Cuentas

martes, 17 de diciembre de 2019 11:25

Algoritmo de Euclides

Se usa para calcular el $mcd(a,b) \overset{not}{=} (a,b)$

P.1) Se calcula cual es el número mayor (mayor módulo, mayor norma, mayor grado...)

P.2) Se divide el mayor entre el menor (supongamos $a > b$)

$$\begin{array}{l|l} a & b \\ \hline r_1 & q_1 \end{array}$$

- Si $r_1 = 0 \Rightarrow (a,b) = b$
- Si $r_1 \neq 0 \Rightarrow$ (P.3)

P.3) Se divide el divisor de la división anterior entre el resto

$$\begin{array}{l|l} b & r_1 \\ \hline r_2 & q_2 \end{array}$$

- Si $r_2 = 0 \Rightarrow (a,b) = r_1$
- Si $r_2 \neq 0 \Rightarrow$ (P.4)

P.4) Se divide

$$\begin{array}{l|l} r_{n-2} & r_{n-1} \\ \hline r_n & q_n \end{array}$$

- Si $r_n \neq 0 \Rightarrow$ Se vuelve a hacer el (P.4)
- Cuando $r_n = 0 \Rightarrow (a,b) = r_{n-1}$

Algoritmo de Euclides extendido

Se usa para calcular los coeficientes de Bezout

Una vez calculado el $mcd(a,b)$, lo expresamos como combinación lineal de a y b. Para ello, usaremos la siguiente tabla:

q	r	u	v
	a	1	0
q_1	b	0	1
q_2	r_1	$u_1 - q_1 u_2$	$v_1 - q_1 v_2$
q_3	r_2	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
q_{n+1}	r_n		

$$u_i = u_{i-2} - q_{i-2} u_{i-1}$$

$$v_i = v_{i-2} - q_{i-2} v_{i-1}$$

De este modo, nos queda que:

$$mcd(a,b) = u_n a + v_n b$$

. ... , ... que:

$$mcd(a,b) = u_n a + v_n b$$

mcd, mcm e ideales

Sean a y b dos elementos de un DIP $= A$

$d = (a,b)$: $aA + bA = dA \iff d = (a,b)$

$m = [a,b]$: $aA \cap bA = mA \iff m = [a,b]$

$a \cdot b = (a,b) \cdot [a,b]$

(Propiedades del mcd y mcm en apuntes)

Primos e irreducibles

- Primo: p es primo si:
  - no es una unidad
  - si $p \mid ab \Rightarrow p \mid a$ ó $p \mid b$
- Irreducible: a es irreducible si:
  - no es una unidad
  - sólo se divide por sus asociados o por unidades:

    si $q \mid a \Rightarrow q \sim_s a$ ó $q \in \mathcal{U}(A)$

Siempre se cumple que primo $\Rightarrow$ irreducible

Pero irreducible $\Rightarrow$ primo (se cumple la condición de primo) sólo si estamos en un DIP.

Factorización

Diremos que dos factorizaciones "son iguales" si:

- Tienen el mismo número de factores
- Existe una permutación entre los índices tal que:

  $a = a_1 \cdot \ldots \cdot a_r = a'_1 \cdot \ldots \cdot a'_s \quad \exists r$ permutación

  $a_i \sim_s a_{r(i)}$

Algoritmo de factorización en $\mathbb{Z}[\sqrt{n}]$

Sea $\alpha = a + b\sqrt{n}$

P.1) Factorizamos la norma de α:

$\|\alpha\| = a^2 - nb^2 = p_1 \cdot \ldots \cdot p_n$

P.2) Para el primer p_1, buscamos un número en $\mathbb{Z}[\sqrt{n}]$ cuya norma:

- Sea igual a p_1
- Sea igual a p_1^2

P.3) Una vez encontrado este número, al que llamaremos $\pi_1 \in \mathbb{Z}[\sqrt{n}]$, dividimos

$$\begin{array}{l|l} \alpha & \pi_1 \\ \hline 0 & \gamma \end{array}$$

Repetimos desde el (P.1) con γ, hasta quedarnos sin primos y vamos expresando $\alpha = \pi_1 \cdot \ldots \cdot \pi_n$

Ecuaciones diofánticas

$$ax + by = c$$

Este tipo de ecuaciones tiene solución $\Leftrightarrow (a,b) = d \mid c$. Una vez comprobado que tiene solución, comenzamos:

P.1) Aplicamos el Algoritmo de Euclides extendido y expresamos:

$$d = ua + vb$$

P.2) Como hemos comprobado que $d \mid c$, dividimos.

$$\begin{array}{l|l} c & d \\ \hline 0 & q \end{array}$$

P.3) Multiplicamos

$$qd = c = (u \cdot q)a + (v \cdot q)b \Rightarrow \begin{cases} x_0 = uq \\ y_0 = vq \end{cases}$$

