IIC2283 — Diseño y Análisis de Algoritmos
Ayudantía 9

Problema 1

Parte 1 Proponga una definición inductiva razonable para el MCD de un conjunto de enteros no vacío.

Solución Se considera un conjunto $\{d_1, \ldots, d_N\}$.

- Si $N = 1$, entonces $\text{MCD}(d_1) = d_1$.
- Si $N = 2$, entonces la definición estándar.
- Si $N = t + 1 > 2$, entonces $\text{MCD}(d_1, \ldots, d_N) = \text{MCD}(d_1, \text{MCD}(d_2, \ldots, d_N))$

Parte 2 Dados enteros $\{d_1, \ldots, d_N\}$, demuestre que existen enteros $x_1, \ldots, x_N$ tales que:

$$\text{MCD}(d_1, \ldots, d_N) = d_1 \cdot x_1 + \cdots + d_N \cdot x_N$$

Solución Por inducción en N, siguiendo la definición anterior. Si $N = 1$ es trivial. Si $N = 2$, corresponde a la identidad de Bezout. Si $N > 2$, entonces por definición:

$$\text{MCD}(d_1, \ldots, d_N) = \text{MCD}(d_1, \text{MCD}(d_2, \ldots, d_N))$$

Luego, por la identidad de Bezout, existen x_1, t tales que

$$\begin{aligned}\text{MCD}(d_1, \ldots, d_N) &= x_1 d_1 + t\text{MCD}(d_2, \ldots, d_N) \\ &= x_1 d_1 + t(y_2 d_2 + \cdots + y_N d_N) \\ &= x_1 d_1 + x_2 d_2 + \cdots + x_N d_N,\end{aligned}$$

si se define $x_i = t y_i$.

Parte 3 Sea $J \subseteq \mathbb{Z}$ tal que para todo $x \in J$, y para todo $z \in \mathbb{Z}$ se cumple que $xz \in J$, y además que si $a, b \in J$, entonces $a + b \in J$. Demuestre que existe $a \in \mathbb{Z}$ tal que:

$$J = \{at | t \in \mathbb{Z}\}$$

Para esto puede ser útil considerar dos casos: $J = \{0\}, J \neq \{0\}$.

Solución Si $J = \{0\}$, entonces es directo. En otro caso, por principio de buen orden J contiene un menor elemento positivo. Si este se denota por d, entonces se cumple que $\{dt | t \in \mathbb{Z}\} \subseteq J$ por definición, y además si $b \in J$, entonces existen q, r tales que $b = qd + r$, donde $0 \leq r < d$. Por otra parte, como b y d pertenecen a J, entonces $r = b - qd$ también, y entonces, por minimalidad de d, $r = 0$. Esto implica que d divide a b, y por lo tanto que $J \subseteq \{dt | t \in \mathbb{Z}\}$.

Parte 4 Dados enteros positivos $\{d_1, \ldots, d_N\}$, se define el conjunto:

$$(d_1, \ldots, d_N) \doteq \{d_1 \cdot x_1 + \cdots + d_N \cdot x_N | x_1 \cdots x_N \in \mathbb{Z}\}$$

Desarrolle y analice un algoritmo que dado un entero X, y enteros $\{d_1, \ldots, d_N\}$, determine si X pertenece o no a $(d_1, \ldots, d_N)$.

Idea de solución Calcular $g = MCD(d_1, \ldots, d_N)$, y verificar si $g|X$. Por la parte 2, $g \in (d_1, \ldots, d_N)$, y por otra parte, para cualquier elemento positivo a en el conjunto, se tiene que $g|a$, por lo tanto g es el menor entero positivo, y por lo tanto $(d_1, \ldots, d_N) = \{gt | t \in \mathbb{Z}\}$.

Problema 2

Se tiene el siguiente algoritmo:

```
function ALGORITMO(a, b)
    a' ← a
    b' ← b
    while a' ≠ 0 and b' ≠ 0 do
        if 2|a' then
            a' ← a'/2
        else if 2|b' then
            b' ← b'/2
        else
            [a', b'] ← [(a' + b')/2, (a' − b')/2]
        end if
    end while
    return a' + b'
end function
```

Parte 1 Demuestre que si a y b no son ambos pares, entonces el algoritmo termina.

