Ampliació d'Algorísmia

1 Problema 3 - Enunciado

Let *a,b* and c be natural numbers. Show that the linear equation

$$ax + by = c$$

has integer solutions x and y if and only if *gcd(a,b)* divides c, and that it has either zero or infinitely many integer solutions. Then give a polynomial time algorithm that returns a solution (x,y) where the integers $x, y \geq 0$ or reports that no such solution exists

2 Demostración

Hay que demostrar la implicación en ambos sentidos:

$$x, y \in \mathbb{Z} \iff gcd(a,b)|c$$

2.1 <=

Demostracion de porque solo tiene integer solution en caso de gcd
Para probar esto decimos que:

$$if gcd(a,b)|c \implies \exists t \in \mathbb{Z}\, tal\ que\ t * gcd(a,b) = c$$

Por otro lado por definición de gcd...

$$gcd(a,b) = a * i + b * j$$

(identidad de Bézout)
Donde i y j son enteros.
Si la identidad de Bézout la multiplicamos por t en ambos lados...

$$gcd(a,b) * t = a * i * t + b * j * t$$

En vez de *i,j*, llamemosle de forma apropiada....

$$gcd(a,b) * t = a * x_o * t + b * y_o * t$$

Por lo tanto se demuestra que existe una solución entera (x_o, y_o) si el $gcd(a,b)|c$ (todos los parametros son enteros)

2.2 =>

Demostracion de que si x_0, y_0 son enteros el gcd de a y b divide a c. Esto es directamente la Id. de Bézout:

$$gcd(a, b) = a * x_0 + b * y_0$$

(identidad de Bézout)

2.3 Prueba de la Id. de Bézout

Partimos de todas las combinaciones lineales de dos números a y b sea esta de la forma ax + by, dicho esto se coge el mínimo elemento positivo llamemoslo d.

Sea $S = ax + by | x, y \in \mathbb{Z}$

y d el menor de estos numeros

Se demuestra que $a|d$, ya que tanto a como d estan en el conjunto S.
(a esta dentro ya que a*1 + b*0 = a)

b sigue el mismo razonamiento,es decir se cumple que $b|d$.

Sea *d'* otro divisor común de a y b, este divide a todo el conjunto S entre ellos al elemento d, por tanto d' ≤ d, ergo d = MCD.

3 Algoritmo:

Para esto podemos utilizar el algoritmo de Euclides (tiempo polinómico), este nos sirve para calcular el gcd entre dos numeros (especificar coste).
Nosotros pondremos la versión de divisiones:

Nota: hay que asegurar que $a \geq b$

```
def gcd(a, b)
    if a % b == 0
        return b
    else
        return gcd(b, a % b)
```

Coste: *O(log (a + b))*
Nota: esto se demuestra con la sucesion de fibonacci, asi que como el objetivo es que sea polinómico tambien se puede hacer con la versión de restas (más facil de justificar) y de igual forma polinómico.

Nota: hay que asegurar que $a \geq b$

```
def gcd(a, b)
    if a == b
        return a
    if a > b
        gcd(a - b, b)
    else
        gcd(a, b - a)
```

Coste: *O(a + b)*

Una vez dicho esto y como hemos demostrado arriba, si este numero divide a c (una operacion O(1)).

Podemos afirmar que la solucion existe y es de la forma:

$$x = x_0 + \frac{b}{d} * t$$

$$y = y_0 + \frac{a}{d} * t$$

Donde $d = gcd(a,b)$ y $t = \{0, \pm1, \pm2...\}$