Algorithms Exercises Answers and Notes

exceedhl@gmail.com

May 7, 2012

1 Chapter 1

1.1 Basic facts about rem

Theorem 1 *Rem operation has the following features: $(a, b \in \mathbb{Z})$*

$$
\begin{aligned}
ab\ rem\ N &= a(b\ rem\ N)\ rem\ N && (1)\\
&= (a\ rem\ N)b\ rem\ N && (2)\\
&= (a\ rem\ N)(b\ rem\ N)\ rem\ N && (3)\\
a^b\ rem\ N &= (a\ rem\ N)^b && (4)\\
(a \pm b)\ rem\ N &= ((a\ rem\ N) \pm (b\ rem\ N))\ rem\ N && (5)
\end{aligned}
$$

These features are important for solving $a^b\ rem\ N$ problems.

Theorem 2 *Suppose $a \equiv b \mod n$ and $c \equiv d \mod n$. Then*

1. $a + c \equiv b + d \mod n$
2. $ac \equiv bd \mod n$
3. $f(a) \equiv f(b) \mod n$ *for any polynomial f(x) with integer coefficients.*

Theorem 3 *About mod operation:*

$$
\begin{aligned}
&a \equiv b \mod N, b \equiv c \mod N && (6)\\
\Rightarrow &a \equiv b \equiv c \mod N && (7)
\end{aligned}
$$

If p is a prime, a is not a multiplicative of p, then

$$
\begin{aligned}
&a^{p-1} \equiv 1 \mod p && (8)\\
\Rightarrow &a^n\ rem\ p = a^{n\ rem\ (p-1)}\ rem\ p && (9)\\
&a^m \equiv 1 \mod p && (10)\\
\Rightarrow &a^n\ rem\ p = a^{n\ rem\ m}\ rem\ p && (11)
\end{aligned}
$$

(9) is the key to solve a^{b^c} *rem* p problems.

Theorem 4 *If a has a multiplicative inverse modulo N , then this inverse is unique (modulo N).*

Theorem 5 *If a has an inverse modulo b, then b has an inverse modulo a.*

1.2 Theorems

Theorem 6 *If p, q are primes, and a is not a multiplier of either of them then we have:*

$$a^{(p-1)(q-1)} \equiv 1 \ mod \ pq \tag{12}$$

This is useful in the proof of RSA algorithm.

Theorem 7 *If p is a prime, a is an integer, then $GCD(a, p^n) \neq 1$ ($n > 0$ is a integer) if and only if $a = kp$ (k is an integer).*

Theorem 8 *If p is a prime and $a < p$, then there exists $b < p$, so that $ab \equiv 1 \ mod \ p$.*

Proof. According to Fermat's theorem: there exists a a^{-1} so that $aa^{-1} \equiv 1 \ mod \ p$.

Because $aa^{-1} \ rem \ p = a(a^{-1} \ rem \ p) \ rem \ p$ so ethat there exists a $b < p$ so that $ab \equiv 1 \ mod \ p$.

This theorem can be used to prove Wilson's theorem:

$$(p-1)! \equiv -1 \mod p \tag{13}$$

Theorem 9 *For any two adjacent Fibonacci numbers F_n and F_{n+1}, $GCD(F_n, F_{n+1}) = 1$.*

1.3 RSA algorithm

p, q are primes, $N = pq$, $N' = (p-1)(q-1)$, e is relatively prime to N'. x is the plain text and $x \in 0, 1, \ldots, N-1$.

N and e are public, the encrypted text is:

$$x' = x^e \ rem \ N, \quad x' \in 0, 1, \ldots, N-1 \tag{14}$$

Let

$$de \equiv 1 \ mod \ N', \tag{15}$$

d can be calculated using extended Euclid algorithm.

Then

$$
\begin{align}
de - 1 &= kN' \tag{16}\\
de &= kN' + 1 \tag{17}\\
x'^d - x &= x^{ed} - x = x^{1+k(p-1)(q-1)} - x \tag{18}
\end{align}
$$

Because p, q are primes and $x < N$, so according to Fermat's little theorem:

$$
\begin{align}
&x^{p-1} \equiv 1\ mod\ p \tag{19}\\
\Rightarrow &x^{(p-1)k(q-1)} \equiv 1\ mod\ p \tag{20}\\
&x^{q-1} \equiv 1\ mod\ q \tag{21}\\
\Rightarrow &x^{(q-1)k(p-1)} \equiv 1\ mod\ q \tag{22}\\
\Rightarrow &x^{(q-1)k(p-1)} - 1\ rem\ pq = 0 \tag{23}\\
\Rightarrow &x^{(q-1)k(p-1)} \equiv 1\ mod\ pq \tag{24}\\
\Rightarrow &x'^d - x = x^{ed} - x = x^{1+k(p-1)(q-1)} - x\ rem\ pq = 0 \tag{25}\\
\Rightarrow &x'^d \equiv 1\ mod\ N \tag{26}
\end{align}
$$

Because $x^{ed} \equiv x\ mod\ N$ and $x < N$, we can get x by calculating $x'^d\ rem\ N$.

