Theory of Computation

Final Examination on June 19, 2008
Spring Semester, 2008

Problem 1 (20 points)**.** Show that if SAT $\in$ P, then FSAT has a polynomial-time algorithm. (Hint: You may want to use the self-reducibility of SAT.)

Problem 2 (20 points)**.** Let $U = \{u_1, \ldots, u_n\}$, $V = \{v_1, \ldots, v_n\}$ and $G = (U, V, E)$ be a bipartite graph with a perfect matching. Consider the $n \times n$ matrix $A^G(x_{11}, \ldots, x_{nn})$ whose (i, j)-th entry is a variable x_{ij} if $(u_i, v_j) \in E$ and zero otherwise. Does there exist an integer assignment $i_{11}, \ldots, i_{nn}$ to $x_{11}, \ldots, x_{nn}$ such that $\det(A^G(i_{11}, \ldots, i_{nn})) \neq 0$?

Problem 3 (20 points)**.** For $c \in [\,0, 1\,]$, let $P(c)$ be the following statement:

> There exists a randomized polynomial-time algorithm outputting "Hamiltonian" with probability at least c when its input is a Hamiltonian graph, and "Not Hamiltonian" with probability 1 otherwise.

Show that $P(3/5)$ implies $P(3/4)$.

Problem 4 (20 points)**.** Let M be a polynomial-time Turing machine that, given as input an odd prime p, a primitive root g of p and $-g^x \bmod p$ for an unknown x, finds $x \bmod (p-1)$. Show how to break the discrete logarithm in polynomial time. That is, given an odd prime p, a primitive root g of p and $g^x \bmod p$ for an unknown x, show how to find $x \bmod (p-1)$ in time polynomial in the length of the inputs. (Hint: You may want to consider $g^{(p-1)/2} \bmod p$.)

Problem 5 (20 points)**.** Does PRIMES belong to IP? Briefly justify your answer.

Problem 6 (20 points)**.** Prove that INDEPENDENT SET is NP-hard. You may assume the NP-completeness of CLIQUE or any other problem shown to be NP-complete in class.