Theory of Computation

Final Examination on January 12, 2010

Problem 1 (25 points)**.** Let p be a prime and $m \in \Phi(p)$ have exponent k modulo p. Prove that $k \mid p-1$.

Proof. Suppose $k \nmid p-1$ for contradiction. Let $p-1 = qk+a$ for $0 < a < k$. Then $m^a = m^{qk+a} = m^{p-1} = 1 \bmod p$ (where the last equality is Fermat's little theorem), a contradiction to the premise that k is the exponent of m. $\square$

Problem 2 (25 points)**.** Suppose $x_1, \ldots, x_n$ are independent random variables taking the values 1 and 0 with probabilities p and $1-p$, respectively. Let $t > 0$ and $X = \sum_{i=1}^n x_i$. Show that

$$\text{prob}\,[\,X \geq 2pn\,] \leq \frac{\prod_{i=1}^n E\,[\,e^{t\,x_i}\,]}{e^{2tpn}}.$$

You may want to use the fact

$$\text{prob}\,[\,X \geq 2pn\,] = \text{prob}\left[\,e^{tX} \geq e^{2tpn}\,\right].$$

Proof. By Markov's inequality,

$$\text{prob}\left[\,e^{tX} \geq e^{2tpn}\,\right] \leq \frac{E\left[\,e^{tX}\,\right]}{e^{2tpn}}.$$

As the x_i's are independent,

$$E\left[\,e^{tX}\,\right] = \prod_{i=1}^n E\left[\,e^{t\,x_i}\,\right].$$

So

$$\text{prob}\,[\,X \geq 2pn\,] = \text{prob}\left[\,e^{tX} \geq e^{2tpn}\,\right] \leq \frac{E\left[\,e^{tX}\,\right]}{e^{2tpn}} = \frac{\prod_{i=1}^n E\,[\,e^{t\,x_i}\,]}{e^{2tpn}}.$$

$\square$

Problem 3 (25 points)**.** Let A be a deterministic polynomial-time algorithm such that (1) for each prime p and primitive root g of p,

$$|\{x : x \in \{0, 1, \dots, p-2\}, A(p, g, g^x) = x\}| \geq \frac{p-1}{2},$$

and (2) for each $x \in \{0, 1, \dots, p-2\}$ with $A(p, g, g^x) \neq x$, $A(p, g, g^x) =$ "fail." That is, A solves the discrete logarithm problem for at least half of the exponents and reports failure whenever it fails to solve the discrete logarithm problem. Find a randomized polynomial-time algorithm B that solves the discrete logarithm problem with probability at least $1/2$. In other words, given any prime p, primitive root g of p and $g^x \bmod p$ for *any* $x \in \{0, 1, \dots, p-2\}$, B outputs x with probability at least $1/2$.

Proof. Given a prime p, a primitive root g of p and $g^x \bmod p$, B finds $y = A(p, g, g^{x+r})$ where r is uniformly distributed over $\{0, \dots, p-2\}$. If $y \neq$ "fail", then B outputs $y - r \bmod p-1$. Note that $y \neq$ "fail" means $g^y = g^{x+r} \bmod p$, which together with $g^{p-1} = 1 \bmod p$ gives $g^{y-r \bmod p-1} = g^x \bmod p$. So B correctly breaks discrete logarithm when $y \neq$ "fail," which happens with probability at least $1/2$.

Calculating $g^{x+r} \bmod p$ from $g^x \bmod p$ takes polynomial time by the method of recursive doubling. Simulating A also takes polynomial time. So B is a randomized polynomial-time algorithm. □

Problem 4 (25 points)**.** Let $\Phi = \{\phi_1, \dots, \phi_m\}$ be a set of Boolean expressions in n variables. For $1 \leq i \leq m$, assume that ϕ_i involves exactly k variables and is satisfied by exactly one of the 2^k truth assignments to the k variables. Show that there exists a truth assignment T satisfying at least $\sum_{i=1}^{m} 1/2^k$ expressions in Φ.

Proof. A random truth assignment satisfies ϕ_i with probability $1/2^k$ for $1 \leq i \leq m$. Hence it satisfies an expected $\sum_{i=1}^{m} 1/2^k$ expressions in Φ. So there must be a truth assignment satisfying at least $\sum_{i=1}^{m} 1/2^k$ expressions in Φ. □