Comprehensive Exam on Data Structures and Algorithms: Spring 2008

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. Let $f(n)$ and $g(n)$ be asymptotically positive functions. Decide whether each of the following statements is true or false **and** give either a proof or a counterexample.

   (a) $f(n) = O(g(n))$ implies that $g(n) = O(f(n))$.

   (b) $f(n) = O(g(n))$ implies that $2^{f(n)} = O(2^{g(n)})$.

   (c) $f(n) = O((f(n))^2)$.

2. Suppose that a hash table has n cells (with indices $0, 1, \ldots, n-1$), and that each cell can hold a single object. Moreover, the hash function h is a random-index generator, in that it will hash an object to any cell with equal probability. Finally, in the event of a collision, the object is rehashed using h until it is mapped to an open cell. Assuming that $n/2$ objects are to be hashed into the empty table, on the average how many calls to h will be made until all objects have been successfully hashed? Prove your answer. You may assume that all outputs of h are probabilistically independent.

3. Solve the recurrence $T(n) = 2T(\sqrt{n}) + \lg n$ be making a **change of variables**. Do not worry whether $\sqrt{n}$ is an integer.

4. The input for this problem is a set of n tasks $a_1, \ldots, a_n$. The tasks are to be executed by a single processor starting at time $t = 0$. Each task a_i requires one unit of processing time, and has an integer deadline d_i. Moreover, if the processor finishes executing a_i at time t, where $t \leq d_i$ then a profit p_i accrues (similarly, zero profit accrues if the task is not completed by its deadline). For example, if task a_1 has a deadline of 3 and a profit of 10, then it must be either the first, second, or third task executed in order to accrue the profit.

   (a) Describe in detail a greedy algorithm which determines a schedule that maximizes the sum of all accrued profits.

   (b) Apply your algorithm from part a to the following problem instance.

a_i	1	2	3	4	5	6	7	8	9	10	11
d_i	4	3	1	4	3	1	4	6	7	2	5
p_i	40	50	10	30	60	20	60	10	40	20	50

5. Suppose the matrix $C[1...n, 1...n]$ contains the cost $C[i, j]$ of flying **directly** from airport i to airport j. Consider the problem of finding the cheapest flight from k to l where we may fly to as many intermediate airports as desired.

   (a) Prove that this problem has the optimal substructure property.

   (b) Does the problem of finding the costliest trip also have the optimal substructure property (without revisiting of airports, i.e. on each trip from k to l we can visit each airport at most once)? Prove your claim or give a counterexample.

6. Given a simple graph $G = (V, E)$, an m-clique of G is a collection of m of its vertices that are all pairwise adjacent. For example, when $m = 3$, a 3-clique would be three pairwise adjacent vertices, thus forming a triangle. The m-CLIQUE problem is the problem of deciding if a graph has an m-clique. Prove that m-CLIQUE is NP-complete. Hint: use 3-SAT.