



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN  
IIC2283 - DISEÑO Y ANÁLISIS DE ALGORITMOS

## Ayudantía 2

28 de agosto de 2020

Profesor Marcelo Arenas

Bernardo Barías y Ricardo Rodríguez

---

### Pregunta 1 - Inducción Constructiva

Sea

$$T(n) = \begin{cases} e_1 & \text{si } n \leq 3 \\ T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + T(\lceil n/2 \rceil + 1) + e_2 n & \text{si } n > 3 \end{cases}$$

Demuestre usando inducción constructiva que  $T(n) \in O(n^{\log 3})$ .

### Pregunta 2 - Programación Dinámica

Dadas dos palabras  $u = a_1 a_2 \dots a_n$ ,  $w = b_1 b_2 \dots b_n$  se define el problema de la subsucesión común máxima (conocido como longest common subsequence) como el problema de encontrar la subsucesión de caracteres (no necesariamente contiguos)  $x = c_1 \dots c_k$ , de mayor largo tal que aparezca en ambas palabras.

Esta no es única necesariamente, por ejemplo, las subsucesiones comunes máximas entre  $ABCD$  y  $ACBAD$  son  $ABD$  y  $ACD$ . Mediante programación dinámica diseñe un algoritmo que encuentre una subsucesión común máxima y cuya complejidad sea  $O(|u||w|)$

### Pregunta 3 - Programación Dinámica

Sean  $A_1, \dots, A_n$  matrices donde la  $i$ -ésima matriz es de orden  $p_{i-1} \times p_i$ . Si queremos calcular el producto  $A_1 A_2 \dots A_n$ , dado que la multiplicación de matrices es asociativa, tenemos muchas opciones para colocar paréntesis de forma que exprese el producto.

Si  $n = 4$ , por ejemplo, algunas opciones son  $(A_1 A_2 A_3) A_4 = (A_1 A_2)(A_3 A_4) = A_1(A_2 A_3 A_4) = A_1 A_2 A_3 A_4$ . Recordemos que al multiplicar matrices de dimensiones  $p \times q$  y  $q \times r$  realizamos  $pqr$  multiplicaciones escalares. Nuestro objetivo es encontrar la forma de poner paréntesis en el producto de manera tal que minimice la cantidad de multiplicaciones escalares a realizar.

- (**Propuesto**) Demuestre que la cantidad de maneras de asociar el producto del enunciado es  $\Omega(2^n)$
- Diseñe un algoritmo, utilizando la técnica de programación dinámica, que encuentre la asociación del producto de matrices que minimice el número de multiplicaciones escalares
- Pruebe que el algoritmo de la parte b) tiene complejidad  $O(n^3)$

## Ejercicios propuestos

a) Considere la siguiente ecuación de recurrencia:

$$T(n) = \begin{cases} n & \text{si } n \leq 3 \\ T(\lfloor n/3 \rfloor) + T(\lfloor n/4 \rfloor) + n & \text{si } n > 3 \end{cases}$$

Encuentre una función  $f : \mathbb{N} \rightarrow \mathbb{N}$  tal que  $T(n) \in \Theta(f(n))$  y demuestre que se cumple la propiedad.

b) Considere la siguiente ecuación de recurrencia:

$$T(n) = \begin{cases} 1 & \text{si } n = 1 \\ T(n-1) + \log(T(n-1)) + n & \text{si } n > 1 \end{cases}$$

Encuentre una constante  $k \in \mathbb{N}$  tal que  $T(n) \in \Theta(n^k)$  y demuestre que se cumple la propiedad.