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

Diseño y Análisis de Algoritmos - IIC2283
Interrogación 1

El plazo de entrega de la interrogación es el jueves 8 de octubre a las 5pm. Para entregar la interrogación debe utilizar el mismo repositorio de git que usted usó para entregar la tarea 1. La interrogación tiene 4 preguntas; las soluciones de estas preguntas deben ser colocadas en el directorio `I1` de su repositorio con los nombres `p1.ext`, `p2.ext`, `p3.ext` y `p4.ext`, donde `ext` es la extensión correspondiente al tipo de archivos que usted suba en su repositorio (se recomienda utilizar `pdf` o `jpeg`). Las respuestas pueden ser escritas a mano o en la Latex (no se recomienda el uso de Word, pero no está prohibido). Para realizar consultas sobre la interrogación utilice el foro `https://github.com/PUC-IIC2283/2020-IIC2283-Pruebas/issues`.

1. [1.5 puntos] Considere la siguiente ecuación de recurrencia:

$$T(n) \quad = \quad \begin{cases} 1 & n = 0 \\ T(\lfloor \frac{n}{3} \rfloor) + T(\lfloor \frac{2 \cdot n}{3} \rfloor) + n & n > 0 \end{cases}$$

   Utilizando inducción constructiva, demuestre que $T(n) \in O(n \cdot \log_2(n))$.

2. Dada una función $f : \mathbb{N} \to \mathbb{R}_0^+$, defina:

$$o(f) \quad = \quad \{g : \mathbb{N} \to \mathbb{R}_0^+ \mid (\forall c \in \mathbb{R}^+)(\exists n_0 \in \mathbb{N})(\forall n \geq n_0)\,(g(n) \leq c \cdot f(n))\}.$$

   Responda las siguientes preguntas sobre esta notación.

   (a) [0.5 puntos] De ejemplos de funciones $f, g : \mathbb{N} \to \mathbb{R}^+$ tales que $g \in o(f)$, y demuestre que estas funciones satisfacen esta propiedad.

   (b) [1 punto] Sea $f : \mathbb{N} \to \mathbb{R}_0^+$ una función que satisface la condición:

$$(\exists n_0 \in \mathbb{N})(\forall n \geq n_0)\,(f(n) > 0).$$

   Vale decir, la función $f(n)$ es mayor a cero a partir de un cierto número natural n_0. Demuestre que si $g \in o(f)$, entonces $g \in O(f)$ y $f \notin O(g)$.

3. [1.5 puntos] Sea Σ un alfabeto. Recuerde que dado $w \in \Sigma^*$ con $w = a_1 \cdots a_n$ y $n \geq 0$, en clases vimos las siguientes operaciones para transformar w: dado $i \in \{1, \ldots, n\}$ y $b \in \Sigma$, tenemos que

$$\begin{aligned} \texttt{eliminar}(w, i) &= a_1 \cdots a_{i-1} a_{i+1} \cdots a_n \\ \texttt{agregar}(w, i, b) &= a_1 \cdots a_i b a_{i+1} \cdots a_n \\ \texttt{cambiar}(w, i, b) &= a_1 \cdots a_{i-1} b a_{i+1} \cdots a_n \end{aligned}$$

y tenemos que $\texttt{agregar}(w, 0, b) = bw$. Además, dadas dos palabras $w_1, w_2 \in \Sigma^*$, definimos la distancia entre w_1 y w_2 como la menor cantidad de operaciones que es necesario realizar para transformar w_1 en w_2, lo cual es denotado como $\texttt{ed}(w_1, w_2)$. Finalmente, suponiendo que $w_1 = a_1 \cdots a_n$ y $w_2 = b_1 \cdots b_m$, con $n \geq 0$ y $m \geq 0$, para $i \in \{0, \ldots, n\}$ y $j \in \{0, \ldots, m\}$ definimos $\texttt{ed}(i, j) = \texttt{ed}(a_1 \cdots a_i, b_1 \cdots b_j)$, y obtuvimos:

$$\texttt{ed}(i,j) = \begin{cases} \text{máx}\{i, j\} & i = 0 \text{ o } j = 0 \\ \text{mín}\{1 + \texttt{ed}(i-1, j), 1 + \texttt{ed}(i, j-1), \texttt{dif}(i, j) + \texttt{ed}(i-1, j-1)\} & i > 0 \text{ y } j > 0 \end{cases}$$

donde $\texttt{dif}(i, j)$ es 0 si $a_i = b_j$, y es 1 en caso contrario.

