Teoría de Números I

U.D. Computación

DSIC - UPV

Contenidos del tema

Bibliografía

- Handbook of applied crytography. *A. J. Menezes, P. C. van Oorshot and S. A. Vanstone*. CRC Press. 1996.

  (Capítulo 2)

- Introduction to algorithms. *C. E. Leiserson, C. Stein, R. Rivest and T. H. Cormen*. The MIT Press (3rd edition) 2009.

  (Capítulo 31)

Aritmética Modular

Aritmética Modular: Grupo

- Grupo $\langle G, \otimes \rangle$
  - $\otimes$ es de composición interna
  - $\otimes$ es asociativa
  - Existe un elemento neutro en G
  - Todo $a \in G$ tiene inverso respecto $\otimes$

Aritmética Modular: Reducción modulo n

- $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$

Aritmética Modular: Reducción modulo n

- $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$
- Congruencia módulo n:

$$a \equiv b \pmod{n} \iff a - b = kn,\ k \in \mathbb{Z}$$

Aritmética Modular: Reducción modulo n

- $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$
- Congruencia módulo n:

$$a \equiv b \pmod{n} \iff a - b = kn,\ k \in \mathbb{Z}$$

- Reducción módulo n:

$$a \bmod n$$

Aritmética Modular: Reducción modulo n

- $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$
- Congruencia módulo n:

$$a \equiv b \pmod{n} \iff a - b = kn,\ k \in \mathbb{Z}$$

- Reducción módulo n:

$$a \bmod n$$

- Relación de equivalencia.

Aritmética Modular

La suma y el producto son compatibles con la congruencia. Dados $a \equiv a' \pmod{n}$ y $b \equiv b' \pmod{n}$:

- $a + b \equiv a' + b' \pmod{n}$
- $ab \equiv a'b' \pmod{n}$

Definimos en $\mathbb{Z}_n$:

- $[a]_{\equiv_n} + [b]_{\equiv_n} = [a + b]_{\equiv_n}$
- $[a]_{\equiv_n}[b]_{\equiv_n} = [ab]_{\equiv_n}$

Aritmética Modular

$(\mathbb{Z}_n, +, \cdot)$ posea estructura de anillo conmutativo.

- Son operaciones cerradas y conmutativas.
- $(\mathbb{Z}_n, +)$ es un grupo (la operación es asociativa, tiene elemento neutro e inverso para todo valor en $\mathbb{Z}_n$).
- $(\mathbb{Z}_n, .)$ es un semigrupo (la operación es asociativa y tiene elemento neutro).
- El producto distribuye respecto la suma.

Aritmética Modular

Cálculo de inversos para el producto

- Teorema (de congruencia lineal):
  $ax \equiv b \pmod{n}$ tiene una única solución sii $mcd(a, n)$ divide b.

- a es invertible si $mcd(a, m) = 1$.
- Si $mcd(a, m) = 1$ entonces a y m son relativamente primos

Máximo común divisor

- $d|a$ y $d|b$ implica que $\forall x, y \in \mathbb{Z}$, se cumple que $d|xa + yb$
- $a|b$ implica que $|a| \leq |b| \ \vee \ b = 0$
- $a|b$ y $b|a$ implica que $a = \pm b$

Máximo común divisor

Divisores: Propiedades

- Teorema: $mcd(a, b)$ es el menor entero estrictamente positivo del conjunto $\{xa + yb \ : \ x, y \in \mathbb{Z}\}$ (combinaciones lineales de a y b)
- Corolario: $d|a$ y $d|b$ implica que $d|mcd(a, b)$
- Teorema: $mcd(a, b) = mcd(b, a \text{ mód } b)$

Máximo común divisor

Cálculo del *mcd*: Implementación

Algoritmo de Euclides:

Euclides(a, b):

if $b = 0$ **then**
    Return(a)
else
    Return(*Euclides*(b, a mód b))
end if

Máximo común divisor

Cálculo del *mcd*: Implementación

Algoritmo de Euclides:

Euclides(a, b):

if $b = 0$ **then**
    Return(a)
else
    Return(*Euclides*(b, a mód b))
end if

- Coste del algoritmo: $\mathcal{O}(\log b)$
- Ejemplo:
  $Euclides(30, 21) = Euclides(21, 9) = Euclides(9, 3) = Euclides(3, 0) = 3$

Máximo común divisor
Cálculo del *mcd*: Implementación

Para el cálculo de inversos del producto, es interesante obtener el $mcd(a, b)$ como combinación lineal de a y b.

