Sept29

Monday, September 28, 2020 9:57 PM

Corrections for HW4

① $\phi : \mathbb{R}^* \longrightarrow GL_2(\mathbb{R})$

$r \longmapsto \cancel{\begin{pmatrix} r & 0 \\ 0 & -r \end{pmatrix}} \begin{pmatrix} r & 0 \\ 0 & r^{-1} \end{pmatrix}$

② $G[n] = \{x \in G \mid x^n = \underset{e}{\cancel{1}}\}$

Announcement

Monday Office Hours 1:30-2:30

Defn Let G be a group, $g, h \in G$.
The discrete log problem for G is the problem of finding a solution to
$$g^x = h.$$

Attacking the DLP

Prop If $g \in G$ has order N. The we can solve the DLP in $O(N)$-time w/ $O(1)$ memory. ← storage.

Proof
$g, g^2, g^3, \ldots$ terminates in N multiplications.
$g_i = g^i$
Find $g^{i+1} = g_i \cdot g$
At any time need only store $[g, g_i]$
$(2 = O(1))$

Rmk $G = \mathbb{F}_p^*$. $p \sim 2^k$.
This is $O(2^k)$ expontal time.

The underlying group matters!!!

Example

$G = \mathbb{F}_p$ with $+$

DLP is solving

$$\text{"}g^x\text{"} = \underbrace{g+g+\cdots+g}_{x\text{-times}} \equiv h \mod p$$

$$xg \equiv h \mod p.$$

multiply both sides by g^{-1}

$$x \equiv g^{-1}h \mod p \quad \swarrow$$

Finding g^{-1} takes $\leq 2\log_2 p + 2 = O(\log_2 p)$

(if $p = 2^k$, $\leq 2k+2 = O(k)$

$\uparrow$ linear time.

DLP for $G \ni g$ w/ order N takes $O(N)$ $O(1)$

Present an algorithm that takes $O(\sqrt{N}\log N)$ $\uparrow$ storage

$O(\sqrt{N}) \leftarrow$ storage

Prop (Babysteps & Giant Steps Algorithm)

G a group $g \in G$ $|g| = N \geq 2$. The following algorithm solves $g^x = h$ in $O(\sqrt{N}\log N)$ steps w/ $O(\sqrt{N})$ storage.

① Compute $n = \lfloor \sqrt{N} \rfloor + 1$ $\quad (n > \sqrt{N})$

② Create two lists

i) $e, g^1, g^2, \dots, g^n$ $\leftarrow$ baby steps

ii) $h, hg^{-n}, h \cdot g^{-2n}, \dots, h \cdot g^{-n^2}$ $\leftarrow$ giant steps

→ ③ Find a match btwn 2 lists.

$$\rightarrow g^i = h \cdot g^{-nj}$$

ther is x!

④ $h = g^i g^{nj} = g^{\boxed{i+nj}}$

Proof

① Correctnes.

suffice to show lists overlap.

Suppose $g^x = h$ in G

Long divide x by n.

$$x = nq + r \qquad 0 \le r < n$$

Solve for q ($|g| = N$ know $x < N$)

$$q = \frac{x-r}{n} < \frac{N}{n} < \frac{N}{\sqrt{N}} = \sqrt{N} < n$$

$$g^x = h \rightsquigarrow g^{nq+r} = h$$

$$\rightsquigarrow \underbrace{g^r}_{\text{List i}} = \underbrace{h \cdot g^{-nq}}_{\text{list ii}}$$

② Time

list i) Takes n multiplications

list ii) First compute g^{-n} 1 inversion

Do n multiplication

Making lists 2n multiplications ← $O(n)$

1 inversion. ← $O(\log N)$

(Sorting & finding a match between two lists of length n. $O(n \log n)$) standard

Total $O(\sqrt{N} + \log N + \underline{\sqrt{N} \log \sqrt{N}}) = O(\sqrt{N} \log \sqrt{N})$

$$\left(\lim_{n \to \infty} \frac{\sqrt{N} + \log N + \sqrt{N} \log \sqrt{N}}{\sqrt{N} \log \sqrt{N}} = 1 \right)$$

③ Storage

Remember 2 lists lenght $n+1$

$2n+2 = O(n) = O(\sqrt{N})$ $\cancel{O}$

Remark $O(\sqrt{N} \log \sqrt{N})$

$$= O(\sqrt{N} \log N^{1/2}) = O(\tfrac{1}{2}\sqrt{N} \log N)$$
$$= O(\sqrt{N} \log N)$$

② $G = \mathbb{F}_p$ $p \sim 2^k \sim N$ In terms of k-bits

Then $O(\sqrt{N} \log N)$

$$= O(\sqrt{2^k} \log 2^k) =$$
$$= O(k 2^{k/2})$$ ← Still exponential time in k

HW Implement this attack.

