Fall 2000 **CMSC 451: Homework 1** Clyde Kruskal

Due at the start of class Tuesday, September 26, 2000.

Problem 1. Devise an algorithm that finds the second largest of four elements.

(a) Briefly describe the algorithm in English. How many comparisons steps does it use?

(b) Give a decision tree for it.

Problem 2. Let k be a positive integer. Assume you had a special hardware component that could find the minimum of k numbers in one step.

(a) Give a decision tree based lower bound for the the number of comparison steps needed to sort in the worst using this component.

(b) Give an efficient algorithm to sort using this component.

Problem 3. Consider the problem of sorting a sequence of n 0's and 1's using comparisons. For each comparison of two values x and y, the algorithm learns which of $x < y$, $x = y$, or $x > y$ holds.

(a) Give a decision tree based lower bound for the the number of comparisons needed to sort in the worst case. (Note that alternative arguments will give stronger lower bounds.)

(b) Give an algorithm to sort in $n - 1$ comparisons in the worst case. Show that your algorithm is optimal.

(c) Give an algorithm to sort in $2n/3$ comparisons in the average case (assuming each of the n inputs is 0 or 1 with equal probability). Show that your algorithm is optimal.

Problem 4.

(a) Write a routine in psuedo-code to merge two sorted arrays A and B of sizes a and b into a third array C. Your routine should use $a + b - 1$ comparisons.

(b) Write a routine in psuedo-code to merge two sorted lists stored in the array A. The first list uses the first $n/2$ locations of A and the second list uses the remaining locations. Your routine should use $n - 1$ comparisons. Use as little extra storage as possible. How much extra storage do you need?

(c) What happens to the extra storage if the first list uses a locations and the second b locations where $a < b$ and $a + b = n$? Your routine should still use $n - 1$ comparisons (even though you could in principle now use fewer comparisons).

(d) Can you reduce the amount of extra storage by increasing the number of comparisons in a reasonable way?

Problem 5. Do Exercise 9.4-1 on page 182 of CLR.