Comprehensive Exam on Data Structures and Algorithms: Fall 2007

Instructions: Choose any four of the following six questions to answer. Clearly mark which question you want graded on the front of the exam by placing an X in the appropriate slot.

Problem	1	2	3	4	5	6
Graded?						

Note that you will be graded not just on the answers, but also on work. An answer alone will not be worth much if anything at all. No books, calculators, notes, talking, etc. Using any communications device at all (cell phones, walkie talkies, etc.) will be an automatic failing grade.

WARNING: If you do not mark the questions you want graded or mark them ambiguously (i.e. if you decide to mark 5 questions rather than 4, etc.), we will grade your lowest 4 questions.

1. (a) Give an example of two functions $f(n)$ and $g(n)$ for which $f(n) \neq \Omega(g(n))$ and $g(n) \neq \Omega(f(n))$.

(b) Given the function $f(n) = (1 + \frac{1}{n})^{n \log n}$, indicate which of the following classifications best describes its growth: constant, linear, log-linear, polynomial, superpolynomial, or exponential. Justify your answer with natural-language sentences and mathematics.

2. Can the Master Method be applied to the recurrence $T(n) = 4T(n/2) + n^2 \lg n$? Why or why not? Give details for your reasoning.

3. (a) Suppose you have a set of n numbers and you ask a friend to silently choose one. Describe an $O(\log n)$ algorithm for determining the chosen number by asking your friend a sequence of "yes-or-no" questions. Describe the questions and justify why $O(\log n)$ questions are needed.

(b) Repeat the previous problem, but now suppose that your friend is allowed to lie (i.e. answer a question untruthfully) at most k times, where $k > 0$ is a constant. Again describe an $O(\log n)$ algorithm for finding the number.

4. Using dynamic programming, find an optimal parenthesization of a matrix-chain product whose sequence of dimensions is $\langle 5, 10, 20, 12, 5, 6\rangle$. To get full credit you must show **all steps**, i.e. the table $m[i, j]$ etc.

5. Let DOUBLE-SAT denote the decision problem that takes as input a Boolean formula F, and decides if F has two or more satisfying assignments. Prove that DOUBLE-SAT is an NP-complete problem.

6. Richard Branson wants to fly his plane around the world from Singapore to Singapore. His plane's gas tank when full holds enough gas to fly n miles (under all conditions) and his map gives the distances between all potential refueling stations $r_1, \ldots, r_n$ on his route. Branson wishes to make as few refueling stops as possible along the way. Give an efficient algorithm by which Richard Branson can determine at which refueling stations he should land, and **prove** that your algorithm yields an optimal solution.