Chinese Remainder Theorem (Sun Tzu ~300AD)

Goal Solve systems of congruences across varying moduli

For example Special case of CRT

$$x \equiv a \mod m$$
$$x \equiv b \mod n$$

If $\gcd(m,n)=1$, there is a solution.
Furthermore this solution is unique mod mn.

Example

① $x \equiv 2 \mod 3$
② $x \equiv 3 \mod 7$
③ $x \equiv 4 \mod 16$
*

① 2, 5, 8, 11, 14, ⑰, 20, 23, ---
$2+3y = x$
② 3, 10, ⑰, 24, ----

sub into equation #2.

$2+3y = 3 \mod 7$ ← solve for y

$3y \equiv 1 \mod 7$ ← $\gcd(3,7)=1$ so $3^{-1} \equiv 5 \mod 7$

$\times 5$ $\times 5$

$y \equiv 5 \mod 7$ ← plug into $x = 2+3y$

$x = 17$ ← $17 \equiv 2 \mod 3$, $17 \equiv 3 \mod 7$ ✓

* 17 doesn't solve ③

Observation $\boxed{x' \equiv 17 \mod 21.}$ ⟹

⟹ ① $x \equiv 2 \mod 3$
② $x \equiv 3 \mod 7$

$x' = 17 + 21z = 17 + 3 \cdot 7 \cdot z$
$\equiv 17 \mod 3$
$\equiv 2$
$\equiv 17 \mod 7$
$\equiv 3$

Combine ① & ② into
①' $x \equiv 17 \mod 21$ ← $x = 17 + 21z$
③ $x \equiv 4 \mod 16$

$17 + 21z \equiv 4 \mod 16$

$5z \equiv 3 \mod 16$ ← $\gcd(21,16)=1$
$\qquad \| $
$\gcd(5,16)$

$z = 3 \cdot 13 \mod 16$ $\qquad 5^{-1} = 13 \mod 16$

$$\equiv 39 \bmod 16$$
$$\equiv 7 \bmod 16.$$

$$x = 17 + 21\cdot 7 = 164$$

Sanity Check

$$\left.\begin{aligned} 164 &= 2+162 \cong 2 \bmod 3 \\ 164 &= 3+161 \cong 3 \bmod 7 \\ 164 &= 4+160 \cong 4 \bmod 16 \end{aligned}\right\} \text{It Works!!}$$

Uniqueness

$$\left|\begin{aligned} x &\equiv 2 \bmod 3 \\ x &\equiv 3 \bmod 7 \end{aligned}\right. \rightsquigarrow 17.$$

If $x' \equiv 17 \bmod 3\cdot 7$

Then x' works too.

Check Suppose x' satisfies 1 & 2.

$$\Rightarrow 3 \mid (x-x') \quad \& \quad 7 \mid (x-x')$$
$$\Rightarrow 21 \mid (x-x')$$
$$\Rightarrow x \equiv x' \bmod 21$$

$$\left(\begin{aligned} &\text{Solutions to} \\ &① \; x \equiv 2 \bmod 3 \\ &② \; x \equiv 3 \bmod 7 \end{aligned}\right) \longleftrightarrow \left(\begin{aligned} &\text{Solutions to} \\ &x \equiv 17 \bmod 21 \end{aligned}\right)$$

Theorem (Sun Tzu (Chinese Remainder Theorem))

Let $m_1, m_2, m_3, \ldots m_k$ a collection of coprime integers (i.e. $\gcd(m_i, m_j) = 1$ $\forall i \neq j$)

Let $a_1, \ldots, a_k \in \mathbb{Z}$. Then the system

$$\left.\begin{aligned} x &\equiv a_1 \bmod m_1 \\ x &\equiv a_2 \bmod m_2 \\ &\vdots \\ x &\equiv a_k \bmod m_k \end{aligned}\right\}$$

Has a solution.