En esta pregunta usted debe implementar un algoritmo **TRANS** que reciba como entrada dos palabras w_1 y w_2, y retorne una lista de operaciones $[o_1, \ldots, o_\ell]$ tales que las operaciones $o_1, \ldots$, o_ℓ aplicadas desde w_1 generan w_2 y $\ell = \texttt{ed}(w_1, w_2)$. Por ejemplo, **TRANS**$(\texttt{casa}, \texttt{asado})$ debe retornar $[\texttt{agregar}(\texttt{casa}, 4, \texttt{d}), \texttt{eliminar}(\texttt{casad}, 1), \texttt{agregar}(\texttt{asad}, 4, \texttt{o})]$. Para implementar el algoritmo **TRANS** utilice el lenguaje para representar algoritmos utilizado en clases. Además usted debe analizar la complejidad de su algoritmo, y debe demostrar que funciona en tiempo polinomial en el tamaño de la entrada $|w_1| + |w_2|$.

4. Recuerde que utilizamos la notación $(a_0, \ldots, a_{\ell-1})$ para representar un polinomio

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

con coeficientes en los números racionales. Además, suponiendo que $\ell \geq 2$ y ℓ es una potencia de 2, utilizamos $\textbf{FFT}(\bar{a})$ para denotar la transformada rápida de Fourier aplicada a $\bar{a}$.

Considere el siguiente algoritmo que recibe como entradas un polinomio representado por la tupla de coeficientes $\bar{a} = (a_0, \ldots, a_{\ell-1})$ y un número natural k, donde $\ell \geq 2$, ℓ **no** es necesariamente una potencia de 2 y k es una potencia de 2.

ALG$(\bar{a},\ k)$
  if $k = 1$ **then return** $\bar{a}$
  else
    $(b_0, \ldots, b_{n-1}) := \textbf{ALG}(\bar{a},\ \frac{k}{2})$
    $m := 2^{\lceil \log_2(2n) \rceil}$
    $(c_0, \ldots, c_{n-1}, c_n, \ldots, c_{m-1}) := (b_0, \ldots, b_{n-1}, 0, \ldots, 0)$
    $[y_0, \ldots, y_{m-1}] := \textbf{FFT}(\bar{c})$ /* $\bar{c} = (c_0, \ldots, c_{n-1}, c_n, \ldots, c_{m-1})$ */
    $\bar{z} := (y_0^2, y_{m-1}^2, \ldots, y_1^2)$
    $[d_0, \ldots, d_{m-1}] := \frac{1}{m} \cdot \textbf{FFT}(\bar{z})$
    return $(d_0, \ldots, d_{2n-2})$

Responda las siguientes preguntas sobre este algoritmo.

(a) [0.7 puntos] Indique qué retorna la llamada **ALG**$(\bar{a},\ k)$, y demuestre formalmente que este es el caso (por ejemplo, utilizando inducción en k).

(b) [0.8 puntos] Considerando a la suma y multiplicación de números complejos como las operaciones a contar, encuentre una función $f : \mathbb{N} \times \mathbb{N} \to \mathbb{R}_0^+$ tal que el número de operaciones realizadas por **ALG**$(\bar{a}, k)$ esté acotado superiormente por $f(\ell, k)$ (recuerde que $\bar{a} = (a_0, \ldots, a_{\ell-1})$). Al igual que en la pregunta anterior, usted debe demostrar formalmente que $f(\ell, k)$ cumple lo pedido.