PhD Qualifier Exam for Spring 2006—— Theory Area

Spring 2006, CS Department, IIT

Your random number____

There are 4 questions in this exam. For every question, please write your answer in a clean and concise way.

If you are asked to write an algorithm for a question, you have to write the **pseudo-code** of your algorithm and also put explanation about your pseudo-code.

1. Suppose that you are given an $n \times n$ checkerboard and a checker. You must move the checker from the bottom edge of the board to the top edge of the board according to the following rule. At each step you may move the checker to one of three squares:

   (a) the square immediately above,

   (b) the square that is one up and one to the left (but only if the checker is not already in the leftmost column),

   (c) the square that is one up and one to the right (but only if the checker is not already in the rightmost column).

   Each time you move from square x to square y, you receive $p(x, y)$ dollars. You are given $p(x, y)$ for all pairs (x, y) for which a move from x to y is legal. Do **not** assume that $p(x, y)$ is positive.

   Give an algorithm that figures out the set of moves that will move the checker from somewhere along the bottom edge to somewhere along the top edge while gathering as many dollars as possible. Your algorithm is free to pick any square along the bottom edge as a starting point and any square along the top edge as a destination in order to maximize the number of dollars gathered along the way. What is the running time of your algorithm?

2. Suppose you are given two sets A and B, each containing n positive integers. You can choose to reorder each set however you like. After reordering, let a_i be the i^{th} element of set A, and let b_i be the i^{th} element of set B. You then receive a payoff of $\prod_{i=1}^{n} a_i^{b_i}$. Give an algorithm that will maximize your payoff. Prove that your algorithm maximizes the payoff, and state its running time.

3. Arbitrage is the use of discrepancies in currency exchange rates to transform one unit of a currency into more than one unit of the same currency. For example, suppose that 1 U.S. dollar buys 46.4 Indian rupees, 1 Indian rupee buys 2.5 Japanese yen, and 1 Japanese yen buys 0.0091 U.S. dollars. Then, by converting currencies, a trader can start with 1 U.S. dollar and buy $46.4 \times 2.5 \times 0.0091 = 1.0556$ U.S. dollars, thus turning a profit of 5.56 percent.

   Suppose that we are given n currencies c_1, c_2, $\cdots$, c_n and an $n \times n$ table R of exchange rates, such that one unit of currency c_i buys $R[i, j]$ units of currency c_j.

   (a) Give an efficient algorithm to determine whether or not there exists a sequence of currencies $< c_{i_1}, c_{i_2}, \cdots, c_{i_{k-1}}, c_{i_k} >$ such that

   $$R[i_1, i_2] \cdot R[i_2, i_3] \cdots R[i_{k-1}, i_k] \cdot R[i_k, i_1] > 1.$$

   Analyze the running time of your algorithm.

   (b) Give an efficient algorithm to print out such a sequence if one exists. Analyze the running time of your algorithm.

4. Assume that you *only* know the following problems are NP-complete: SAT, 3SAT, VERTEX-COVER, CLIQUE.

   Consider the following set cover problem. Given a collection of m sets $S_1, S_2, \cdots, S_m$, are there K of these sets whose union is the same as the whole collection of m sets?

   Prove that the set cover problem is NP-complete.