CS 70 Discrete Mathematics and Probability Theory
Fall 2011 Rao HW 4

Due September 23, 5:00pm

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 put your answer to each problem on its own sheet of paper, and paper-clip (don't staple!) the sheets of paper together. Label each sheet of paper with your name, your discussion section number (101–108), and "CS70–Fall 2011". Turn in your homework and problem x into the box labeled "CS70 – Fall 2011, Problem x" whereon the 2nd floor of Soda Hall. Failure to follow these instructions will likely cause you to receive no credit at all.

1. (10 pts.) Euclid's argument

Consider the following result, first proved many centuries ago.

Theorem (Euclid) There exist infinitely many primes.

Proof: Assume to the contrary that there exist finitely many primes. Let these primes (in increasing order) be $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, ..., p_k. Let $q_k = p_1 p_2 p_3 \cdots p_k + 1$. Note that q_k is a new number not in the list of primes $p_1, \ldots, p_k$. At the same time, it is not divisible by p_i for any i, since $q_k \equiv p_1 p_2 p_3 \cdots p_k + 1 \equiv 1 \pmod{p_i}$, which would mean that q_k is a new prime different from $p_1, \ldots, p_k$, which is a contradiction. This completes the proof.

Let $p_1, \ldots, p_k$ represent the first k primes. Are we guaranteed that $p_1 p_2 p_3 \cdots p_k + 1$ is always prime for all $k \geq 1$? Is the above proof valid? Explain your answers.

2. (20 pts.) GCD

(a) (3d from homework 3) Use Extended Euclid's algorithm to find some pair of integers j, k such that $52j + 15k = 3$. Show your work.

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

3. (25 pts.) Nonnegative Combinations

Given positive integers m and n with $\gcd(m,n) = 1$ and any integer k, the Eudlidean Algorithm finds integers x and y so that $mx + ny = k$. But what if we limit the choices of x and y to integers which are at least zero? Then we can't get every integer k.

(a) For every integer k, we know that k can be represented as the sum $mx + ny = k$ for integers x and y in many different ways. (For example, if $m = 3$, $n = 5$, $k = 7$, then $k = (-1)m + 2n = 4m + (-1)n$.) Prove that among all these representations, one and only one of them satisfies $0 \leq x < n$. Call such a representation *nice*.

(b) From (a), we know that k is a non-negative integral sum of m and n if and only if k has a nice representation $k = mx + ny$ with a non-negative integer y. Prove that the largest k that cannot be written as a non-negative integral sum of m and n is $mn - m - n$.

Hint: What are the largest possible values of x and y in the nice representation of such a k?

4. (15 pts.) Binary GCD

On most computers, the operations of subtraction, testing the parity (odd or even) of a binary integer, and halving can be performed more quickly than computing remainders. This problem investigates the binary gcd algorithm, which avoids the remainder computations used in Euclid's algorithm. (It may be helpful to note that if x and y are positive, and x divides y and y divides x, then $x = y$.)

1. Prove that for any positive integers d, x, and y, d divides $gcd(x,y)$ if and only if d divides x and d divdes y.
2. Prove that if a and b are both even, then $gcd(a,b) = 2gcd(a/2,b/2)$.
3. Prove that if a is odd and b is even, then $gcd(a,b) = gcd(a,b/2)$.
4. Prove that if a and b are both odd, then $gcd(a,b) = gcd((a-b)/2,b)$ where we assume $a \geq b$.
5. Design an efficient binary gcd algorithm that uses $O(\log(max(a,b)))$ subtractions, halving, and parity tests. (Do not use remainders.)

5. (10 pts.) Easy 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 $2 \leq e < p-1$ 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 $E(x) = \mod(x^e, p)$ and Bob could decrypt via $D(y) = \mod(y^d, p)$.

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.

6. (20 pts.) RSA

Let p and q be primes and let $N = pq$. Show how to determine p and q given N and $(p-1)(q-1)$. (In other words, given the public key (e,N), e the encryption exponent and N the RSA modulus, and the value $\varphi(N) = (p-1)(q-1)$, it is possible to compute p and q by simple (polynomial time) algebraic operations. This shows that determining $\varphi(N)$ is "as hard as factoring.")