So if we make N, e public, keep d, p, q secret, then we can encrypt and decrypt messages using power and rem operations.

Finally, since $x^e\ rem\ N = x'$, and $x'^d\ rem\ N = x$, it is obvious that they are bijection functions.

1.4 Exercises

1.4.1 e1.1

Consider the biggest possible value of adding three digits in base b: $3(b-1)$. The biggest value of two digits in base b is: $b^2 - 1$. Now we just need to prove $b^2 - 1 \geq 3(b-1)$.

$$
\begin{align}
b^2 - 1 - 3(b-1) &= b^2 - 3b + 2 \tag{27}\\
&= (b-1)(b-2) \tag{28}\\
&\geq 0 \quad (\forall\ b \geq 2) \tag{29}
\end{align}
$$

1.4.2 e1.26

According to Feramt's theorem, if p is a prime and k is not a multiplier of p, then $k^{p-1} \equiv 1\ mod\ p$.

If there are two primes and k is not multiplier of either one, then we can have (refer to the proof of RSA algorithm):

$$k^{(p-1)(q-1)} \equiv 1\ mod\ pq \tag{30}$$

Let the least significant digit of $17^{17^{17}}$ is d:

$$17^{17^{17}} \equiv d\ mod\ 10 \tag{31}$$

$10 = 2 * 5$ so set $p = 2$, $q = 5$, we just need to find a which is not multiplier of 2 and 5, then we have $a^4 \equiv 1\ mod\ 10$.

Continuously use this a^4 to divide the original number to get d.

Let $a = 17$, so

$$\begin{align} 17^{17^{17}} &\equiv 17^{289} \tag{32}\\ &\equiv 17 * 17^{288} \tag{33}\\ &\equiv 17 * (17^4)^{72} \tag{34}\\ &\equiv 17 \tag{35}\\ &\equiv d\ mod\ 10 \tag{36}\\ &\Rightarrow\ d = 7 \tag{37} \end{align}$$

1.4.3 e1.29

The problem assumes $x_1 < m,\ x_2 < m$.

(a) is universal hashing function, the proof is same with the IP hashing function example.

(b) is not universal. Suppose (x_1, x_2) is different with (y_1, y_2) and x_2 is different with y_2. $h_{a_1,a_2}(x_1, x_2)$ equals with $h_{a_1,a_2}(y_1, y_2)$ means:

$$\begin{align} a_1x_1 + a_2x_2 &\equiv a_1y_1 + a_2y_2\ mod\ m \tag{38}\\ a_1(x_1 - y_1) &\equiv a_2(y_1 - y_2)\ mod\ m \tag{39} \end{align}$$

Suppose the left side equals to c, then if $(y_1 - y_2)$ is relative prime to m, then a_2 must be $c(y_1 - y_2)^{-1}$.

Because $m = 2^k$ is not prime and the number of numbers that are relative prime to m is $\phi(m) = \phi(2^k) = 2^k - 2^{k-1}$.

The chance of $(y_1 - y_2)$ being relative prime to $m = 2^k$ is $1/2$ because any odd number is relative prime to 2^k, so the chance of (39) holding is: $1/2 * 1/(2^k - 2^{k-1}) = 1/2^k$.

When $(y_1 - y_2)$ is even there exists some a_2 to make (39) hold. For example:

$$m = 2^3 = 8 \tag{40}$$

$$y_1 - y_2 = 2 \tag{41}$$

$$c = 2 \tag{42}$$

$$2 \equiv a_2 * 2 \ mod \ 8 \tag{43}$$

$$\Rightarrow a_2 = 5 \tag{44}$$

So the overall probability of making (39) hold is greater than $1/2^k = 1/m$.

(c) is not universal. Take an arbitrary f, the probability of a number p's key being conflict with another's key is $1/(m-1) > 1/m$.

1.4.4 1.32 Perfect square/power check