CS 70 Discrete Mathematics and Probability Theory
Spring 2011 Demmel HW 4

Due Feb 18, 8:00am

You may work in groups of up to 3 people (no larger!). Please read the group collaboration policies on bSpace or `www.cs.berkeley.edu/~demmel/cs70_Spr11` before beginning group work. You *must* write up the solution set entirely on your own. You must never look at any other students' solutions (not even a draft), nor share your own solutions (not even a draft).

Please begin your answer to each question on a new sheet of paper, and make sure that each sheet is labeled with your name, section number, GSI name, the assignment number, the question number, and "CS70–Spring 2011".

Turn in each question in a different box in 283 Soda Hall: Question 1 in the box labeled "CS70 - 1", Question 2 in the box labeled "CS70 - 2", etc. Reason: Different problems will be graded in parallel by different readers.

Warning: You risk receiving no credit, or losing points, for any homework that does not conform to the above regulations! Please take the time to write clear and concise solutions; we will not grade messy or unreadable submissions. No late homeworks will be accepted. We will drop the lowest two homework scores.

1. (10 pts.) Modular arithmetic

1. Give the addition and multiplication tables for modular-5 arithmetic. Write down the inverse for each of the elements which have one, and identify the ones which have no inverse.
2. Solve the following equations for x and y or show that no solution exists. Show your work (in particular, what division must you carry out to solve each case).
   (a) $5x+11 \equiv 6 \pmod{46}$
   (b) $16x+63 \equiv 3 \pmod{64}$
   (c) The system of simultaneous equations
   $30x+3y \equiv 0 \pmod{37}$ and $y \equiv 4+13x \pmod{37}$
3. Compute $\gcd(5694, 2016)$ and show your steps.

2. (6 pts.) GCD
In class we saw that, if $\gcd(m,x) = 1$ then there are m distinct elements in the set $\{\text{mod}(ax,m) : a \in \{0,\ldots,m-1\}\}$. If $gcd(m,x) > 1$, how many distinct elements are there? Prove your answer.

3. (15 pts.) Poker mathematics
A *pseudorandom number generator* is a way of generating a large quantity of random-looking numbers, if all we have is a little bit of randomness (known as the *seed*). One simple scheme is the *linear congruential*

generator, where we pick some modulus m, some constants a, b, and a seed x_0, and then generate the sequence of outputs $x_0, x_1, x_2, x_3, \ldots$ according to the following equation:

$$x_{t+1} = \bmod(ax_t + b, m)$$

(Notice that $0 \le x_t < m$ holds for every t.)

You've discovered that a popular web site uses a linear congruential generator to generate poker hands for its players. For instance, it uses x_0 to pseudo-randomly pick the first card to go into your hand, x_1 to pseudo-randomly pick the second card to go into your hand, and so on. For extra security, the poker site has kept the parameters a and b secret, but you do know that the modulus is $m = 2^{31} - 1$ (which is prime).

Suppose that you can observe the values x_0, x_1, x_2, x_3, and x_4 from the information available to you, and that the values $x_5, \ldots, x_9$ will be used to pseudo-randomly pick the cards for the next person's hand. Describe how to efficiently predict the values $x_5, \ldots, x_9$, given the values known to you.

4. **(15 pts.) RSA**

In class, we said that RSA uses as its modulus a product of two primes. Let's look at a variation that uses a single prime number as the modulus. In other words, Bob would pick a 1024-bit prime p and a public exponent e satisfying $2 \le e < p - 1$ and $\gcd(e, p - 1) = 1$, calculate his private exponent d as the inverse of e modulo $p - 1$, publish (e, p) as his public key, and keep d secret. Then Alice could encrypt via the equation $E(x) = \bmod(x^e, p)$ and Bob could decrypt via $D(y) = \bmod(y^d, p)$.

Explain why this variation is insecure. In particular, describe a procedure that Eve could use to recover the message x from the encrypted value y that she observes and the parameters (e, p) that are known to her. Analyze the running time of this procedure, and make sure to justify why Eve could feasibly carry out this procedure without requiring extravagant computation resources.

5. **(15 pts.) Simultaneous equations**

1. Solve the two simultaneous equations:

$$x \equiv 2 \bmod 3$$

$$x \equiv 3 \bmod 5$$

2. We want to solve the three simultaneous equations:

$$x \equiv a_1 \bmod m_1$$

$$x \equiv a_2 \bmod m_2$$

$$x \equiv a_3 \bmod m_3$$

Where each gcd(m_i, m_j) = 1. Prove that the following algorithm gives a solution and explain why the multiplicative inverses must exist.

Compute $M = m_1 * m_2 * m_3$
Compute $M_1 = M/m_1$, $M_2 = M/m_2$, and $M_3 = M/m_3$
Compute the multiplicative inverse of M_1 modulo m_1, call it y_1
Compute the multiplicative inverse of M_2 modulo m_2, call it y_2
Compute the multiplicative inverse of M_3 modulo m_3, call it y_3
Let $x = a_1 * M_1 * y_1 + a_2 * M_2 * y_2 + a_3 * M_3 * y_3$

3. In general, we can solve k simultaneous equations:

$$x \equiv a_1 \bmod m_1$$

$$x \equiv a_2 \bmod m_2$$

$$\vdots$$

$$x \equiv a_k \bmod m_k$$

Where each gcd(m_i, m_j) = 1 by extending the algorithm we used in part 2. Prove that we can find a solution for x using the following algorithm:

Compute $M = m_1 * m_2 * \cdots * m_k$
For all i where $1 \leq i \leq k$
    Compute $M_i = M/m_i$
    Compute the multiplicative inverse of M_i modulo m_i, call it y_i
Let $x = \sum_{i=1}^{k} a_i * M_i * y_i$

6. (6 pts.) Practice with prime numbers

Find all primes n, where $n+2$ and $n+4$ are also prime. Prove your answer is correct.