CS 3333 Analysis of Algorithms – Fall 07

9/25/07

4. Homework

Due **10/2/07** before class

Always justify your answers.

1. **Guessing and Induction**

   For each of the following recurrences find a good guess of what it could solve to (make your guess as tight as possible). Use the recursion tree method to find your guess. Then prove that $T(n)$ is in big-Oh of your guess by induction (inductive step and base case).

   Every recursion below is stated for $n \geq 2$, and the base case is $T(1) = 1$.

   (a) $T(n) = 4T(\frac{n}{4}) + 3n$
   Hint: In order to make the proof simpler use $\log_4 n$ *instead of* $\log_2 n$.

   (b) $T(n) = 2T(\frac{n}{2}) + 5n^2$

2. **Strassen's Algorithm**
   Apply Strassen's algorithm to compute

$$\begin{pmatrix} 1 & 0 & 2 & 1 \\ 4 & 1 & 1 & 0 \\ 0 & 1 & 3 & 0 \\ 5 & 0 & 2 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 & 0 & 1 \\ 2 & 1 & 0 & 4 \\ 2 & 0 & 1 & 1 \\ 1 & 3 & 5 & 0 \end{pmatrix}$$

   The recursion should exit with the base case $n = 1$, i.e., 2×2 matrices should recursively be computed using Strassen's algorithm. In order to save you some work, you may assume that the following is a partial solution and you only have to fill in the missing values by using Strassen's algorithm:

$$\begin{pmatrix} 5 & 4 & & \\ 4 & 5 & & \\ 8 & 1 & 3 & 7 \\ 5 & 8 & 7 & 7 \end{pmatrix}$$