



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

Diseño y Análisis de Algoritmos - IIC2283

Interrogación 1

Tiempo: 2.5 horas

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

$$T(n) = \begin{cases} 0 & n = 1 \\ a \cdot T(\lfloor \frac{n}{2} \rfloor) + b \cdot T(\lceil \frac{n}{2} \rceil) + c \cdot n^k & n > 1 \end{cases}$$

donde $a, b, k \in \mathbb{N}$, $a, b > 0$ y $c \in \mathbb{R}^+$.

De una clasificación completa del orden de $T(n)$ dependiendo de los valores a , b y k . Esta clasificación debe ser hecha en términos de la notación asintótica Θ .

2. [1 punto] Demuestre que **Quicksort** en el caso promedio es $\Theta(n \cdot \log_2(n))$, considerando como la operación a contar la comparación y suponiendo que la entrada del algoritmo es una lista sin elementos repetidos.
3. [1.5 puntos] Considere el siguiente problema: dadas dos listas ordenadas de menor a mayor $L_1[1 \dots n]$ y $L_2[1 \dots n]$, construir una lista ordenada de menor a mayor $L[1 \dots 2 \cdot n]$ que resulta de mezclar los elementos de L_1 y L_2 . Demuestre que un algoritmo que resuelve este problema y que está basado en comparaciones de elementos de las listas debe realizar al menos $2 \cdot n - 1$ comparaciones.
4. [2 puntos] Sea $\mathbb{M}$ la clase de matrices cuadradas $A_{n \times n}$ de números enteros tales que:
- para cada $i \in \{1, \dots, n-1\}$ y $j \in \{1, \dots, n\}$, se tiene que $A[i, j] \leq A[i+1, j]$, y
 - para cada $i \in \{1, \dots, n\}$ y $j \in \{1, \dots, n-1\}$, se tiene que $A[i, j] \leq A[i, j+1]$.

Por ejemplo, para las siguientes matrices cuadradas:

$$A = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \\ 3 & 4 & 7 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 2 & 6 \\ 3 & 4 & 7 \end{pmatrix}$$

tenemos que $A \in \mathbb{M}$, mientras que $B \notin \mathbb{M}$ puesto que $B[1, 2] > B[2, 2]$.

En esta pregunta usted debe considerar el siguiente problema: dada una matriz $A \in \mathbb{M}$ de $n \times n$ y un número entero a , determinar una posición (i, j) tal que $A[i, j] = a$ o retornar no si tal posición no existe. En particular, en esta pregunta vamos a estudiar los algoritmos que resuelven este problema considerando como la operación básica a contar el acceso a una posición de la matriz A .

- a) [1 punto] Encuentre una cota inferior para el número de accesos a las posiciones de A que debe realizar un algoritmo que resuelve el problema.
- b) [1 punto] Construya un algoritmo que resuelva el problema.

El puntaje de su pregunta dependerá de que tan cerca esté la cota inferior del tiempo de ejecución del algoritmo (medido en términos del número de accesos a las posiciones de A). Por ejemplo, si su cota inferior es 1 y su algoritmo resuelve el problema utilizando una búsqueda exhaustiva que realiza n^2 accesos a una matriz de orden $n \times n$ entonces su puntaje será 0 en ambas preguntas.

Importante: Es posible dar un algoritmo cuyo número de accesos a las posiciones de A en el peor caso es exactamente la cota inferior. Para tener dos puntos esta pregunta basta obtener una cota inferior $f(n)$ y un algoritmo que en el peor caso es $\Theta(f(n))$. Si además se tiene que el algoritmo en el peor caso realiza exactamente $f(n)$ accesos a las posiciones de la matriz de entrada entonces usted obtendrá 0.5 puntos adicionales en esta pregunta.