P.4) Calculamos las soluciones generales:

$$\begin{cases} b' = \dfrac{b}{d} \\ a' = \dfrac{a}{d} \end{cases}$$

$$x = x_0 + b' \cdot k$$
$$y = y_0 - a' \cdot k$$

Ecuaciones básicas en congruencias

$ax \equiv c \mod b$ Tiene solución $\Leftrightarrow (a,b) = d \mid c$

P.1) La expresamos como una ecuación diofántica:

$ax \equiv c \bmod b \Rightarrow ax - c \in bA \Rightarrow \exists -y \in A \mid q \Rightarrow ax - c = b(-y) \Rightarrow$

$ax \equiv c \bmod b \Rightarrow ax - c \in bA \Rightarrow \exists -y \in A$ tq $\Rightarrow ax - c = b(-y) \Rightarrow$

$$ax + by = c$$

P.2) Resolvemos según lo explicado en el apartado anterior, y nos quedamos con x_0.

P.3) Ahora, la solución general viene dada "aplicando" módulo b:

$$x \equiv x_0 \bmod b$$

Apunte

Para ver si a es una unidad de $\mathbb{Z}_n$, debemos ver si la ec básica:

$$ak \equiv 1 \bmod n$$

tiene solución.

De esto, se desprende que, si n es un número primo, el anillo será un cuerpo, pues $ax \equiv 1 \bmod n$ tendrá solución siempre, pues: $mcd(a,n) = 1 \ \forall a \in \mathbb{Z}_n$ (al ser n primo). Por tanto, todos los elementos, salvo el 0 son unidades

Función φ de Euler

La función φ de Euler nos dará la cantidad de unidades que hay en un anillo $\mathbb{Z}_n$, y está definida por las dos siguientes propiedades:

① $\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b)$

② $\varphi(p^e) = p^{e-1}(p-1)$

p = primo

Teorema de Euler

Si $(n,a) = 1 \Rightarrow a^{\varphi(n)} \equiv 1 \bmod n$

Aplicación del Teorema de Euler: RSA

Calcular $a^b \bmod n$

Necesitamos n primo y $(a,n) = 1$

Por tanto, calculamos $\varphi(n)$

$$b \;\underline{|\;\varphi(n)}\qquad r \quad q$$

$$a^b \bmod n \equiv \cancel{(a^q)^{\varphi(n)}}^{1} \circ a^r \bmod n$$

RSA:

Tenemos un dato a. Lo codificamos: $y = a^b \bmod n$.

- Las dos llaves públicas son b y n.
- Normalmente $n = p \cdot q$ (siendo p y q dos primos grandes)
- Como ya sabemos a, n deben ser coprimos
- Además $y, \varphi(n)$ también deben ser coprimos.

$$a \xrightarrow{\text{cod}} y = a^b \bmod n \xrightarrow[\text{el dato } y]{\text{se pasa}} y \xrightarrow{\text{decod}} a$$

Para calcular el dato codificado:

$(b, \varphi(n))$ debe ser 1, por tanto:

$$1 = ub + v\varphi(n)$$

Por tanto:

$$a \equiv a^1 \equiv a^{(ub + v\varphi(n))} \equiv (a^b)^u \cdot (a^{\varphi(n)})^v \bmod n$$

como: $a^{\varphi(n)} \equiv 1 \bmod n$ y $a^b = y$:

$$a \equiv y^u \bmod n$$

Sistemas de ecuaciones en congruencias

$$x \equiv a_1 \bmod n_1$$
$$x \equiv a_2 \bmod n_2$$
$$\vdots$$
$$x \equiv a_r \bmod n_r$$

Teorema chino de los restos

Un sistema de ecuaciones en congruencias tiene solución si y sólo si:

$$a_i \equiv a_j \bmod (n_i, n_j) \qquad \forall i = j$$

Y su solución general vendrá dada por:

$$x = x_0 + k \, \text{mcm}(n_1, \dots, n_r)$$

Algoritmo:

P.0) Escogemos una pareja de ecuaciones

P.1) Expresamos en la primera ecuación:

$$x = a_1 + k \cdot n_1$$

P.2) Sustituimos x en la segunda. Resolvemos la ecuación básica y despejamos $k_0 \Rightarrow x_0 = a_1 + k_0 n_1$

P.3) Buscamos la solución general:

Como $k = k_0 + \frac{n_2}{(n_1, n_2)} t$

$$x = a_1 + \left(k_0 + \frac{n_2}{(n_1, n_2)} t\right) n_1 = \underbrace{(a_1 + k_0 n_1)}_{x_0} + \underbrace{\frac{n_2 n_1}{(n_1, n_2)}}_{[n_1, n_2]} t \Rightarrow$$

$$x = x_0 + [n_1, n_2] t$$

P.4) Sustituimos las dos ecuaciones escogidas por: $x \equiv x_0 \mod [n_1, n_2]$

Quedando el sistema:
$$\begin{aligned} x &\equiv x_0 \mod [n_1, n_2] \\ x &\equiv a_3 \mod n_3 \\ &\vdots \\ x &\equiv a_r \mod n_r \end{aligned}$$

Retomamos el (P.0) hasta quedarnos sin ecuaciones.