Máximo común divisor
Cálculo del *mcd*: Implementación

Para el cálculo de inversos del producto, es interesante obtener el $mcd(a, b)$ como combinación lineal de a y b.

$$\begin{aligned} mcd(a, n) = 1 \quad &\Rightarrow \; xa + yn = 1 \; \Rightarrow \\ &\Rightarrow \; xa \equiv 1 \pmod{n} \; \Rightarrow \\ &\Rightarrow \; a \equiv x^{-1} \pmod{n} \end{aligned}$$

Máximo común divisor

Cálculo del *mcd*: Implementación

Para el cálculo de inversos del producto, es interesante obtener el $mcd(a, b)$ como combinación lineal de a y b.

EuclidesExt(a, b):

if $b = 0$ **then**
    $Return(a, 1, 0)$
else
    $(d', x', y') = EuclidesExt(b, a \text{ mód } b))$
    $(d, x, y) = (d', y', x' - \lfloor a/b \rfloor y')$
    $Return(d, x, y)$
end if

Máximo común divisor
Cálculo del *mcd*: Ejemplo

- Ejemplo:

a	b	(d', x', y')	$\lfloor a/b \rfloor$	d	x	y
8	5					

Máximo común divisor

Cálculo del *mcd*: Ejemplo

- Ejemplo:

a	b	(d', x', y')	$\lfloor a/b \rfloor$	d	x	y
8	5					
5	3					

Máximo común divisor

Cálculo del *mcd*: Ejemplo

- Ejemplo:

a	b	(d', x', y')	$\lfloor a/b \rfloor$	d	x	y
8	5					
5	3					
3	2					

Máximo común divisor

Cálculo del *mcd*: Ejemplo

- Ejemplo:

a	b	(d', x', y')	$\lfloor a/b \rfloor$	d	x	y
8	5					
5	3					
3	2					
2	1					

Máximo común divisor

Cálculo del *mcd*: Ejemplo

- Ejemplo:

a	b	(d', x', y')	$\lfloor a/b \rfloor$	d	x	y
8	5					
5	3					
3	2					
2	1					
1	0					

Máximo común divisor

Cálculo del *mcd*: Ejemplo

- Ejemplo:

a	b	(d', x', y')	$\lfloor a/b \rfloor$	d	x	y
8	5					
5	3					
3	2					
2	1	$(1, 1, 0)$				
1	0					

Máximo común divisor

Cálculo del *mcd*: Ejemplo

- Ejemplo:

a	b	(d', x', y')	$\lfloor a/b \rfloor$	d	x	y
8	5					
5	3					
3	2					
2	1	$(1, 1, 0)$	2	1	0	1
1	0					

$$(d, x, y) = (d', y', x' - \lfloor a/b \rfloor y')$$

Máximo común divisor

Cálculo del *mcd*: Ejemplo

- Ejemplo:

a	b	(d', x', y')	$\lfloor a/b \rfloor$	d	x	y
8	5					
5	3					
3	2	$(1, 0, 1)$				
2	1	$(1, 1, 0)$	2	1	0	1
1	0					

$$(d, x, y) = (d', y', x' - \lfloor a/b \rfloor y')$$

Máximo común divisor

Cálculo del *mcd*: Ejemplo

- Ejemplo:

a	b	(d', x', y')	$\lfloor a/b \rfloor$	d	x	y
8	5					
5	3					
3	2	$(1, 0, 1)$	1	1	1	-1
2	1	$(1, 1, 0)$	2	1	0	1
1	0					

