Summer 2013 **CMSC 351: Homework 3** Clyde Kruskal

Due at the start of class Friday, June 7, 2013.

Problem 1. Assume your machine has 32 bit words. Assume you can multiply two n word numbers in time $3n^2$ with a standard algorithm. Assume you can multiply two n word numbers in time $20n^{\lg 3}$ with a "fancy" algorithm.

(a) Approximately, how large does n have to be for the fancy algorithm to be better?

(b) How many bits is that?

(c) How many decimal digits is that?

Problem 2. Use the same assumptions as for problem (1), except assume you can multiply two n word numbers in time only $10n^{\lg 3}$ with a "fancy" algorithm.

(a) Approximately, how large does n have to be for the fancy algorithm to be better?

(b) How many bits is that?

(c) How many decimal digits is that?

Problem 3.

(a) Consider the recurrence:

$$T(n) \;=\; 3T(n-2) + 2^{(n-1)/2}, \quad T(1) = 10$$

Calculate $T(7)$.

(b) Consider the recurrence:

$$T(n) \;=\; 3T(n/2) + 2^{n/2}, \quad T(1) = 10$$

Calculate $T(8)$.

Problem 4. Recall that in class we showed that a sorted list of size m and a sorted list of size n can be merged with $m + n - 1$ comparisons.

Consider the following "merge sort" algorithm for a list of size n, where n is even: Remove the last two elements and recursively sort the remaining $n-2$ elements. Sort the two removed elements with one comparison. Form the final sorted list by using the merge algorithm from class to merge the two sorted elements and the sorted list (of size $n-2$).

(a) Give a recurrence for the *exact* number of comparisons this algorithm uses.

(b) Use the iteration method to solve the recurrence. Simplify as much as possible.

(c) Use mathematical induction to verify your solution.