Elementos de aritmética (2)
(Teoría de números)

Computación Cuántica: Algorítmica y Software

Joaquín Keller

2026 UCV

Algoritmos clásicos de factorización

Problema: conseguir los factores primos de N

El mejor algoritmo es GNFS, de complejidad: $O(e^{c\sqrt[3]{n\,log^2\,n}})$

n: número de dígitos de N $\quad$ $n = log_2\ N$

Es sub-exponencial pero super-polinomial

GNFS podría romper `RSA-1024`, no se ha hecho porque es muy caro

La recomendación es utilizar `RSA-2048`, seguro hasta el 2030

Algoritmos de test de primalidad

Problema: es p primo?

AKS es de complejidad polinomial, pero es muy lento y no se utiliza

Miller-Rabin es mas rápido, de complejidad polinomial (sin demostrar)

Se 'demuestra' utilizando la conjetura de Riemann

En criptografía se utiliza Miller-Rabin probabilístico que es muy rápido

Y que a veces se equivoca... con una probabilidad ínfima

Congruencia módulo n (1)

Definición: $a \equiv b \pmod{n}$ ssi $n|(a-b)$ $\qquad a, b \in \mathbb{Z} \quad n \in \mathbb{N}^*$

ssi $\exists k \in \mathbb{Z} \quad a = kn + b$

$a \equiv b \pmod{n}$ se lee: a es congruente a b módulo n

Ejemplos: $75 \equiv 3 \pmod{9}$ ya que $75 = 72 + 3 = 9 \times 8 + 3$

$75 \equiv -15 \pmod{9}$ ya que $75 = 9 \times 10 + (-15)$

a es par ssi $a \equiv 0 \pmod{2}$ y a es impar ssi $a \equiv 1 \pmod{2}$

Periodicidad: $a \equiv b \pmod{n}$ ssi $\forall k' \in \mathbb{Z}^* \quad a \equiv k'n + b \pmod{n}$

$a = b \implies a \equiv b \pmod{n}$ (si son iguales, también son congruentes)

Congruencia módulo n (2)

Si $a \equiv b \pmod{n}$ y $a' \equiv b' \pmod{n}$ entonces $a+a' \equiv b+b' \pmod{n}$

Demostración: $\exists k, k' \in \mathbb{Z} \quad a = kn+b \ \wedge \ a' = k'n+b'$

$$\Rightarrow a + a' = (k + k')n + (b + b')$$

Si $a \equiv b \pmod{n}$ y $a' \equiv b' \pmod{n}$ entonces $aa' \equiv bb' \pmod{n}$

Demostración: $\exists k, k' \in \mathbb{Z} \quad a-b = kn \ \wedge \ a'-b' = k'n$

$$\Rightarrow aa' - bb' = aa' - ba' + ba' - bb' = (a - b)a' + (a' - b')b$$
$$= a'kn + bk'n = (a'k + bk')n$$

Y también si $a \equiv b \pmod{n}$ entonces $\forall i \in \mathbb{N} \quad a^i \equiv b^i \pmod{n}$

Por inducción: si $a^{j-1} \equiv b^{j-1} \pmod{n}$ entonces $aa^{j-1} \equiv bb^{j-1} \pmod{n}$

Algoritmos de exponenciación módulo n (1)

Calcular $f(x)$ módulo n es conseguir un y verificando $f(x) \equiv y \pmod{n}$
Escogemos $0 \leq y < n$, el resto de la división entera de $f(x)$ entre n

Calcular $a^1, a^2, a^3, ...a^{k-1}, a^k$ módulo n
Ya sabemos que $a^j \equiv aa^{j-1} \pmod{n}$
Primero calcular $a^1 \pmod{n}$ que es el resto de a^1 divido por n
Repetir con j recorriendo 2 a k:

Multiplicar a por $a^{j-1} \pmod{n}$
el resto de la división por n nos da $a^j \pmod{n}$

Ejemplo:

$7^1 \equiv 2 \pmod{5}$ $7^2 \equiv 7 \times 2 \equiv 14 \equiv 4 \pmod{5}$ $7^3 \equiv 28 \equiv 3 \pmod{5}$

Algoritmos de exponenciación módulo n (2)

Para calcular $a^k \pmod n$ se escribe k en binario
y solo se calculan las potencias de dos correspondientes: a^{2^i}

