Fall 2007 Theory Comprehensive Exam

Answer any 4 out of 7 questions. State clearly which questions you want us to grade; we will only grade those 4, and only they will contribute to your score. The exam is closed-book and closed-notes.

If you run out of time, the idea of the proof is much more important (and will earn more partial credit) than mathematical formalism.

1. You have T days to sell as many widgets as possible. Your company provides you with 1) a graph G where each vertex represents a city and there is an edge between two cities if it's possible to drive from one city to the other one overnight, and 2) an uncannily accurate prediction matrix W where $W(v,t)$ gives the number of widgets you will sell if you spend day t in city v. Note that for different days t and t', $c(v,t)$ might be different from $c(v,t')$ because the demand for widgets in each city varies with time.

   Design a polynomial-time algorithm that given G, W, and a start vertex s in G, will return a path of T vertices through G that will maximize the number of widgets sold. What is the time and space cost of your algorithm?

2. Assume you are given a list of real numbers. Your goal is to determine if there is a pair of numbers in this list whose product is exactly 1. What is the fastest deterministic algorithm you can devise to solve this problem? Assume that you can do exact multiplication and division in $O(1)$ time.

   Now suppose you have access to a hash table. Can you use it to get a faster algorithm? What properties would you like the hash function to have?

3. Assume that a set of n distinct numbers are inserted into a binary tree in random order. What is the expected number of times that the minimum value in the tree changes?

4. A conjunctive normal form (CNF) formula is an AND of ORs, such as

   $$(x_2 \vee x_3 \vee x_7) \wedge (\overline{x_3} \vee x_6 \vee x_8) \wedge \cdots$$

   CNF-satisfiability is the question of whether there is any truth assignment for the variables $\{x_i\}$ that makes the formula true. This problem is NP-complete, since 3-SAT is a special case of it. On the other hand, a disjunctive normal form (DNF) formula is an OR of ANDs, such as

   $$(x_1 \wedge \overline{x_7}) \vee (\overline{x_9} \wedge x_{13} \wedge x_{14}) \vee \cdots$$

Show that DNF-satisfiability can be solved in polynomial time. Show also that any CNF formula can be converted into an equivalent DNF formula. Why doesn't this imply that P = NP?

5. A cellular automaton is a rule for updating a string of bits. Suppose we have the rule

$$000 \to 0,\ 001 \to 1,\ 010 \to 1,\ 011 \to 0,\ 100 \to 1,\ 101 \to 0,\ 110 \to 0,\ 111 \to 0$$

Then we go from one string to the next by applying this rule everywhere. For instance,

before :	1	0	1	0	0	1	1
after :		0	1	1	1	0	

We say a string s of length n has a *predecessor* if there is a string t of length $n+2$ such that applying the rule to t gives s. For instance, the example above shows that, for this rule, the string 01110 has a predecessor, namely 1010011; note that the predecessor might not be unique. (Note that s is two symbols shorter than t since we assume that we don't know any symbols to the left or right of t's endpoints.)

Now, given a cellular automaton rule, consider the set of all strings that have predecessors. Is this set a regular language? If not, how hard is it to determine whether a string has a predecessor; is it NP-complete?

Finally, how would your answer change if we looked a two-dimensional cellular automaton, in which the rule depends on a small neighborhood in the grid, and we asked which two-dimensional arrays of symbols were reachable? (You don't need to prove your answer to the two-dimensional case; just make an educated guess.)

6. 1-in-3 SAT is a variant of 3-SAT where each clause has three literals, say $(x_1, \overline{x_3}, x_5)$, and a clause is satisfied if *exactly one* literal in the clause is true. (Note the difference from 3-SAT, which requires at least one literal to be true.) A 1-in-3 SAT formula is then a list of clauses. Show that for any 1-in-3 SAT formula, there exists a truth assignment which satisfies at least $3/8$ of the clauses.

7. Each day, I choose to wear a red shirt, a blue shirt, or a green shirt. I like all three colors equally, but I don't want to wear the same color two days in a row. So, each day I flip a coin, and choose randomly from between the two colors other than the one I wore on the previous day: for instance, if I wore red yesterday, I choose randomly between blue and green.

Let $p(t)$ be the probability that I choose the same color today that I chose t days ago. For instance, $p(0) = 1$, $p(1) = 0$, and $p(2) = 1/2$. Show how to calculate $p(t)$ efficiently for any value of t, and if possible, find a simple closed-form expression for $p(t)$.