Homework 3

Due: Tuesday, July 16, 4:00pm

CS 70: Discrete Mathematics and Probability Theory, Summer 2013

1. [12 points] With these problems comes great responsibility.

   1a. [4 points] Calculate 4^{273} mod 11 using repeated squaring. Show your work. You should not need a calculator.

   1b. [4 points] Find a trick to calculate 4^{273} mod 11 more easily using Fermat's Little Theorem. Justify that your trick works.

   1c. [4 points] Similarly to part 1b, calculate $4^{5^{273}}$ mod 11 using Fermat's Little Theorem. (Note that $4^{5^{273}}$ means $4^{(5^{273})}$.)

2. [12 points] Bob decides to use RSA with $p = 11$, $q = 23$, and $e = 7$. Bob publishes $N = pq = 253$ and e as his public key. (You should convince yourself that e is indeed a valid encryption exponent.)

   2a. [4 points] Show how to find Bob's private key d using the extended Euclidean algorithm. (Hint: If you do it right, you should get $d = 63$.)

   2b. [4 points] Alice wants to send the message 44 (an integer between 0 and 252) to Bob. What is the encrypted message that Alice sends? You may use a calculator, but you must show all intermediate steps of the repeated squaring algorithm.

   2c. [4 points] Suppose Bob receives from Alice the ciphertext 103. What was the original message that Alice sent? You may use a calculator, but you must show all intermediate steps of the repeated squaring algorithm.

3. [10 points] Suppose we tried to simplify the RSA cryptosystem by using just a prime p instead of the composite modulus $N = pq$. Similarly to RSA, we would have an encryption exponent e that is relatively prime to $p - 1$, and the encryption of message x would be (x^e mod p). Show that this scheme is not secure by giving an efficient algorithm that, given p, e, and (x^e mod p), computes x mod p.

   3a. [3 points] Describe your algorithm.

   3b. [5 points] Prove that your algorithm is correct.

   3c. [2 points] Analyze the running time of your algorithm.

4. [14 points] Let f be a polynomial of degree at most d. The coefficient representation of f is the sequence $a_d, a_{d-1}, \ldots, a_1, a_0$ of coefficients of f. A point-value representation of f is

a collection $\{(x_1, f(x_1)),\ (x_2, f(x_2)),\ \ldots,\ (x_t, f(x_t))\}$ of the values of f at any t points $x_1, x_2, \ldots, x_t$, where $t \geq d+1$. (Recall from Lecture Note 7 that a polynomial of degree d is completely determined by its values at any $d+1$ points. Note that t may be greater than $d+1$, so more points than necessary may be given.) In the following questions, let f and g be any two real polynomials of degree at most d.

4a. [2 points] What is the maximum degree of the product polynomial fg?

4b. [4 points] Given coefficient representations of f and g, explain how to compute the coefficient representation of fg using $O(d^2)$ arithmetic operations (additions / subtractions / multiplications / divisions) over real numbers.

4c. [4 points] Now suppose that f and g are specified by point-value representations at t points for some $t \geq d+1$, i.e., f is specified as $(x_1, f(x_1)),\ (x_2, f(x_2)),\ \ldots,\ (x_t, f(x_t))$, and g as $(x_1, g(x_1)),\ (x_2, g(x_2)),\ \ldots,\ (x_t, g(x_t))$. With a suitable value of t (which you should specify), show how to compute a point-value representation of fg using only $O(d)$ arithmetic operations.

4d. [4 points] Suppose that polynomial g divides polynomial f, and that f, g are given in point-value representation as in part (4c) with $t = d+1$. Show how to compute a point-value representation for the quotient f/g using $O(d)$ arithmetic operations. **Be very careful proving that your algorithm is correct.**

5. [8 points] In this problem, you will give two different proofs of the following theorem: For every prime p, every polynomial over $GF(p)$ is equivalent to a polynomial of degree at most $p-1$. (Two polynomials f, g over $GF(p)$ are said to be equivalent iff $f(x) = g(x)$ for all $x \in GF(p)$.)

5a. [4 points] Show how the theorem follows from Fermat's Little Theorem. (Hint: Be careful! It is not true that $x^{p-1} \equiv 1 \pmod{p}$ for all $x \in \{0, 1, \ldots, p-1\}$. Why not?)

5b. [4 points] Now prove the theorem using what you know about Lagrange interpolation.

6. [10 points] Suppose we wish to share a secret among five people, and we decide to work modulo 7. We construct a degree-two polynomial $q(x) = ax^2 + bx + s$ by picking the coefficients a and b at random (mod 7); the constant term is the secret s (also a number mod 7). We give shares $q(1), \ldots, q(5)$ to each of the five people (all operations being done mod 7). Now suppose that three of the people get together and share the information that $q(1) = 5$, $q(2) = 2$, and $q(4) = 2$. Use Lagrange interpolation to find the polynomial q and the secret s. Show all your work.

7. [6 points] Consider the following variant of the secret sharing problem. We wish to share a secret among twenty-one people, divided into three groups of seven, so that the following condition is satisfied. A subset of the twenty-one people can recover the secret if and only if it contains majorities (at least four out of seven) of at least two of the groups. How would you

modify the standard secret sharing scheme to achieve this condition? (Hint: Try a two-level scheme, one level for groups, the other for people within the group.)

8. [10 points] Alice wants to send the message (a_0, a_1, a_2) to Bob, where each $a_i \in \{0, 1, 2, 3, 4\}$. She encodes it as a polynomial P of degree ≤ 2 over $GF(5)$ such that $P(0) = a_0$, $P(1) = a_1$, and $P(2) = a_2$, and she sends the packets $(0, P(0))$, $(1, P(1))$, $(2, P(2))$, $(3, P(3))$, $(4, P(4))$. Two packets are dropped, and Bob only learns that $P(0) = 4$, $P(3) = 1$, and $P(4) = 2$. Help Bob recover Alice's message. Use Lagrange interpolation and show all your work.

9. [9 points] Consider undirected graphs with multi-edges allowed, and self-loops not allowed. In class we proved that a graph has an eulerian tour if and only if it is connected (except possibly for isolated vertices) and every vertex has even degree. In this question we will consider what we can say about graphs that have a certain number of odd-degree vertices.

   9a. [3 points] Let $G = (V, E)$ be a graph on n vertices. Show that the number of vertices of G that have odd degree must be even. (There is a simple proof of this fact.)

   9b. [6 points] Now suppose G is connected and has exactly $2c$ vertices of odd degree. (We know from part (9a) that this number must be even.) Prove that it is possible to find exactly c paths (that can go through the same vertex more than once) that together cover all of the edges of G exactly once (i.e., each edge of G occurs in exactly one of the c paths, and that path contains the edge only once).

10. [9 points] A *tournament* is defined to be a directed graph such that for every pair of distinct nodes v and w, exactly one of (v, w) and (w, v) is an edge (representing which player beat the other in a round-robin tournament.) In an earlier lecture, we proved that if a tournament has a cycle, then it has a cycle of length 3. Now, prove that every tournament has a hamiltonian path (which visits each node exactly once, and need not end up back where it started). In other words, you can always arrange the players in a line so that each player beats the next player in the line. (Hint: There are at least two ways to prove this. One is a simple induction on the number of nodes, and the other is a strong induction on the number of nodes. Of course, you only need to give one proof!)