CS 70 Discrete Mathematics for CS Spring 2000 Papadimitriou/Russell/Sinclair Midterm 2

You have 1 hour and 20 minutes. The exam is open-book, open-notes.
100 points total (4 questions of 25 points each).

You should be able to finish all the questions, so avoid spending too long on any one question. Write your answers in blue books. Check you haven't skipped any by accident. Hand them all in. Panic not.

1. (25 pts.) Modular arithmetic

(a) Use the Extended GCD Algorithm to find the inverse of 5 mod 36. Show all your working.

(b) Hence solve for x the equation

$$5x + 19 = 35 \bmod 36.$$

(c) Does the equation

$$6x + 19 = 35 \bmod 36$$

have a solution? Justify your answer.

2. (25 pts.) Lagrange interpolation

(a) Find the lowest degree polynomial f (over the real numbers) such that $f(0) = 2$, $f(1) = 6$, $f(3) = 20$. [Hint: Remember to check your answer by substitution.]

(b) Prove that no polynomial of lower degree than the one in your answer to part (a) could possibly satisfy these three equations.

(c) **[Extra credit, 5 pts.]** Find another polynomial g (different from f above) such that g also satisfies $g(0) = 2$, $g(1) = 6$, $g(3) = 20$.

3. (25 pts.) Efficient double exponentiation

(a) Let a, k and p be positive integers such that p is prime. Use Fermat's Little Theorem to show that

$$a^k \bmod p = a^{k \bmod (p-1)} \bmod p.$$

(b) Use part (a) to design an algorithm that, given positive integers a, b, c and a prime p, outputs the value $a^{(b^c)} \bmod p$. Your algorithm should run in polynomial time (i.e., polynomial in the number of bits in the inputs a, b, c and p). [Hint: You may use the fact from class that there is an algorithm that computes $b^c \bmod m$, for any positive integers b, c and m, in polynomial time. You should justify carefully that your algorithm runs in polynomial time.]

4. (25 pts.) Cracking RSA

Suppose Bob's RSA public key is (e, n), where e is the encryption key, and $n = pq$ is the product of two primes. Alice has just sent a secret message $c = m^e \bmod n$ to Bob using Bob's public key.

(a) Explain how Bob can decrypt the message he has received.

(b) Now suppose that, by eavesdropping on their conversation, you managed to overhear the ciphertext c. Moreover, when crafting his public key Bob foolishly chose primes that were too small, so that by continuously running a fast factoring algorithm on one of Berkeley's supercomputing clusters for two weeks, you eventually manage to factor n, and recover p and q. Given e, p, q, and c, explain how you can now efficiently (in polynomial time) recover the plaintext m of Alice's message to Bob.