CS60094: Computational Number Theory, Spring 2018-19
Class Test 1

CSE 107, 8AM — 15th February, 2018
Duration = 55 minutes — Total marks = 20

1. Let $a, b \in \mathbb{N}$ with $d = \gcd(a, b) = ua + vb$ for some $u, v \in \mathbb{Z}$.

   (a) Demonstrate that u, v are not unique. **[2]**

   Solution: $5 = \gcd(20, 15)$. We can write 5 as

   $$1 \times 20 + (-1) \times 15$$

   or

   $$(-2) \times 20 + (3) \times 15.$$

   This shows the multipliers are not unique.

   (b) Assume that $(a, b) \neq (1, 1)$. Prove that u, v can be chosen to satisfy $|u| \leq b/d$ and $|v| \leq a/d$. **[6]**

   Solution: Suppose that $d = \gcd(a, b) = a$. Then $a = 1 \times a + 0 \times b$ and since $1 \leq b/a$ and $0 \leq a/a$, the statement holds for this case. Similar argument holds for the case $d = b$. Now assume $d = \gcd(a, b) \neq \min(a, b)$. Let $x, y \in \mathbb{Z}$ such that $d = xa + yb$. Then

   $$\left(x - k \cdot \frac{b}{d}\right) a + \left(y + k \cdot \frac{a}{d}\right) b = xa + yb = d$$

   for all $k \in \mathbb{Z}$. Since $x, b/d \in \mathbb{Z}$, there exist integers q, r such that $x = q(b/d) + r$ with $0 \leq r < |b/d|$. Set $k = q$ and hence $u = x - k(b/d)$, $v = y + k(a/d)$. Clearly $u < |b/d|$ as $u = r$. Since $b, d \in \mathbb{N}$, $|u| < b/d$. Now, $v = y + q(a/d) = (d - u)/b$. Using the fact that $-b/d < u < b/d$, we have

   $$\frac{d}{b} - \frac{a}{d} < v < \frac{d}{b} + \frac{a}{d}$$

   Since $d/b < 1$ (assuming $d \neq \min(a, b)$) and $v \in \mathbb{Z}$, we have $|v| \leq a/d$.

2. Suppose that we want to compute $x^r y^s \pmod m$, where r and s are positive integers of the same bit size. By the repeated square-and-multiply algorithm, one can compute $x^r \pmod m$ and $y^s \pmod m$ independently, and then multiply these two values. Alternatively, one may rewrite the square-and-multiply algorithm using *only one loop* in which the bits of both the exponents r and s are simultaneously considered. After each square operation, one multiplies by 1, x, y or xy.

   Elaborate the algorithm outlined above. What speedup is this modification expected to produce? **[4]**

   Solution: Let r, s be k-bit long. Below is the algorithm named DoubleExp that takes as input x, y, r, s, m and returns $x^r y^s \pmod m$.

   DoubleExp (x, y, r, s, m)

   let $r = (r_{k-1}, r_{k-2}, \ldots, r_1, r_0)_2, s = (s_{k-1}, s_{k-2}, \ldots, s_1, s_0)_2$
   $w \leftarrow 1 \pmod m$
   for $i \leftarrow k - 1$ down to 0, do

$w \leftarrow w^2 \pmod m$
if $r_i = 1$ and $s_i = 1$, set $w \leftarrow wxy \pmod m$
if $r_i = 1$ and $s_i = 0$, set $w \leftarrow wx \pmod m$
if $r_i = 0$ and $s_i = 1$, set $w \leftarrow wy \pmod m$
return w

Computing $x^r \pmod m$ and $y^s \pmod m$ independently would require $\log_2 r + \log_2 s = 2k$ modular squarings and atmost $2k$ modular multiplications. Computing $x^r y^s \pmod m$ would require an additional modular multiplication. On the other hand, DoubleExp computes $x^r y^s \pmod m$ using k modular squarings and atmost k modular multiplications. Note that $xy \pmod m$ can be precomputed. This indicates that DoubleExp achieves a factor 2 speed-up over the naive method.

3. (a) Let m_1, m_2 be coprime moduli and let $a_1, a_2 \in \mathbb{Z}$. By the extended gcd algorithm, one can compute $u, v \in \mathbb{Z}$ such that $um_1 + vm_2 = 1$. Prove that $x = um_1a_2 + vm_2a_1 \pmod{m_1m_2}$ is the simultaneous solution of the congruences $x \equiv a_i \pmod{m_i}$ for $i = 1, 2$. **[4]**

Solution: Given that $um_1 + vm_2 = 1$, we have

$$um_1a_2 + vm_2a_1 \equiv um_1a_2 \pmod{m_2} \equiv (1 - vm_2)a_2 \pmod{m_2} \equiv a_2 \pmod{m_2}.$$

Similarly, we can show that $um_1a_2 + vm_2a_1 \equiv a_1 \pmod{m_1}$. Hence $um_1a_2 + vm_2a_1 \pmod{m_1m_2}$ is a solution to the congruences $x \equiv a_i \pmod{m_i}$, $i = 1, 2$.