$$(d, x, y) = (d', y', x' - \lfloor a/b \rfloor y')$$

Máximo común divisor

Cálculo del *mcd*: Ejemplo

- Ejemplo:

a	b	(d', x', y')	$\lfloor a/b \rfloor$	d	x	y
8	5					
5	3	$(1, 1, -1)$				
3	2	$(1, 0, 1)$	1	1	1	-1
2	1	$(1, 1, 0)$	2	1	0	1
1	0					

$$(d, x, y) = (d', y', x' - \lfloor a/b \rfloor y')$$

Máximo común divisor

Cálculo del *mcd*: Ejemplo

- Ejemplo:

a	b	(d', x', y')	$\lfloor a/b \rfloor$	d	x	y
8	5					
5	3	$(1, 1, -1)$	1	1	-1	2
3	2	$(1, 0, 1)$	1	1	1	-1
2	1	$(1, 1, 0)$	2	1	0	1
1	0					

$$(d, x, y) = (d', y', x' - \lfloor a/b \rfloor y')$$

Máximo común divisor

Cálculo del *mcd*: Ejemplo

- Ejemplo:

a	b	(d', x', y')	$\lfloor a/b \rfloor$	d	x	y
8	5	$(1, -1, 2)$				
5	3	$(1, 1, -1)$	1	1	-1	2
3	2	$(1, 0, 1)$	1	1	1	-1
2	1	$(1, 1, 0)$	2	1	0	1
1	0					

$$(d, x, y) = (d', y', x' - \lfloor a/b \rfloor y')$$

Máximo común divisor

Cálculo del *mcd*: Ejemplo

- Ejemplo:

a	b	(d', x', y')	$\lfloor a/b \rfloor$	d	x	y
8	5	$(1, -1, 2)$	1	1	2	-3
5	3	$(1, 1, -1)$	1	1	-1	2
3	2	$(1, 0, 1)$	1	1	1	-1
2	1	$(1, 1, 0)$	2	1	0	1
1	0					

$$(d, x, y) = (d', y', x' - \lfloor a/b \rfloor y')$$

- Ejemplo:

a	b	(d', x', y')	$\lfloor a/b \rfloor$	d	x	y
8	5	$(1, -1, 2)$	1	1	2	-3
5	3	$(1, 1, -1)$	1	1	-1	2
3	2	$(1, 0, 1)$	1	1	1	-1
2	1	$(1, 1, 0)$	2	1	0	1
1	0					

Por lo que el algoritmo devuelve $(1, 2, -3)$, esto es:
$mcd(8, 5) = 1 = 2 \cdot 8 \ - \ 3 \cdot 5$
$5^{-1} \equiv 5 \pmod 8$

Máximo común divisor

Complejidad del cálculo del *mcd*

Dados a y b, el algoritmo extendido de Euclides realiza $\mathcal{O}(\log b)$ llamadas recursivas (consideramos la talla de b como su representación binaria).

Otra forma de analizar el coste es considerar que, si $a > b \geq 1$ y $b < \mathit{Fibonacci}(k)$, entonces el algoritmo realiza $k - 1$ llamadas recursivas

Resíduos cuadráticos

Resíduos cuadráticos
Definición

$a \in \mathbb{Z}_n^*$ es *resíduo cuadrático módulo n* si existe un $x \in \mathbb{Z}_n^*$ tal que $x^2 \equiv a \pmod{n}$. En caso de existir, x se comoce como *raíz (de a) módulo n*.

Dado un entero impar n y un entero $0 < a < n$, determinar si a es un resíduo cuadrático módulo n puede reducirse al problema de factorizar el entero n.

Si el valor modular es primo, existe un algoritmo probabilista que obtiene (con cierta seguridad) las raíces de un número a con complejidad $\mathcal{O}(|a|^4)$. No se conoce algoritmo polinómico para este problema.