PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACION

Diseño y Análisis de Algoritmos - IIC2283
Interrogación 1

1. Recuerde que las funciones que representan el tiempo de ejecución de un algoritmo son de la forma $f : \mathbb{N} \to \mathbb{R}_0^+$. Responda las siguientes preguntas sobre este tipo de funciones.

   a) [0.7 puntos] Demuestre que $f(n) + g(n) \in \Theta(\text{máx}\{f(n), g(n)\})$

   b) [0.8 puntos] Demuestre que $3^n \notin O(2^n)$

2. [1.5 puntos] Considere la siguiente ecuación de recurrencia:

$$T(n) \quad = \quad \begin{cases} 1 & n = 1 \\ 2 \cdot T(\lfloor \frac{n}{2} \rfloor) + \lfloor n \cdot \log_2(n) \rfloor & n > 1 \end{cases}$$

   Demuestre que $T(n) \in O(n \cdot (\log_2(n))^2)$.

3. [1.5 puntos] Dadas dos matrices $A_{m\times n}$ y $B_{n\times r}$ de números enteros, el algoritmo usual para calcular $A \cdot B$ debe realizar $m \cdot n \cdot r$ multiplicaciones de números enteros. Además, dadas tres de estas matrices $A_{m\times n}$, $B_{n\times r}$ y $C_{r\times s}$, podemos calcular su multiplicación como $(A \cdot B) \cdot C$ realizando $m \cdot n \cdot r + m \cdot r \cdot s$ multiplicaciones de números enteros, o como $A \cdot (B \cdot C)$ realizando $m \cdot n \cdot s + n \cdot r \cdot s$ multiplicaciones de números enteros.

   Dada una matriz A, usamos filas(A) y col(A) para denotar el número de filas y columnas de A. En esta pregunta, queremos utilizar programación dinámica para construir un algoritmo **NumMult** que tenga como entrada una secuencia de matrices A_1, A_2, $\ldots$, A_ℓ tal que $\ell \geq 1$ y $\text{col}(A_i) = \text{filas}(A_{i+1})$ para cada $i \in \{1, \ldots, \ell - 1\}$, y retorne el número mínimo de multiplicaciones de números enteros que debe realizar el algoritmo usual para calcular la multiplicación de la secuencia de matrices. Para este objetivo, utilizamos la siguiente definición recursiva para $\textbf{NumMult}(A_1, \ldots, A_\ell)$:

$$\textbf{NumMult}(A_1, \ldots, A_\ell) \quad = \quad \begin{cases} 0 & \ell = 1 \\ \displaystyle\min_{1 \leq i \leq \ell-1} \big\{ \textbf{NumMult}(A_1, \ldots, A_i) + \\ \qquad \textbf{NumMult}(A_{i+1}, \ldots, A_\ell) + & \ell > 1 \\ \qquad \text{filas}(A_1) \cdot \text{col}(A_i) \cdot \text{col}(A_\ell) \big\} \end{cases}$$

En esta pregunta usted debe explicar por qué se puede aplicar programación dinámica en este caso, y por qué la recursión anterior es correcta. Además, debe implementar un algoritmo que utilice esta recursión para calcular **NumMult** en tiempo polinomial, suponiendo que la operación básica a contar es la multiplicación de números enteros.

4. [1.5 puntos] Dado un polinomio

$$p(x) \quad = \quad \sum_{i=0}^{n-1} a_i x^i$$

representando por la tupla de coeficientes $\bar{a} = (a_0, \ldots, a_{n-1})$, defina la transformada alternativa de Fourier de la siguiente forma:

$$\mathbf{TAF}(\bar{a}) \quad = \quad [p(\omega_{2n}^{1}),\, p(\omega_{2n}^{3}),\, \ldots,\, p(\omega_{2n}^{2n-1})].$$

Muestre que las mismas ideas utilizadas para desarrollar el algoritmo **FFT** pueden ser utilizadas para desarrollar un algoritmo para calcular $\mathbf{TAF}(\bar{a})$ en tiempo $O(n \cdot \log(n))$, considerando la suma y multiplicación de números complejos como la operación básica a contar.