Algoritmos en álgebra lineal (2do cuatrimestre de 2005)
Hoja de ejercicios n° 2

Complejidad de algoritmos / Transformada de Fourier rápida

Ejercicio 1. ◁ Sean $a, b \in \mathbb{N}$ dos números de longitud binaria $\tau(a), \tau(b) \leq \tau$.

1. Estimar en $O(\ell^2)$ la complejidad del algoritmo de *multiplicación larga* (el que aprendimos en la escuela) para el cálculo de $a \cdot b$.

2. Mostrar que la complejidad de la división con resto

$$a = q\,b + r \quad , \quad 0 \leq r \leq b - 1$$

via el algoritmo standard, es también $O(\tau^2)$. Deducir un algoritmo para el cálculo del mcd(a, b) de complejidad $O(\tau^3)$.

▷

Ejercicio 2. ◁ **Cómo multiplicar sin saber las tablas.** En base 2, la multiplicación larga se reduce a operaciones triviales: por cada $''1''$ en multiplicador, hay que desplazar el multiplicando un cierto espacio, y luego sumar las cifras desplazadas. Este algoritmo se puede implementar en cualquier base sin necesidad de escribir los números en binario: sólo es necesario saber multiplicar y dividir por 2!

En un papel, anote el resultado de dividir por 2 el multiplicador (sin tener en cuenta la parte fraccionaria) y de multiplicar por 2 el multiplicando, iterativamente hasta llegar al 1. Anule las líneas en las cuales el primero de los números es par y luego sume los números que le quedan en la segunda columna para obtener el resultado.

Ejemplo: $197 \times 351 = 69.147$.

197	351	
98	702	NO
49	1.404	
24	2.808	NO
12	5.616	NO
6	11.232	NO
3	22.464	
1	44.928	
	69.147	

1. Demuestre que este algoritmo calcula $a \cdot b$ correctamente.

2. Estime su complejidad en $O(\tau^2)$ para números $a, b \in \mathbb{N}$ de longitud τ dados en base 10.

▷

Ejercicio 3. ◁ **Multiplicación de Karatsuba.** Sean $a, b \in \mathbb{N}$ dos números de $\leq \tau$ bits y supongamos que τ es par e igual a 2σ. Escribimos

$$a = a_1 2^\sigma + a_0 \quad , \quad b = b_1 2^\sigma + b_0$$

con números a_i, b_i de $\leq \sigma$ dígitos. Para calcular el producto

$$a \cdot b = a_1 b_1 2^{2\sigma} + (a_1 b_0 + a_0 b_1) 2^{\sigma} + a_0 b_0;$$

alcanza con determinar $a_1 b_1$, $a_1 b_0 + a_0 b_1$ y $a_0 b_0$. La clave del algoritmo de Karatsuba es la observación de que esto se puede hacer con 3 multiplicaciones en lugar de 4: si hacemos

$$A := a_1 b_1 \quad , \quad B := a_0 b_0 \quad , \quad C = (a_1 + a_0)(b_0 + b_1)$$

entonces

$$a_1 b_0 + a_0 b_1 = C - A - B.$$

Para calcular estas tres multiplicaciones A, B, C empleamos el mismo procedimiento, y así hasta que los números son suficientemente pequeños (por ejemplo, de una o dos cifras) para calcular las multiplicaciones directemente. Una vez calculados todos los productos, los tenemos que sumar entre sí, lo cual tiene una complejidad $O(\tau)$.

1. Sea $\mathcal{C}_K$ denota la complejidad del algoritmo de Karatsuba; mostrar que
$$\mathcal{C}_K(\tau) \leq 3\mathcal{C}_K(\tau/2) + O(\tau).$$
2. Deducir que
$$\mathcal{C}_K(\tau) = O(\tau^{\log_2(3)}) = O(\tau^{1,585}).$$
3. Explicite las constantes escondidas en la notacion $''O''$, y determine un τ_0 a partir del cual el algoritmo de Karatsuba es mas rápido que la multiplicación larga.

▷

Ejercicio 4. ◁ Chusmee la página
`http://www.swox.com/gmp/manual/Multiplication-Algorithms.html`
para enterarse de los algoritmos de multiplicación utilizados por la librería GMP de múltiple precisión. ▷

Ejercicio 5. ◁ Sea $n \geq 1$ y consideremos el anillo de restos módulo n:

$$\mathbb{Z}_n := \mathbb{Z}/(n\mathbb{Z}) = \{\overline{0}, \overline{1}, \ldots, \overline{n-1}\}.$$

Usando versiones "módulo n" de los algoritmos clásicos, mostrar que la suma y la multiplicación de $\mathbb{Z}_n$ tienen complejidad $O(n)$ y $O(n^2)$ respectivamente. ▷

Ejercicio 6. ◁ Calcule la TFD sobre $\mathbb{C}$ de los vectores

1. $(0, 1, 2, 3)$;
2. $(1, 2, 0, 2, 0, 0, 0, 1)$.

▷

Ejercicio 7. ◁ El numero complejo $\omega = \exp(2i\pi/8)$ es una raiz primitiva de 1 de orden 8. Sean

$$f = 5x^3 + 3x^2 - 4x + 3 \quad , \quad g = 2x^3 - 5x^2 + 7x - 2 \in \mathbb{C}[x];$$

calcular la multiplicacion $f \cdot g$ *via* el algoritmo de convolucion.

Sugerencia: se permite multiplicar polinomios de grado 1 por el método clasico. Use ω simbolicamente, con el hecho de que $\omega^4 = -1$.

▷

Ejercicio 8. ◁ Calcule $(10101110011110)_2$ modulo 2^5+1. ▷

Ejercicio 9. ◁ Sea $N=2^n$; el objetivo de este ejercicio es de mostrar que

$$2^M-1$$

no es un divisor de cero en el anillo $\mathbb{Z}_{2^N+1}$, para todo $1 \leq M \leq 2N-1$, o equivalementemente

$$\mathrm{mcd}(2^M-1, 2^N+1) = 1 \quad , \quad 1 \leq M \leq 2N-1. \tag{1}$$

Junto con el hecho (demostrado en clase) de que $2N$ es el menor entero $Q \geq 1$ tal que $2^Q \equiv 1 \pmod{2^N+1}$ muestra que 2 es una *raíz principal* $2N$-ésima de 1 en el anillo $\mathbb{Z}_{2^N+1}$.

1. Sea $a \geq 2$ y $k, \ell \geq 1$, mostrar que

   $$\mathrm{mcd}(a^k-1, a^\ell-1) = a^{\mathrm{mcd}(k,\ell)}-1.$$

   Sugerencia: usar que si consideramos la división con resto $\ell = c \cdot k + r$ con $0 \leq r \leq k-1$, entonces

   $$a^\ell \equiv a^r \pmod{a^k-1}.$$

2. Deducir de lo anterior que para todo $k, \ell \geq 1$

   $$\mathrm{mcd}(a^k-1, a^\ell+1) | a^{\mathrm{mcd}(k,2\ell)}-1.$$

3. Sea $N=2^n$ y $1 \leq M \leq 2N-1$, usando lo anterior mostrar que

   $$\mathrm{mcd}(2^M-1, 2^N+1) | 2^{\mathrm{mcd}(M,N)}-1 | 2^N-1.$$

   Deducir que $\mathrm{mcd}(2^M-1, 2^N+1) = 1$.

4. Encontrar una demostración de (1) más corta y sencilla, o bien probar que la anterior es la demostración más sencilla posible.

▷