of ways of solving this problem but we want to do it with recurrences. Levels are normally numbered 0 for just the root (node 1), 1 for nodes 2 and 3, 2 for nodes 4, 5, 6, and 7, etc.

(a) Write a recurrence for the number edges crossed visited when completing k levels (levels 0 to $k-1$). (Note that $n = 2^k - 1$.)

(b) Solve your recurrence.

(c) Rewrite your answer in terms of n.

Problem 4. (Challenge Problem) Not all of these problems have been solved by the instructor so you are on your own.

(a) Calculate the number of edge crossings for ending at a general node n (not just completing a level).

(b) Assume that every time you visit a node you add its value to a sum that starts at 0. For example, ending at node 1 will give sum 1, and ending at node 3 will give sum 8 (1+2+1+3+1). Calculate this sum when completing level k.

(c) Calculate the above sum when completing (a general) node n.

[This problem will not count towards your grade on this homework.]