Furthermore, if x, x' both solve the system

$x \equiv x' \mod m_1 m_2 \cdots m_k$

Proof Existence of a solution

induction

Base Case:

$c_1 = a_1$ solves $x \equiv a_1 \mod m_1$ ✓

Inductive Step:

Assume c_i solves

$x \equiv a_1 \mod m_1$
$x \equiv a_2 \mod m_2$
$\vdots$
$x \equiv a_i \mod m_i$

use this to produce a (possibly different) solution the first $i+1$ congruences.

Notice Any $x = \underline{c_i + (m_1 m_2 \cdots m_i) y}$ solves first i congruences. plug this into

$x \equiv a_{i+1} \mod m_{i+1}$

$c_i + (m_1 m_2 \cdots m_i) y \equiv a_{i+1} \mod m_{i+1}$

$(m_1 \cdots m_i) y \equiv a_{i+1} - c_i \mod m_{i+1}$

$\gcd(m_1 \cdots m_i, m_{i+1}) = 1$
$(m_1 \cdots m_i)^{-1} \mod m_{i+1}$ exists

plug $\boxed{y} = \frac{a_{i+1} - c_i}{m_1 m_2 \cdots m_i} \mod m_{i+1}$

↳ Have x which solves first $i+1$ congruences.

⊗

Uniqueness: HW.

HW: Implement this algorithm.

Applications: Solving a congruence mod composite number

Example $m = pq$

$$\left(\text{solve a congruence mod } m\right) \xleftrightarrow{\text{CRT}} \left(\text{solve a congruence mod } p \text{ and another mod } q\right)$$

$$m = p_1^{\alpha_1} \cdots p_k^{\alpha_1}$$

$$\left(\text{Solve congruence mod } m\right) \xleftrightarrow{\text{CRT}} \left(\text{Solve a system mod } p_i^{\alpha_i} \text{ for } i = 1, \ldots, k\right)$$

Use this to attack problem of finding square roots modulo m.

Question: $m = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$

Does $x^2 = a$ mod m have a root?

$\Updownarrow$

Does $x^2 = a \bmod p_i^{\alpha_i}$ have a root *

Pf/ $b_i^2 = a \bmod p_i^{\alpha_i}$

$\rightsquigarrow$ $x \equiv b_i \bmod p_i^{\alpha_i}$ $\forall$ i has a solution.

$x^2 \equiv b_i^2 \equiv a \bmod p_i^{\alpha_i}$ $\forall i$

$\xRightarrow[\text{CRT}]{}$ $x^2 \equiv a \bmod m$. $\emptyset$

*HW3 #7

$x^2 \equiv a \bmod p_i^{\alpha_i}$ has soln $\iff$ $x^2 \equiv a \bmod p_i$ does.

Prop $m = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$.

Then $x^2 = a \bmod m \iff x^2 = a \bmod p$ all p.

Finding square roots becomes a lot easier assuming we can factor m. (***)

Question ① How can we figure out if a is a square mod p?

② How can we use this to find it explicitly.

③ How can we lift this to a square root mod p^k?

Answer questions 1 & 2 In the case that $p \equiv 3 \mod 4$.

Prop Suppose $p \equiv 3 \mod 4$. a prime to p. Suppose a has a square root mod p. Then $b = a^{p+1/4}$ is a square root mod p.

Remark

Gives us an algorithm to check if a is a square.

① Compute $b = ^{p+1/4} \mod p$

② Square b & see if it returns a.

Proof $g \in \mathbb{F}_p^*$ a primitive root. So $g^\ell = a$.

HW showed a a square $\Longleftrightarrow$ ℓ even.

So $\underline{g^{2k} = a}$

$b = a^{\frac{p+1}{4}} \mod p$. Let's square it!!

$$b^2 \equiv a^{\frac{p+1}{2}} \mod p$$

$$\equiv (g^{2k})^{\frac{p+1}{2}} \mod p$$

$$\equiv g^{(p+1)k} \mod p$$

$$\equiv g^{(p-1)k+2k} \mod p$$

$$\equiv \underbrace{g^{(p-1)k}}_{1 \text{ by FLT}} \underbrace{g^{2k}}_{a \text{ by defn}} \mod p$$

$$\equiv a \mod p$$

$(p+1)k$

$= (p-1+2)k$

$= (p-1)k + 2k$