Algoritmo de Euclides y ecuaciones de congruencia

Taller de Álgebra I

Primer cuatrimestre 2019

Algoritmo de Euclides

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

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

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

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,

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

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, $q = 0$, $r = 30$

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, $q = 1$, $r = 18$

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. $= (30 : 18)$ — $q = 1$, $r = 12$

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. $= (18 : 12)$ — $q = 1$, $r = 6$

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. $= (12 : 6)$ — $q = 2$, $r = 0$

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. $= (6 \ : \ 0)$

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. $= 6$

Algoritmo de Euclides: Ejercicios

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. Ideas:
   - usar la función `menorDivisor` programada en clases anteriores.
   - extender la función `menorDivisor` para que calcule el `mayorDivisorComun`

3. Comparar el tiempo que tardan ambos programas para números pequeños y números grandes (por ejemplo, números de 10 dígitos). En GHCI se puede saber el tiempo que tarda una función utilizado el comando: `set +s`

Algoritmo de Euclides extendido

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

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

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

Ejemplos

- $(8 \ : \ 5) = 1 \quad \text{y} \quad 2 \cdot 8 - 3 \cdot 5 = 1$
- $(9 \ : \ 15) = 3 \quad \text{y} \quad 2 \cdot 9 - 1 \cdot 15 = 3$

Algoritmo de Euclides extendido

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

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

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

Ejemplos

- $(8 : 5) = 1 \quad \text{y} \quad 2 \cdot 8 - 3 \cdot 5 = 1$
- $(9 : 15) = 3 \quad \text{y} \quad 2 \cdot 9 - 1 \cdot 15 = 3$

Un ejemplo fácil

Para el caso de $(n : 0)$ es trivial.

Algoritmo de Euclides extendido

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

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

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

Ejemplos

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

Un ejemplo fácil

Para el caso de $(n \ : \ 0)$ es trivial.
$(n \ : \ 0) = n = n * \mathbf{1} + 0 * \mathbf{0}$.
Es decir, $s = 1$ y $t = 0$

Algoritmo de Euclides extendido

Por algoritmo de división:

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

Y por el algoritmo de Euclides clásico, sabemos que:

$$(a : b) = (b : r) = g \tag{2}$$

Algoritmo de Euclides extendido

Por algoritmo de división:

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

Y por el algoritmo de Euclides clásico, sabemos que:

$$(a : b) = (b : r) = g \tag{2}$$

Si asumimos que tenemos s', t' tales que:

$$s'b + t'r = g \tag{3}$$

¿Cómo obtenemos $sa + tb = g$?

Algoritmo de Euclides extendido

Por algoritmo de división:

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

Y por el algoritmo de Euclides clásico, sabemos que:

$$(a : b) = (b : r) = g \tag{2}$$

Si asumimos que tenemos s', t' tales que:

$$s'b + t'r = g \tag{3}$$

¿Cómo obtenemos $sa + tb = g$?
Multiplicamos (1) por $\boldsymbol{t'}$:

$$\boldsymbol{t'}a = \boldsymbol{t'}bq + \boldsymbol{t'}r$$

Algoritmo de Euclides extendido

Por algoritmo de división:

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

Y por el algoritmo de Euclides clásico, sabemos que:

$$(a : b) = (b : r) = g \tag{2}$$

Si asumimos que tenemos s', t' tales que:

$$s'b + t'r = g \tag{3}$$

¿Cómo obtenemos $sa + tb = g$?
Multiplicamos (1) por $\boldsymbol{t'}$:

$$\boldsymbol{t'}a = \boldsymbol{t'}bq + \boldsymbol{t'}r$$

Reemplazamos la expresión de $t'r$ según (3):

$$t'a = t'bq + (\boldsymbol{g} - \boldsymbol{s'b})$$

Algoritmo de Euclides extendido

Por algoritmo de división:

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

Y por el algoritmo de Euclides clásico, sabemos que:

$$(a : b) = (b : r) = g \tag{2}$$

Si asumimos que tenemos s', t' tales que:

$$s'b + t'r = g \tag{3}$$

¿Cómo obtenemos $sa + tb = g$?
Multiplicamos (1) por $\boldsymbol{t'}$:

$$\boldsymbol{t'}a = \boldsymbol{t'}bq + \boldsymbol{t'}r$$

Reemplazamos la expresión de $t'r$ según (3):

$$t'a = t'bq + (\boldsymbol{g - s'b})$$

Despejamos g

$$t'a - t'bq + s'b = g$$

Algoritmo de Euclides extendido

Por algoritmo de división:

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

Y por el algoritmo de Euclides clásico, sabemos que:

$$(a : b) = (b : r) = g \tag{2}$$

Si asumimos que tenemos s', t' tales que:

$$s'b + t'r = g \tag{3}$$

¿Cómo obtenemos $sa + tb = g$?
Multiplicamos (1) por $\boldsymbol{t'}$:

$$\boldsymbol{t'}a = \boldsymbol{t'}bq + \boldsymbol{t'}r$$

Reemplazamos la expresión de $t'r$ según (3):

$$t'a = t'bq + (\boldsymbol{g - s'b})$$

Despejamos g

$$t'a - t'bq + s'b = g$$

Sacamos factor común

$$\boldsymbol{t'}a + (\boldsymbol{s' - t'q})b = g$$

Es decir, $s = t'$ y $t = (s' - t'q)$.

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 $g = (a : b) = sa + tb$.
Recordar que $s = t'$ y $t = (s' - t'q)$ (ver diapositiva anterior).

Sugerencia: para acceder a los elementos de la 3-upla, podemos definir las funciones

```
fst3 (x, _, _) = x
snd3 (_, y, _) = y
trd3 (_, _, z) = z
```

Ecuaciones de congruencia

Usando el algoritmo extendido de Euclides podemos resolver ecuaciones de congruencia

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

Ecuaciones de congruencia

Usando el algoritmo extendido de Euclides podemos resolver ecuaciones de congruencia

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

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

$$ax + mk = b$$

Ecuaciones de congruencia

Usando el algoritmo extendido de Euclides podemos resolver ecuaciones de congruencia

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

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

$$ax + mk = b$$

Ecuaciones diofánticas

Utilizando la función `emcd` encontrar alguna solución de las siguientes ecuaciones:

1. $29x + 17y = 1$
2. $89x + 23y = 3$
3. $93x + 27y = 3$

Ecuaciones de congruencia

Usando el algoritmo extendido de Euclides podemos resolver ecuaciones de congruencia

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

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

$$ax + mk = b$$

¿Cuándo tiene solución la ecuación?

Ecuaciones de congruencia

Usando el algoritmo extendido de Euclides podemos resolver ecuaciones de congruencia

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

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

$$ax + mk = b$$

¿Cuándo tiene solución la ecuación?
Tiene solución si y solo si $(a : m)$ divide a b.

Ecuaciones de congruencia

Usando el algoritmo extendido de Euclides podemos resolver ecuaciones de congruencia

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

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

$$ax + mk = b$$

¿Cuándo tiene solución la ecuación?
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 cómo resolver una ecuación de congruencia.
Ejemplo: $35x \equiv 15 \pmod{20}$

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 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)

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 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 $7x + 4y = 3$.

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 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 $7 \cdot (-1) + 4 \cdot 2 = 1$.

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

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 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 (b, m) (con $m > 0$) representa todos los $x \in \mathbb{Z}$ con $x \equiv b \pmod{m}$.

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)`