CS 3343 Analysis of Algorithms – Fall 10

9/21/10

4. Homework

Due **9/28/10** before class

1. **Master theorem (10 points)**
   Use the master theorem to solve the following recurrences. Justify your results. Assume that $T(1) = 1$.

   (a) $T(n) = 3T(\frac{n}{3}) + 3$
   (b) $T(n) = 64T(\frac{n}{2}) + n^6 \log^2 n$
   (c) $T(n) = 27T(\frac{n}{3}) + n \log n$
   (d) $T(n) = T(\frac{n}{2}) + \sqrt{n}$
   (e) $T(n) = 64T(\frac{n}{4}) + n^3$

2. **Multiplying polynomials (10 points)**
   A polynomial of degree n is a function

$$p(x) = \sum_{i=0}^{n} a_i \cdot x^i = a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \cdots + a_1 \cdot x + a_0$$

   where a_i are constants and $a_n \neq 0$. For example, $4 \cdot x^3 + 5 \cdot x^2 + 2 \cdot x + 1$ is a polynomial of degree 3. For simplicity you may assume that $n+1$ is a power of 2.

   (a) (1 point) What is the runtime of the straight-forward algorithm for multiplying two polynomials of degree n?

   (b) (5 points) We can rewrite the polynomial $a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \cdots + a_1 \cdot x + a_0$ as $x^{(n+1)/2}(a_n \cdot x^{(n-1)/2} + a_{n-1} \cdot x^{(n-1)/2-1} + \cdots + a_{(n-1)/2+2} \cdot x + a_{(n-1)/2+1}) + (a_{(n-1)/2} \cdot x^{n/2-1} + \cdots + a_1 \cdot x + a_0)$. This effectively divides the polynomials into two parts of half the size each. For example, $4 \cdot x^3 + 5 \cdot x^2 + 2 \cdot x + 1$ can be rewritten as $x^2(4 \cdot x + 5) + (2 \cdot x + 1)$. And $7 \cdot x^7 + 6 \cdot x^6 + 5 \cdot x^5 + 4 \cdot x^4 + 3 \cdot x^3 + 2 \cdot x^2 + 1 \cdot x + 9$ can be rewritten as $x^4(7 \cdot x^3 + 6 \cdot x^2 + 5 \cdot x + 4) + (3 \cdot x^3 + 2 \cdot x^2 + 1 \cdot x + 9)$; here $n = 7$.

   Use this as a starting point to design a divide-and-conquer algorithm for multiplying two degree-n polynomials (recurse on polynomials of degree $(n-1)/2$). Give a runtime analysis of your algorithm by setting up and solving a recurrence. The runtime of your algorithm should be the same as the runtime of part a).

   (c) (1 point) Show how to multiply two degree-1 polynomials $a \cdot x + b$ and $c \cdot x + d$ using only three multiplications. *Hint: One of the multiplications is* $(a+b) \cdot (c+d)$.

   (d) (3 points) Design a divide-and-conquer algorithm for multiplying two polynomials of degree n in time $\Theta(n^{\log_2 3})$. *Hint: Reuse part* **b)** *and speed it up with the knowledge of part* **c)**