(b) Let $m_1, m_2, \ldots, m_t$ be pairwise coprime moduli, and $a_1, a_2, \ldots, a_t \in \mathbb{Z}$. Write an incremental procedure for the Chinese remainder theorem that starts with the solution $x \equiv a_1 \pmod{m_1}$ and then runs a loop (for $i = 2, 3, \ldots, t$ in that order), the i-th iteration of which computes the simultaneous solution of $x \equiv a_j \pmod{m_j}$ for $j = 1, 2, \ldots, i$. **[4]**

Solution: Let ExtendedEuclid denote the algorithm that on input a, b returns d, u, v such that $d = \gcd(a, b)$ and $d = ua + vb$. When the inputs are coprime, the first output (i.e., d) is always 1. We now describe an algorithm CRT-V that on input $m[1, \ldots, t]$ and $a[1, \ldots, t]$ with $m[i]$'s being pairwise coprime, outputs a solution $x \pmod M$ with $M = \prod_{i=1}^{t} m[i]$ simultaneously satisfying the congruences $x \equiv a[i] \pmod{m[i]}$ for $i = 1, \ldots, t$.

CRT-V $(m[1, 2, \ldots, t], a[1, 2, \ldots, t])$

$y \leftarrow a[1] \pmod{m[1]}$
$M \leftarrow m[1]$
for $i \leftarrow 2$ to t, do
  $(d, u, v) \leftarrow \mathsf{ExtendedEuclid}(M, m[i])$
  $y \leftarrow u \cdot M \cdot a[i] + v \cdot m[i] \cdot y \pmod{M \cdot m[i]}$
  $M \leftarrow M \cdot m[i]$
return y

For convenience we will denote $m[i], a[i]$ as m_i, a_i in what follows.

Claim. *The value of y after k iterations is a simultaneous solution (modulo $\prod_{i=1}^{k} m_i$) to the congruences $x \equiv a_i \pmod{m_i}$ for $i = 1, \ldots, k$.*

Proof. We will prove the claim by induction on k. For $k = 1$, $a[1]$ itself is a solution to $x \equiv a_1 \pmod{m_1}$. Since y is initialised to a_1, the claim is true for $k = 1$. When $k = 2$, the algorithm returns $y = um_1a_2 + v \cdot m_2 \cdot a_1 \pmod{m_1 \cdot m_2}$ where u, v are output by $\mathsf{ExtendedEuclid}(m_1, m_2)$. From part(a), we know that y is a solution and hence the claim holds for $k = 2$. Suppose that after k iterations, y simultaneously satisfies $x \equiv a_i \pmod{m_i}$ for $i = 1, \ldots, k$. At this point $M = \prod_{i=1}^{k} m_i$. In the $(k+1)$-st iteration, u, v returned by $\mathsf{ExtendedEuclid}(M, m_{k+1})$ satisfy $uM + vm_{k+1} = 1$.

Also, y is updated to $uMa_{k+1} + vm_{k+1}y \pmod{Mm_{k+1}}$. We now show that this updated value satisfies the congruences $x \equiv a_i \pmod{m_i}$ for $i = 1, \ldots, k+1$. For $1 \leq i \leq k$, $m_i | M$ and we have

$$\begin{aligned} uMa_{k+1} + vm_{k+1}y \pmod{m_i} &\equiv vm_{k+1}y \pmod{m_i} \\ &\equiv (1 - uM)y \pmod{m_i} \\ &\equiv y \pmod{m_i} \\ &\equiv a_i \pmod{m_i}. \end{aligned}$$

So the updated y still satisfies the congruences $x \equiv a_i \pmod{m_i}$ for $i = 1, \ldots, k$. Also, we have

$$\begin{aligned} uMa_{k+1} + vm_{k+1}y \pmod{m_{k+1}} &\equiv uMa_{k+1} \pmod{m_{k+1}} \\ &\equiv (1 - vm_{k+1})a_{k+1} \pmod{m_{k+1}} \\ &\equiv a_{k+1} \pmod{m_{k+1}} \end{aligned}$$

thus showing that the updated y is a solution to $x \equiv a_{k+1} \pmod{m_{k+1}}$. This concludes the proof of the claim. □