Homework 6 due Friday 2/18/2015

Huimin He , section 1

February 18, 2015

1. 6.6 (c) show that $ab \equiv xy \pmod{m}$.

   Given $a \equiv x \pmod{m}$ and $b \equiv y \pmod{m}$, we have

   $$mk_1 = a - x$$

   and

   $$mk_2 = b - y$$

   where k_1, k_2 are integers.

   $$mk_2a = a(b - y) = ab - ay$$
   $$mk_1y = ay - xy$$

   Adding the above two equations we get

   $$m(k_2a + k_1y) = ab - xy$$

   since k_1, k_2, a, y are integers, m divides $ab - xy$. So by definition $ab \equiv xy \pmod{m}$ is proved.

2. 6.8 Eclid's rounds Exercise 2.3 of the handout

   let B_i,R_i, and q_i be the variable B, R, q after i iterations.We have

   $$B_{i+2} = B_i - B_{i+1}q_{i+2}$$

   By division theorem we know that

   $$0 \leq B_{i+2} < q_{i+2}$$

   Divide both sides by B_i So

   $$\frac{B_{i+2}}{B_i} = \frac{B_i}{B_i} - q_{i+2}\frac{B_{i+1}}{B_i}$$
   $$\frac{B_{i+2}}{B_i}(1 + q_{i+2}) = 1$$
   $$\frac{B_{i+2}}{B_i} = \frac{1}{1 + q_{i+2}}$$

   Since

   $$1 \leq q_{i+2}$$

   from $B_{i+1} < Bi$ So

   $$\frac{B_{i+2}}{B_i} \leq \frac{1}{2}$$

   for all i is proved.

3. 6.9 compute 21^-1 mod 76

part(a)

We want find x such that

$$21x \equiv 1 \pmod{76}$$

We know

$$76x \equiv 0 \pmod{76}$$

So

$$76x - 3 \times 21x \equiv 0 - 3 \pmod{76}$$

$$13x \equiv -3 \pmod{76}$$

substract this from the first equation

$$21x - 2 \times 13x \equiv 1 + 2 \times 3 \pmod{76}$$

$$-5x \equiv 7 \pmod{76}$$

substract this again from the equation above.

$$13x - 2 \times (-5x) \equiv -3 + 14 \pmod{76}$$

so

$$3x \equiv 11 \pmod{76}$$

$$2x \equiv -18 \pmod{76}$$

$$x \equiv 29 \pmod{76}$$

part(b)

$$\begin{aligned}
&gcd(228, 63) \\
&= gcd(228 - 3 \times 63, 63) \\
&= gcd(63, 39) \\
&= gcd(39, 63 - 39 \times 2) \\
&= gcd(39, -15) \\
&= gcd(9, -15) \\
&= gcd(-15 + 9 \times 2, 9) \\
&= gcd(9, 3) \\
&= gcd(6, 3) \\
&= gcd(3, 0) \\
&= 3
\end{aligned}$$

so

$$3 = 228u + 63v$$

where u, v are integers. Divide both sides by 3 we have

$$1 = 76u + 21v$$

so

$$76u \equiv 1 \pmod{21}$$

To calculate u, notice that

$$21u \equiv 0 \pmod{21}$$

substract this we have

$$76u - 21u \times 3 \equiv 1 - 0 \pmod{21}$$

$$13u \equiv 1 \pmod{21}$$

$$21u - 13u \times 2 \equiv 0 - 2 \pmod{21}$$

so

$$-5u \equiv -2 \pmod{21}$$

$$21u - 5u \times 4 \equiv 0 - 8 \pmod{21}$$

$$u \equiv -8 \pmod{21}$$

$$u \equiv 13 \pmod{21}$$

take $u = 13$ and we can calculate that $v = -47$

$$3 = 228 \times 13 + 63 \times (-47)$$

4. 6.10 Multiplicative inverse:pseudocode

   We want to compute $ux \equiv 1 \pmod{m}$

   Input: let $a = max(u, m)$,$b = min(u, m)$. If $u > m$ then $c = 1, d = 0$,otherwise $c = 0, d = 1$. Use euclid's algorithm to test if $gcd(a, b) = 1$. (note if $u = m$, $gcd(a, b) = m = u$.)

   Output: multiplicative inverse of u mod m.

   Pseudocode

   Test existence of multiplicative inverse
   0 **if** $gcd(a, b) \neq 1$
   1   **return** $FALSE$

   Compute multiplicative inverse
   2 Initialize: $A := a$, $B := b$, $C := c$, $D := d$, $T = C$, $R = 0$, $q = 0$
   3 **while** $B \geq 1$ **do**
   4     $R := (A \bmod B)$
   5     $q := (A - R)/B$
   6     $C := D, D := T - qD, T := C$
   7     $A := B, B := R$
   8 **end** (**while**)
   9 **return** D

   The running time of this algorithm is constant times the running time of euclids algorithm. Line 0 and 1 are added to test the exisitence of multiplicative inverse. Line 5 and 6 are added to compute the multiplicative inverse.

5. 6.12 application of Fermat's little Theorem
   From Fermat's little Theorem, since $gcd(7, 101) = 1$. We have

$$7^{101-1} \equiv 1 \pmod{101}$$

   so

$$(7^{10^2})^{10^7} \equiv 1^{10^7} \pmod{101}$$
$$7^{10^9} \equiv 1 \pmod{101}$$

   so

$$7^{10^9} \bmod 101 = 1$$

6. 6.14
   Claim: Yes, Given N, K, p, q can be calculated in polynomial time.

   We know

$$N = pq$$
$$K = (p-1)(q-1)$$

   so

$$p + q = N - k + 1$$
$$pq = N$$

   is a system of 2 equations with 2 unknows. We can solve this with formula.
   let $x_1 = p$,$x_2 = q$, we can write

$$x^2 - (p+q)x + pq = 0$$

   so

$$x^2 - (N - K + 1)x + N = 0$$

   so

$$p, q = x_1, x_2 = \frac{N - K + 1 \pm \sqrt{(N-k+1)^2 - 4N}}{2}$$

7. 6.16 Frank's algorithm time complexity
   let b's bit length be n, so $b < 2^n$. The algorithm terminates at $b^{\frac{1}{2}}$. Plug in $b = 2^n$.

$$T(n) < 2^{\frac{n}{2}}$$
$$T(n) = O(2^{\frac{n}{2}})$$

   The algorithm finishes in exponential time.

8. 6.17 Ashwin and Ming secure scheme break

   Ashwin picks primes p, q and Ming picks primes p, r. Since $n = pq$ and $m = pr$ are known from the public keys of Ashwin and Ming. We can use Euclid's algorithm to find p which is $gcd(n, m)$ in polynomial time.

Extra credit

reference to http://www.cut-the-knot.org/arithmetic/GcdLcmProperties.shtml wan to show that

$$lcm(gcd(n_1, m), gcd(n_2, m), ..., gcd(n_k, m) = gcd(lcm(n_1, n_2, ..., n_k), m)$$

Proof: let p be a prime divisor of $lcm(gcd(n_1, m), ...gcd(n_k, m))$. let a be the largest exponent such that $p^a|lcm(gcd(n_1, m), ..., gcd(n_k, m)$. so p^a divides at least one of $gcd(n_i, m)i = 1, 2, 3, .., k$. let this number be $gcd(n_1, m)$. So p^a is common divisor of n_1 and m.since $p^a|m$, it also divides $lcm(n_1, n_2, ..., n_k)$. So p^a is the common factor of $lcm(n_1, ..., n_k)$, so $p^a|gcd(lcm(n_1, ...n_k), M)$.