Ejemplo $3^{21} \pmod{31}$
$21 = 16 + 4 + 1$ $\quad 3^{21} \equiv 3^{16+4+1} \equiv 3^{16} \times 3^4 \times 3^1 \pmod{31}$
Podemos notar que: $a^{2^{i+1}} = a^{2 \times 2^i} = (a^{2^i})^2$
$3^1 \equiv 3 \pmod{31}$ $\quad 3^2 \equiv 9 \pmod{31}$ $\quad 3^4 \equiv 9^2 \equiv 81 \equiv 50 \equiv 19 \pmod{31}$
$3^8 \equiv 19^2 \equiv 361 \equiv 20 \pmod{31}$ $\quad 3^{16} \equiv 20^2 \equiv 400 \equiv 28 \pmod{31}$

$$3^{21} \equiv 3^{16} \times 3^4 \times 3^1 \equiv 28 \times 19 \times 3 \equiv 1596 \equiv 15 \pmod{31}$$

Inverso módulo n (1)

Se dice que a es invertible módulo n ssi $\exists b \in \mathbb{Z} \quad ab \equiv 1 \pmod{n}$

b es el inverso módulo n de a

Ejemplos: 2 es invertible módulo 15: $2 \times 8 \equiv 16 \equiv 1 \pmod{15}$
7 es invertible módulo 15: $7 \times 13 \equiv 91 \equiv 1 \pmod{15}$
3 no es invertible módulo 15

Teorema de Bézout: $\forall a, b \in \mathbb{Z}^* \quad \exists u, v \in \mathbb{Z} \quad \boxed{mcd(a,b) = au + bv}$

Inverso módulo n (2)

a es invertible módulo n ssi $mcd(a, n) = 1$ [i.e. a y n son coprimos]

$$
\begin{aligned}
mcd(a,n) = 1 &\iff \exists u, v \in \mathbb{Z} \quad au + nv = 1 \qquad \text{[Bézout]} \\
&\iff \exists u, v \in \mathbb{Z} \quad au = 1 - nv \\
&\iff \exists u \in \mathbb{Z} \quad au \equiv 1 \pmod{n}
\end{aligned}
$$

El inverso es u.
Los coeficientes de Bézout, u y v, se calculan
con el algoritmo extendido de Euclides para el mcd

Función totiente o función φ de Euler (1)

$$\varphi(n) = \left|\{\, m \in \mathbb{N} \mid m \leq n \ \wedge \ mcd(m,n) = 1 \,\}\right|$$

$\varphi(n)$ es el número de coprimos de n, anteriores a n

Ejemplos:

$\varphi(12)$	$\mathbf{1}, 2, 3, 4, \mathbf{5}, 6, \mathbf{7}, 8, 9, 10, \mathbf{11}, 12$	$\varphi(12) = 4$
$\varphi(10)$	$\mathbf{1}, 2, \mathbf{3}, 4, 5, 6, \mathbf{7}, 8, \mathbf{9}, 10$	$\varphi(10) = 4$
$\varphi(11)$	$\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \mathbf{7}, \mathbf{8}, \mathbf{9}, \mathbf{10}, 11$	$\varphi(11) = 10$

Si p es primo $\varphi(p) = p - 1$

Función φ de Euler (2)

Si p y q son primos y $p \neq q$ $\quad \varphi(pq) = (p-1)(q-1) = \varphi(p)\varphi(q)$

Los **no**-coprimos de pq son $q, 2q, 3q, ..., (p-1)q, pq,\ p, 2p, 3p, ..., (q-1)p$

Es decir que hay $p + q - 1$ no-coprimos de pq menores que pq

El número de coprimos es: $pq - (p + q - 1) = (p-1)(q-1)$

Si p es primo y $k > 0$ $\quad \varphi(p^k) = p^k - p^{k-1}$

Los **no**-coprimos de p^k son los multiples de p menores que p^k

Es decir los enteros αp con $1 \leq \alpha \leq p^{k-1}$, hay p^{k-1} no-coprimos

El cardinal de los coprimos es: $\varphi(p^k) = p^k - p^{k-1}$

Con la descomposición en factores primos $\varphi\left(\prod_{i=1}^{n} p_i^{k_i}\right) = \prod_{i=1}^{n} p_i^{k_i} - p_i^{k_i-1}$

Teorema de Euler

Si a y n son coprimos entonces: $\boxed{a^{\varphi(n)} \equiv 1 \pmod{n}}$