MODULAR EXPONENTIATION

$m \in \mathbb{N}$, $a, e \in \mathbb{Z}$

Task: compute $a^e \pmod m$

Naive method: $\underbrace{((\ldots((a \times a) \times a) a \ldots \times a)}_{e} \pmod m$

$\underbrace{(\ldots(((a \times a) \pmod m) \times a \pmod m) \ldots) \times a \bmod m}_{e \text{ times}}$

- requires $e-1$ modular multiplications.

Square-and-Multiply: - requires $O(\log_2 e)$ modular multiplications

If $e = -d$, with $d > 0$, then

$$a^e = (a^{-1})^d \pmod m$$

a^{-1} ↓ using extended GCD

$$e = (e_{s-1}, e_{s-2}, \dots, e_1, e_0)_2$$

for $i = s, s-1, \dots, 1, 0$, define

$x_s = 0$ $\quad x_i = (e_{s-1}, e_{s-2}, \dots, e_i)$

(most significant i bits of e)

$$b_i = a^{x_i} \pmod m$$

$x_0 = e \quad b_0 = a^e \pmod m$

$b_s = 1 \pmod m$

$$b_i = a^{x_i} \pmod m$$
$$= a^{2x_{i+1}+e_i} \pmod m$$
$$= \left(a^{x_{i+1}}\right)^2 \cdot a^{e_i} \pmod m$$
$$= \begin{cases} b_{i+1}^2 \pmod m & \text{if } e_i = 0 \\ b_{i+1}^2 \cdot a \pmod m & \text{o.w.} \end{cases}$$

Compute $b_s, b_{s-1}, \dots, b_0$ iteratively.

Algorithm Square-and-Multiply

$b \leftarrow 1 \pmod m$; $a \leftarrow a \pmod m$

for $i \leftarrow s-1$ to 0, do

$b \leftarrow b^2 \pmod m$

if $e_i = 1$ then

$b \leftarrow ba \pmod m$

return b

multiplication followed by Euclidean division

(m: ℓ B-ary digits → multiplication of 2 ℓ-digit numbers followed by Euclidean division of 2ℓ-digit no. by ℓ-digit no)

Running time

s modular squarings + $\leq s$ modular multiplications

$\leq (2 \log_2 e) \cdot$ cost of modular multiplication

Every modular multiplication / squaring involves division by same modulus 'm'

Using this, we can achieve a speed-up at the cost of some pre-computation.

$m = (m_{l-1}, m_{l-2}, \ldots, m_0)_B$

Precompute $T = \left\lfloor \frac{B^{2l}}{m} \right\rfloor$

Let $x = (x_{2l-1}, x_{2l-2}, \ldots, x_1, x_0)_B$

$x \pmod m$ can be computed using T without division.

Algorithm Barrett-Reduction

i/p: $x = (x_{2k-1}, \ldots, x_0)_B$ $\quad m = (m_{k-1}, \ldots, m_0)_B$

o/p: $x \pmod m$

Compute $Q \leftarrow \left\lfloor \frac{x}{B^{k-1}} \right\rfloor$; $Q \leftarrow QT$; $Q \leftarrow \left\lfloor \frac{Q}{B^{k+1}} \right\rfloor$

$R \leftarrow (x - Qm) \pmod{B^{k+1}}$

while $R \geq m$ do

$\quad R \leftarrow R - m$

endwhile

return R

→ executed at most twice

$\because q - 2 \leq Q \leq q$

$q = \left\lfloor \frac{x}{m} \right\rfloor$

$q = \left\lfloor \frac{x}{m} \right\rfloor \qquad q-2 \le Q \le q$ (Assume $m > B^{l-1}$)

$$Q = \left\lfloor \frac{\left\lfloor \frac{x}{B^{l-1}} \right\rfloor \left\lfloor \frac{B^{2l}}{m} \right\rfloor}{B^{l+1}} \right\rfloor \le \left\lfloor \frac{\frac{x}{B^{l-1}} \frac{B^{2l}}{m}}{B^{l+1}} \right\rfloor = \left\lfloor \frac{x}{m} \right\rfloor = q$$

$$Q \ge \left\lfloor \frac{\left(\frac{x}{B^{l-1}} - 1 \right)\left(\frac{B^{2l}}{m} - 1 \right)}{B^{l+1}} \right\rfloor$$

$$= \left\lfloor \frac{x}{m} + \frac{1}{B^{l+1}} - \frac{x}{B^{2l}} - \frac{B^{l-1}}{m} \right\rfloor \ge \left\lfloor \frac{x}{m} \right\rfloor + \left\lfloor \frac{1}{B^{l+1}} \right\rfloor - \left(\left\lfloor \frac{x}{B^{2l}} \right\rfloor + 1 \right) - \left(\left\lfloor \frac{B^{l-1}}{m} \right\rfloor + 1 \right)$$

$$= \left\lfloor \frac{x}{m} \right\rfloor + 0 - (0+1) - (0+1) = q - 2$$

Computation of Q

- 1 multiplication (QT)
- 2 divisions by powers of B

division by B^i: can be done by truncating least significant i digits

Approximate remainder

- 1 multiplication + 1 subtraction
  (Qm) $(r - Qm)$
- r: real remainder
  $q-2 \leq Q \leq q \Rightarrow r \leq R \leq 2m + r$
  at most 2 subtractions

90% cases: $R = r$

9%: $R = r + m$

1%: $R = r + 2m$