FALL 2006
COMPUTER SCIENCES DEPARTMENT
UNIVERSITY OF WISCONSIN-MADISON
PH. D. QUALIFYING EXAMINATION
Theory of Computing
Monday, September 18, 2006
3:00-7:00 PM

GENERAL INSTRUCTIONS:

1. Answer each question in a separate book.

2. Indicate on the cover of *each* book the area of the exam, your code number, and the question answered in that book. On one of your books list the numbers of all the questions answered. *Do not write your name on any answer book.*

3. Return all answer books in the folder provided. Additional answer books are available if needed.

SPECIFIC INSTRUCTIONS:

Answer questions 1 and 2, and, as time permits, two of the remaining three questions.

POLICY ON MISPRINTS AND AMBIGUITIES:

The Exam Committee tries to proofread the exam as carefully as possible. Nevertheless, the exam sometimes contains misprints and ambiguities. If you are convinced a problem has been stated incorrectly, mention this to the proctor. If necessary, the proctor can contact a representative of the area to resolve problems during the *first hour* of the exam. In any case, you should indicate your interpretation of the problem in your written answer. Your interpretation should be such that the problem is nontrivial.

Theory Qual

Fall 2006

Please answer questions 1 and 2, and, as time permits, two of the remaining three questions.

1. A number of friends working for the same company decide to carpool to work on some of the days. Say there are m days and on day i the set P_i of people decide to carpool. On each day a driver must be chosen from among the set of people that carpool. The friends want the scheme to be fair in the following sense. Let I_j denote the set of days i on which person j carpools. Then j's share of driving is $s_j = \sum_{i \in I_j} 1/|P_i|$. The fairness criterion simply requires that person j drive no more than $\lceil s_j \rceil$ times.

   For example, if the set of people going to work on the first three days is $\{A, B\}$, $\{A, B, C\}$ and $\{B, C\}$, then A, B, C is a fair and valid schedule and so is B, B, C, but A, A, B is not.

   Prove that such a fair schedule always exists and give a polynomial-time algorithm for constructing it.

2. Recall that $\oplus$P denotes the class of languages L for which there exists a nondeterministic polynomial-time Turing machine M such that an input x is in L iff M on input x has an odd number of accepting computation paths. The class BP$\cdot\oplus$P denotes the class of languages L' for which there exists a language L in $\oplus$P and an integer c such that for any input x,

$$\begin{aligned} x \in L' &\Rightarrow \Pr_{|y|=|x|^c}[\langle x, y\rangle \in L] \geq 2/3, \\ x \notin L' &\Rightarrow \Pr_{|y|=|x|^c}[\langle x, y\rangle \in L] \leq 1/3. \end{aligned}$$

   (a) Show that $(\oplus\mathrm{P})^{\oplus\mathrm{P}} = \oplus\mathrm{P}$.

   (b) Show that $\mathrm{BP}\cdot\oplus\mathrm{P} = (\Sigma_2^p)^{\oplus\mathrm{P}}$. Hint: Prove both inclusions separately.

3. Show that the following problem is NP-complete: Given a 2-CNF formula ϕ and an integer k, does there exist an assignment that satisfies at least k of the m clauses of ϕ?

   You can assume any of the standard NP-complete problems such as 3-SAT, vertex cover, traveling salesperson, etc.

4. This question is concerned with the function

$$f(n) = \# \text{ of integer pairs } (x, y) \text{ such that } x^2 + y^2 = n.$$

Here, n is restricted to be nonnegative.

(a) Give an algorithm that will evaluate $f(n)$ using $O(n^{1/2})$ steps. Here a "step" is one of the 4 basic arithmetic operations $(+, -, *, \div)$ or a comparison test.

(b) Show that the set of n on which $f = 0$ has positive density. This means that $f(n) = 0$ for $\Omega(N)$ of the $n \leq N$.

5. In this question we analyze a contention resolution system in a balls-and-bins framework. We have n resources (bins) and n jobs (balls); we allocate resources to jobs in rounds. In the first round, we throw all the n balls into n bins uniformly at random. After round $i \geq 1$, we remove every ball that occupies a bin by itself in round i, and in the next round $i + 1$ we redistribute the remaining balls uniformly at random into the n bins. The process continues until no balls are left.

(a) If the number of balls thrown in round i is m, what is the expected number of balls thrown in round $i + 1$?

(b) Prove that the expected number of rounds before we run out of balls is $O(\log \log n)$.

GOOD LUCK!!