Aritmética Modular II

Este documento no pretende ser un curso exhaustivo, en el mejor de los casos, únicamente puede considerarse como un conjunto de notas complementarias en alguna asignatura. Por supuesto, todo documento es susceptible de mejora, cualquier sugerencia o comunicación de error u omisión será bienvenida.

Máximo común divisor

Algunos de los sistemas de cifrado más utilizados hoy en día se definen considerando el conjunto $\mathbb{Z}_n$ y la operación producto. Sin embargo, trabajando módulo n, no para todo valor existe el inverso para el producto. Las condición para la existencia del inverso viene establecida en el siguiente teorema:

Teorema 1. *Si* $mcd(a, n) = 1$ *(a y n son relativamente primos), entonces la congruencia* $ax \equiv b \pmod{n}$ *tiene una única solución módulo* n*.*

En general, dara una congruencia cualquiera, si denotamos con d el $mcd(a, n)$, conociendo una solución x_0 de la congruencia podemos obtener todas las posibles soluciones, ya que estas tienen la forma:

$$x = \left\{ x_0 + k\frac{n}{d} \ : \ k \in \mathbb{Z} \right\}$$

Ejemplo 1. *Consideremos* $\mathbb{Z}_{27}$*, teniendo en cuenta que* $d = mcd(11, 27) = 1$*, por el resultado anterior sabemos que existe un único valor* x *tal que* $11x \equiv 6 \pmod{27}$*, en efecto si* $x = 3$ *tenemos que* $11 \cdot 3 = 33 \equiv 6 \pmod{27}$*.*

Para el cálculo de inversos del producto estaremos interesados en las congruencias del tipo $ax \equiv 1 \pmod{n}$*. En nuestro ejemplo, el inverso de* 11 *módulo* 27 *será el valor* x *que cumpla* $11x \equiv 1 \pmod{27}$*, que en este caso es* $x = 5$*.*

Además de establecer las condiciones en las que existe, será importante disponer de un algoritmo para el cálculo de inversos para el producto en $\mathbb{Z}_n$ de forma eficiente. En este cálculo tiene un papel clave el siguiente resultado:

Teorema 2. *El* $mcd(a,n)=1$ *es el menor entero estrictamente positivo del conjunto de combinaciones lineales de* a *y* b*, esto es:*

$$mcd(a,b) = \underset{>0}{Min}\left\{ d = xa + yb \ : \ x,y \in \mathbb{Z},\ d>0 \right\}$$

El Algoritmo de Euclides (300 a.C) calcula el máximo común divisor de dos enteros reduciendo recursivamente el cálculo al máximo común divisor de dos valores menores y combinación lineal de los de entrada.

Algoritmo 1 Algoritmo de Euclides

Entrada: Dos enteros a y b
Salida: $mcd(a,b)$

1: **Si** $b == 0$ **entonces**
2: $Return(a)$
3: **else**
4: **Devolver** $Euclides(b, a \text{ mód } b)$
5: **FinSi**

Ejemplo 2. *Siguiendo el algoritmo de Euclides, el* $mcd(30,21)$ *puede reducirse sucesivamente al* $mcd(21,9)$*,* $mcd(9,3)$ *y al* $mcd(3,0)=3$*.*

Una modificación ligera de este algoritmo permite recuperar los coeficientes de la combinación lineal cuyo resultado es el máximo común divisor. La modificación se conoce como *Algoritmo Extendido de Euclides.*

Algoritmo 2 Algoritmo Extendido de Euclides

Entrada: Dos enteros a y b
Salida: $mcd(a,b)$, x e y tales que $ax+by=mcd(a,b)$

1: **Si** $b == 0$ **entonces**
2: $Return(a,1,0)$
3: **else**
4: $(d', x', y') = EuclidesExt(b, a \text{ mód } b))$
5: $(d, x, y) = (d', y', x' - \lfloor a/b \rfloor y')$
6: $Return(d,x,y)$
7: **FinSi**

Una vez obtenida la combinación lineal que da como resultado el máximo común divisor

de dos valores x y n relativamente primos ($mcd(x, n) = 1$), tenemos:

$$ax + bn = 1,$$

si consideramos que trabajamos módulo n, entonces:

$$ax + bn \text{ mód } n = ax = 1,$$

por lo que a es el inverso de x para el producto.

Ejemplo 3. *Calcularemos el* $mcd(8, 27)$ *utilizando el algoritmo extendido de Euclides. Primero, el algoritmo desciende recursivamente reduciendo el problema:*

a	b	d	x	y
27	*8*			
8	*3*			
3	*2*			
2	*1*			
1	*0*	*1*	*1*	*0*

una vez alcanzada la solución calcula, mientras deshace la pila de recursión, los coeficientes de la combinación lineal que da como resultado el máximo común divisor (a partir de los valores (d, x', y') *devueltos, se calculan los nuevos coeficientes como* $(d, x, y) = (d', y', x' - \lfloor a/b \rfloor y')$*).*

a	b	d	x	y
27	8	1	3	$-1 - \lfloor 27/8 \rfloor \cdot 3 = -10$
8	3	1	-1	$1 - \lfloor 8/3 \rfloor \cdot (-1) = 3$
3	2	1	1	$0 - \lfloor 3/2 \rfloor \cdot 1 = -1$
2	1	1	0	$1 - \lfloor 2/1 \rfloor \cdot 0 = 1$
1	0	1	1	0

Resaltamos que en en cada iteración se dispone de los coeficiente de una combinación lineal cuyo resultado es d, por ejemplo $2 \cdot 0 + 1 \cdot 1 = 1$ *pero también* $-1 \cdot 8 + 3 \cdot 3 = 1$ *o, finalmente,* $27 \cdot 3 + 8 \cdot (-10) = 1$.

Leyendo el resultado en los términos que nos interesan tenemos que 3 *es el inverso de* 3 *módulo* 8*, o que* $-10 \equiv 17 \pmod{27}$ *es el inverso de* 8 *módulo* 27

Respecto el coste temporal del algoritmo extendido de Euclides, dados a y b, el algoritmo realiza $\mathcal{O}(\log b)$ llamadas recursivas (considerando la talla de b como su representación binaria). Otra forma de analizar el coste es considerar que, si $a > b \geq 1$ y $b < Fibonacci(k)$, entonces el algoritmo realizará $k - 1$ llamadas recursivas.

Ejercicios

Ejercicio 1.

Calcular el $mcd(75, 23)$ *y los coeficientes de la combinación lineal asociada. ¿Cuál es el inverso de* 23 *módulo* 75*?*

Solución:

$75 \cdot 4 + 23 \cdot (-13) = 1$

$23^{-1} \text{ mód } 75 = -13 \text{ mód } 75 = 62$

Ejercicio 2.

Calcular el $mcd(100, 61)$ *y los coeficientes de la combinación lineal asociada. ¿Cuál es el inverso de* 61 *módulo* 100*? ¿y el inverso de* 61 *módulo* 100 *?*

Solución:

$100 \cdot (-25) + 61 \cdot 41 = 1$

$61^{-1} \text{ mód } 100 = 41$

$100^{-1} \text{ mód } 61 = -25 \text{ mód } 61 = 36$

Ejercicio 3.

Calcular el $mcd(321, 27)$ *y los coeficientes de la combinación lineal asociada. ¿Cuál es el inverso de* 27 *módulo* 321*?*

Solución:

$321 \cdot (-1) + 27 \cdot 12 = 3$

No existe el inverso de 27 *módulo* 321 *porque los valores no son relativamente primos.*
