CSE 331: Introduction to Algorithm Analysis and Design Fall 2009

Homework 9

Due Friday, November 20, 2009 in class by 1:10pm

For general homework policies and our suggestions, please see the policy document.
Do not forget to look at Jeff's grading rubric before you write down your solutions.
Please note that I **will not accept any homeworks after 1:10pm even if it is a second late**.
No collaboration is allowed on problem 1(a) and do not hand in problem 4.

1. ($10 + 30 = 40$ points) We will consider the problem of exponentiating an integer to another. In particular, for non-negative integers a and n, define $\textsc{Power}(a,n)$ be the number a^n. (For this problem assume that you can multiply two integers in $O(1)$ time.)

   (a) (**NO collaboration allowed on this problem**) Present a naive algorithm that given non-negative integers a and n computes $\textsc{Power}(a,n)$ in time $O(n)$. (For this part, there is no need to prove correctness of the naive algorithm.)

   (b) Present a divide and conquer algorithm that given non-negative integers a and n computes $\textsc{Power}(a,n)$ in $O(\log n)$ time.

   Note: To get full credit you must present a recursive divide and conquer algorithm and then analyze its running time by solving a recurrence relation. (Justify the correctness of your algorithm– a formal correctness proof if not required.)

   Hint: Recall the following mathematical identity that holds for any real numbers b, c and d: $b^{c+d} = b^c \cdot b^d$.

2. (40 points) Exercise 1 in Chapter 5.

   Note: This problem is asking you to keep track of number of queries made by the algorithm and *not its running time*. Though generally we are interested in the running time of an algorithm, sometimes we are interested in other resource usage by the algorithm, which in this case is the number of queries made by the algorithm.

   Hint: Try to find two "distinguished" elements (one in each of the two lists). Try to see if you can rule out having to process some parts of the input depending on which of the distinguished elements is bigger.

3. (20 points) Exercise 3 in Chapter 5.

   Note: There is a clever algorithm not based on the divide and conquer strategy but the one based on divide and conquer is easier to think of.

4. (**DO NOT hand this problem in**) Exercise 4 in Chapter 5.