CS 3343 (Spring 2008) Assignment 4

Due: Feb 13 before class starts

Note: As I mentioned in class, I expect you to follow the examples of master methods in the lecture slides and work out Problem #2 by yourself. However, if you think you still do not understand the master method after studying the examples, don't worry – we will go over more examples in detail on Monday. You can start working on problem #1 now however.

1. (30 points) On the course website, follow the links nearby HW4 to download three Java programs that I wrote to compute the power of two. The algorithms for two of the programs are very similar to what you have learned in class. Another one is new.

   a. (5 points) Compile the programs on any computer and use each program to compute 2^n with 5 to 10 values of n of your choice (for example, you may use $n = 10, 20, 30, 40, 50, 100, 500, 1000$). To compute 2^{10} using *Alg1*, you can type in *java Alg1 10*. Similar for *Alg2* and *Alg3*. The programs output two values: the value of 2^n, and the number of times that the function *power2* was called. What did you observe? Did the three algorithms give the same result for the same input? What is the largest n that each program can solve? Do you know why? (Answer this after you finished (b)-(d).) Be succint.

      Hint: use **Ctrl c** to stop a program that runs execessively long.

   b. (8 points) Study the algorithms in the three programs, and write down the recurrence for each algorithm.

   c. (12 points) Solve the three recurrence functions using either the recursion tree method or the iteration method. Do not worry about the base cases.

   d. (5 points) Does the second output (i.e., the number of times that *power2* was called) match your expection? Why or why not?

2. (30 points) Assume that $T(1) \in \Theta(1)$. Solve the following recurrence functions using the master method to derive a tight bound (Θ). If the master method cannot be applied, state the reason, and give an upper bound (big-Oh) as tight as you can. Always justify your answer.

   a. $T(n) = 4T(n/2) + n$;

   b. $T(n) = 9T(n/3) + n^2$;

   c. $T(n) = 6T(n/4) + n$;

   d. $T(n) = 2T(n/4) + n$;

   e. $T(n) = T(n/2) + n \log n$;

   f. $T(n) = 4T(n/4) + n \log n$;