CS 2233 Discrete Mathematical Structures – Fall 08

10/22/08

# 6. Homework

Due **11/3/08** before class

1. **Guessing**
   For each of the following recurrences use either the expansion method or the recursion tree method to find a guess of what it could solve to. Make your guess as tight as possible. *(Hint: Use* $\log_3 n$ *instead of* $\log_2 n$*.)*

   (a) (2 points) $T(1) = 1$, and $T(n) = 3T(\frac{n}{3}) + 1$ for $n \geq 2$.

   (b) (2 points) $T(1) = 1$, and $T(n) = 3T(\frac{n}{3}) + n$ for $n \geq 2$.

   (c) (2 points) $T(1) = 1$, and $T(n) = 3T(\frac{n}{3}) + n^2$ for $n \geq 2$.

2. **Big-Oh Induction** (3 points)
   Let $T(n) = 2T(\frac{n}{2}) + 5n^2$ for $n \geq 2$ and $T(1) = 1$.
   Use induction to prove that $T(n) \in O(n^2)$.

3. **Master Theorem**
   Use the master theorem to prove the following claims. Justify your answers.

   (a) (2 points) $T(n) = 9T(\frac{n}{3}) + 1$

   (b) (2 points) $T(n) = 9T(\frac{n}{3}) + n$

   (c) (2 points) $T(n) = 16T(\frac{n}{4}) + n^2$

   (d) (2 points) $T(n) = 8T(\frac{n}{2}) + n^4$

4. **Divide and Conquer**
   Suppose you want to compute $n * 5$ in a programming language that does not have a built-in multiplication operator. The only operators you are allowed to use are **addition**, **subtraction**, **multiplication by 2**, and **division by 2** (the latter two are allowed because they only involve shifting on the bit level).

   (a) (3 points) Write a divide-and-conquer algorithm `int multiply(int n)` that computes $n * 5$, for any $n \geq 1$, using only the allowed operations. Your algorithm should run in $O(\log n)$ time. *(Hint: Use the fact that* $n * 5 = 5 + 5 + 5 + \ldots + 5$*.)*

   (b) (1 point) What is the runtime recurrence for your algorithm?

   (c) (1 point) Why does the runtime recurrence solve to $O(\log n)$?