Math/CS 395 - Analysis of Algorithms - Spring 2012

<u>Homework 7</u>

Assigned: Friday, March 30, 2012
Due: **Friday, April 6, 2012**

The recurrence-tree method

1. (problem 4.4-2) Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = T(n/2) + n^2$. Use the substitution method to verify your answer.

2. (problem 4.4-4) Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = 2T(n-1) + 1$. Use the substitution method to verify your answer.

3. (problem 4.4-6) Argue that the solution to the recurrence $T(n) = T(n/3) + T(2n/3) + cn$, where c is a constant, is $\Omega(n \lg n)$ by appealing to a recursion tree.

The master method

4. (problem 4.5-1) Use the master method to give tight asymptotic bounds for the following recurrences.

   (a) $T(n) = 2T(n/4) + 1$

   (b) $T(n) = 2T(n/4) + \sqrt{n}$

   (c) $T(n) = 2T(n/4) + n$

   (d) $T(n) = 2T(n/4) + n^2$

5. (problem 4.5-2) Professor Caesar wishes to develop a matrix-multiplication algorithm that is asymptotically faster than Strassen's algorithm. His algorithm will use the divide-and-conquer method, dividing each matrix into pieces of size $n/4 \times n/4$, and the divide and combine steps together will take $\Theta(n^2)$ time. He needs to determine how many subproblems his algorithm has to create in order to beat Strassen's algorithm. If his algorithm creates a subproblems, then the recurrence for the running time $T(n)$ becomes $T(n) = aT(n/4) + \Theta(n^2)$. What is the largest integer value of a for which Professor Caesar's algorithm would be asymptotically faster than Strassen's algorithm?

6. (problem 4.5-3) Use the master method to show that the solution to the binary-search recurrence $T(n) = T(n/2) + \Theta(1)$ is $T(n) = \Theta(\lg n)$. (See exercise 2.3-5 for a description of binary search.)