Idea de solución Se considera $f(a', b') = (a')^2 + (b')^2$. En cada iteración el valor de la función decrece, y además siempre es mayor o igual a 0. Luego, se tiene que eventualmente uno de a', b' será igual a 0, y por lo tanto que el algoritmo termina.

Parte 2 Determine que representa ALGORITMO(a, b) cuando a o b es impar, y extienda el algoritmo para resolver el caso general.

Idea de solución El algoritmo calcula el MCD entre los números. Para demostrar esto, se define g^i como el MCD entre a' y b' antes de la iteración i. Luego, se demuestra que $g^i = g^{i+1}$ para todo i. Para esto se consideran los tres casos posibles. Si a' es par, entonces $a' = 2n$, y $b' = 2m + 1$. Como g^i divide a $2m + 1$, entonces es impar, por lo tanto también divide a n, y $g^i = \text{MCD}(n, 2m + 1) = g^{i+1}$. El caso con b' par es análogo. En el tercer caso, si ambos son impares, entonces se demuestra que $g^i|(a' + b')/2$, $g^i|(a' - b')/2$, $g^{i+1}|a'$ y $g^{i+1}|b'$.

Como g^i divide a a' y b', entonces $g^i|(a' + b')$. Por otra parte, como los dos son impares, entonces $g^i|(2n + 1 + 2m + 1)$, y por lo tanto $g^i|2(n + m + 1)$. Luego, $g^i|(a' + b')/2$. El caso con $(a' - b')/2$ es análogo.

En el otro sentido, como g^{i+1} divide a $(a' + b')/2$ y a $(a' - b')/2$, entonces divide a la suma y a la diferencia. Estos son respectivamente a' y b'.

Finalmente, como $g^i|g^{i+1}$ y $g^{i+1}|g^i$, entonces son iguales (asumiendo que son positivos).

Problema 3

En este problema se asume que a y b son enteros positivos.

Parte 2 Demuestre que existen enteros i, j tales que $a^i = b^j$ si y solo si existen r, s, t tales que $a = r^s$ y $b = r^t$.

Idea de solución Se demuestran ambas direcciones. Si $a = r^s$ y $b = r^t$, entonces $a^t = b^s$.

En el otro sentido, se consideran los primos $p_1, \ldots, p_m$, que dividan al menos a a o b. Con ellos, se pueden expresar $a = p_1^{a_1} \cdots p_m^{a_m}$, y $b = p_1^{b_1} \cdots p_m^{b_m}$, donde se observa que para todo k entre 1 y m, $ia_k = jb_k$. Luego, se define $g = MCM(i, j)$, y $r = a^{i/g}$. r es un número entero, por que $a^{i/g} = (p_1^{a_1} \cdots p_m^{a_m})^{i/g}$, y entonces para todo k entre 1 y m, $p_k^{a_k i/j}$ es un entero. Esto se obtiene de que $i | a_k i$, y además $j | a_k i = b_k j$. Luego $a_k i$ es un múltiplo común de i y j, y por lo tanto $g | a_k i$.

Finalmente se tiene que $a = r^{g/i}$, y $b = r^{g/j}$. Esta última igualdad se obtiene de que $a^i = r^g = b^j$.

Parte 3 Proponga un algoritmo que dados dos enteros a y b, determine si existen i, j tales que $a^i = b^j$.

Idea de solución Si existen esos enteros, entonces existen r, s, t tales que $a = r^s$ y $b = r^t$. Luego, asumiendo que $a > b$, entonces $b | a$, y $a/b = r^{s-t}$. Considerando lo anterior, se construye un algoritmo recursivo:

```
function EQ(a, b)                                   ▷ Se asume a > b
    if a = b or b = 1 then
        return true
    end if
    if b ∤ a then
        return false
    else if a ≥ b^2 then
        return EQ(a/b,b)
    else
        return EQ(b,a/b)
    end if
end function
```

Si existen i, j, entonces en cada iteración se reduce un exponente de r hasta que uno llegue a 1. Si no existen i, j, entonces no se cumple que a y b sean potencias de r, y por lo tanto en algún momento $b \nmid a$.