Lec19A Cantor-Zassenhaus Details

March 23, 2021 1:50 PM

Let p be a prime and $a, w \in \mathbb{Z}_p[x]$ with $\deg w < \deg a = d$.

How do we compute $w^p \bmod a \equiv \text{rem}(w^p \div a)$ for large p e.g. $p = 2^{31}-1$.

$$r \leftarrow w; \text{ for } i \text{ to } p-1 \text{ do } r \leftarrow \underbrace{r}_{\le d-1} \cdot \underbrace{w}_{\le d-1} \bmod \underbrace{a}_{d}; \text{ od};$$

$\deg(r \cdot w) \le 2d-2$ $\quad O((d-1)^2) = O(d^2)$

$O((\underbrace{2d-2-d+1}_{d-1}) \cdot d) = O(d^2)$.

Cost $(p-1) \times$ and $\div$

$= (p-1)(O(d^2) + O(d^2))$

$= O(pd^2)$.

Use Binary Powering with remainder [Square & Multiply].

$p = 101 = 64 + 32 + 4 + 1 = 1100101$ in binary.

$w^{101} \bmod a = w^{64} \cdot (w^{32} \cdot (w^4 \cdot w^1)) \bmod a$.

This does $(3+6) \times$
$+ (3+6) \div$
$= 9 \times + 9 \div$
in comparison with
$100 \times + 100 \div$

	$S \leftarrow w$
$w^2 \bmod a$	$S \leftarrow S^2 \bmod a$
$w^4 \bmod a$	$S \leftarrow S^2 \bmod a$.
$w^8 \bmod a$	$S \leftarrow S^2 \bmod a$
$w^{16} \bmod a$	$S \leftarrow S^2 \bmod a$
$w^{32} \bmod a$	$S \leftarrow S^2 \bmod a$
$w^{64} \bmod a$	$S \leftarrow S^2 \bmod a$

Algorithm PowMod (w, n, a)
Input $w, a \in \mathbb{Z}_p[x]$, $n \ge 0$, $\deg(w) < \deg(a) = d \ge 1$.
Output $w^n \bmod a$.

$S \leftarrow w$
$r \leftarrow 1$
while $n > 0$ do
  if n is odd then $r \leftarrow \underbrace{r}_{\le d-1} \cdot \underbrace{S}_{\le d-1} \bmod \underbrace{a}_{d}$ fi

$\le d-1$ $\quad d$ $\quad O(d^2)$ $\quad O(d^2)$

while ...

if n is odd then $r \leftarrow \underline{r \cdot s}$ mod a fi

$s \leftarrow s^2$ mod a $\qquad O(d^2)\ \ O(d^2)$.

$n \leftarrow \lfloor n/2 \rfloor$

od;

return r $\qquad$ # iterations is $\lfloor \log_2 n \rfloor + 1$

end.

Cost $\leqslant (\lfloor \log_2 n \rfloor + 1)(2O(d^2) + 2O(d^2)) = O(d^2 \log n)$.

Maple Powmod(w, n, a, x) mod p;

w^n mod a

Gcd($v^{(p^k-1)/2} \pm 1, a)$

Powmod$(v, (p^k-1)/2, a, x)$ mod p.

A probalistic algorithm for computing the roots of $a \in \mathbb{Z}_p[x]$.
Assume $d = \deg a > 1$ and $\gcd(a, a') = 1$ and $a(0) \neq 0$.

FLT $\quad x^p - x = (x-\underline{0})(x-\underline{1})(x-\underline{2}) \cdots (x-(\underline{p-1}))$.

Step ① $\quad g = \gcd(\underline{a}, \underline{x^p - x}) =$ all linear factors of a.

$= \gcd(\underline{a}, \underline{(x^p \bmod a) - x}) \quad — O(d^2 (\log p))$.

$O(d^2)$ $\qquad$ Powmod. $O(d^2 \log p)$

even

$p \neq 2 \quad x^p - x = x(x^{p-1} - 1) = x(x^{\frac{p-1}{2}} - 1)(x^{\frac{p-1}{2}} + 1)$

half linears $\qquad$ other half

② Randomize: $\qquad h = \gcd(\underline{(x+\alpha)}_{w}^{\frac{p-1}{2}} - 1, g)$ where α is chosen at random from $\mathbb{Z}_p$.

$O(d^2)$. $\qquad$ Powmod. $O(d^2 \log p)$.

If $\deg(g) \gg 1$ this will split g into two factors
h and $\frac{g}{h}$ of degree $\frac{d}{2} \pm \epsilon$.

If $\deg(g) \gg 1$ this will split g into

h and $\frac{g}{h}$ of degree $\frac{d}{2} \pm \epsilon$.

Algorithm Split(g)

Input $g \in \mathbb{Z}_p[x]$ a product of linear factors in $\mathbb{Z}_p[x]$.

if $\deg g) = 0$ then return ϕ

if $\deg(g) = 1$ then return $\{g\}$ $\quad T(1) = 0$

$O(d^2 \log p)$.

$\underline{h} \leftarrow \gcd\left((x+\alpha)^{\frac{p-1}{2}} - 1, g \right)$ for some random $\alpha \in \mathbb{Z}_p$

return Split$(h) \cup$ Split(g/h). $\leftarrow 2T(\frac{d}{2})$

Let $T(d)$ be the # of arithmetic operations that Split does.

$\deg(g) = d$

$\leq 1 \cdot c \cdot d^2 \log p$

Assuming $\deg h = \frac{d}{2}$, $T(d) = 2T(\frac{d}{2}) + \underline{O(d^2 \log p)}$, $T(1) = 0$

rsolve($\{T(d) = 2 \cdot T(d/2) + c \cdot d^2 \cdot \log p, T(1) = 0\}$, $T(n)$);

$2cd^2 \log p - 2cd \log p \in O(d^2 \log p)$

The total cost $<$ twice the cost of the first Powmod.