

Algoritmo de Euclides y ecuaciones de congruencia

Taller de Álgebra I

Primer cuatrimestre 2018

Algoritmo de Euclides

El **algoritmo de Euclides** calcula el máximo común divisor entre dos números $a, b \in \mathbb{Z}$.

Se basa en que si $a, b \in \mathbb{Z}$ y $k \in \mathbb{Z}$ es un número cualquiera, entonces

$$(a : b) = (a + kb : b)$$

Si q y r son el cociente y el resto de la división de a por b , tenemos $a = qb + r$, entonces $a - qb = r$. Por lo tanto,

$$(a : b) = (a - qb : b) = (r : b) = (b : r)$$

Por ejemplo, para calcular $(30 : 48)$:

- 1 $(30 : 48)$ — Dividimos 30 por 48, $q = 0$, $r = 30$
- 2 $= (48 : 30)$ — Dividimos 48 por 30, $q = 1$, $r = 18$
- 3 $= (30 : 18)$ — $q = 1$, $r = 12$
- 4 $= (18 : 12)$ — $q = 1$, $r = 6$
- 5 $= (12 : 6)$ — $q = 2$, $r = 0$
- 6 $= (6 : 0)$
- 7 $= 6$

Ejercicios

- 1 Programar la función
`mcd :: Integer -> Integer -> Integer`
que calcule el máximo común divisor entre dos números utilizando el algoritmo de Euclides.
`mcd a b` debe funcionar siempre que $a > 0$, $b \geq 0$.
- 2 Pensar otro algoritmo para calcular el máximo común divisor modificando la función `menorDivisor` programada en clases anteriores.
- 3 Comparar el tiempo que tardan ambos programas para números pequeños y números grandes (por ejemplo, números de 10 dígitos).^a.

^aPara que el intérprete muestre los tiempos de ejecución hay que ejecutar `:set +s` en la consola

Algoritmo de Euclides extendido

Dados números $a, b \in \mathbb{Z}$, existen enteros s, t tales que

$$sa + tb = (a : b).$$

Ejemplos

- ▶ $(8 : 5) = 1$ y $2 \cdot 8 - 3 \cdot 5 = 1$
- ▶ $(9 : 15) = 3$ y $2 \cdot 9 - 1 \cdot 15 = 3$

Los valores de s, t se pueden obtener con la versión extendida del algoritmo de Euclides.
Por algoritmo de división:

$$a = bq + r \tag{1}$$

y sabemos que $(a : b) = (b : r) = g$.

Si tenemos s_1, t_1 tales que $s_1 b + t_1 r = g$, ¿cómo obtenemos s y t ?

Multiplicamos (??) por t_1 :

$$t_1 a = t_1 b q + t_1 r$$

y reemplazamos la expresión de $t_1 r$:

$$t_1 a = t_1 b q + (g - s_1 b)$$

Obtenemos

$$t_1 a + (s_1 - t_1 q) b = g$$

Es decir, $s = t_1$ y $t = (s_1 - t_1 q)$.

Algoritmo extendido de Euclides

- 1 Programar la función

`emcd :: Integer -> Integer -> (Integer, Integer, Integer)`

que utilice el algoritmo de Euclides extendido para obtener una 3-upla (g, s, t) tal que $g = (a : b) = sa + tb$.

Utilizando la función `emcd` encontrar alguna solución de las ecuaciones diofánticas

- 1 $29x + 17y = 1$

- 2 $89x + 23y = 1$

- 3 $93x + 27y = 3$

Ecuaciones de congruencia

Usando el algoritmo extendido de Euclides podemos resolver ecuaciones de congruencia

$$ax \equiv b \pmod{m}$$

¿Cuándo tiene solución la ecuación?

Recordar que esa ecuación es equivalente a la ecuación *diofántica*

$$ax + mk = b$$

Tiene solución si y solo si $(a : m)$ divide a b .

Ejercicios

- 1 Programar la función
`tieneSolucion :: Integer -> Integer -> Integer -> Bool`
que dados a , b y m determine si la ecuación $ax \equiv b \pmod{m}$ tiene solución.
- 2 Programar la función
`solucionParticular :: Integer -> Integer -> Integer -> Integer`
que dados a , b y m determine, si existe, un entero x tal que $ax \equiv b \pmod{m}$. (La función puede quedar indefinida si la ecuación no tiene solución.)
Sugerencia: encontrar primero una solución de $ax \equiv (a : m) \pmod{m}$

Solución general de una ecuación de congruencia

La solución de la ecuación

$$ax \equiv b \pmod{m}$$

es una clase de congruencia.

Por ejemplo, la solución de $3x \equiv 2 \pmod{5}$ es $x \equiv 4 \pmod{5}$.

Recordamos de la teórica cómo resolver una ecuación de congruencia.

Ejemplo: $35x \equiv 15 \pmod{20}$

- 1 Calculamos $(35 : 20) = 5$ y dividimos la ecuación por 5:

$$7x \equiv 3 \pmod{4}$$

(si el mcd no divide a b , la ecuación no tiene solución)

- 2 Pasamos a ecuación diofántica $7x + 4y = 3$.

- 3 Buscamos una solución particular de $7x + 4y = 1$, por ejemplo $7 \cdot (-1) + 4 \cdot 2 = 1$.

- 4 Multiplicamos por 3: $7 \cdot (-3) + 4 \cdot 6 = 3$

- 5 La solución general es $x \equiv (-3) \pmod{4}$.

Solución general de una ecuación de congruencia

Vamos a representar las clases de congruencia por pares ordenados de enteros. El par (c, n) (con $n > 0$) representa todos los $x \in \mathbb{Z}$ con $x \equiv c \pmod{n}$.

Ejercicios

- 1 Programar la función
`solucionGeneral :: Integer -> Integer -> Integer -> (Integer, Integer)`
que dados a , b y m determine la clase de congruencia solución de la ecuación $ax \equiv b \pmod{m}$.

Ejemplos

- ▶ `solucionGeneral 3 2 5` \rightsquigarrow `(4, 5)`
 - ▶ `solucionGeneral 35 15 20` \rightsquigarrow `(-3, 4)`
- Observación: $(-3, 4) = (1, 4)$