



Diseño y Análisis de Algoritmos - IIC2283

Interrogación 1

Tiempo: 2 horas

1. [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.
2. [2 puntos] Considere la siguiente ecuación de recurrencia:

$$T(n) = \begin{cases} c & n = 0 \\ a \cdot T(\lfloor \frac{n}{2} \rfloor) + f(n) & n \geq 1, \end{cases}$$

donde  $c > 0$ ,  $a \geq 1$ , y  $f : \mathbb{N} \rightarrow \mathbb{R}^+$  es una función que satisface la siguiente propiedad:

$$(\exists \varepsilon \in \mathbb{R}^+)(\exists d \in \mathbb{R}^+)(\forall n \geq 1)(f(n) \leq d \cdot n^{\log_2(a) - \varepsilon}).$$

Nótese que esta propiedad implica que  $f(n) \in O(n^{\log_2(a) - \varepsilon})$  para algún  $\varepsilon > 0$ , de lo cual se puede concluir que  $T(n) \in O(n^{\log_2(a)})$  utilizando el Teorema Maestro. En esta pregunta usted debe demostrar que  $T(n) \in O(n^{\log_2(a)})$  utilizando inducción constructiva.

3. [1.5 puntos] Sea  $\mathcal{A}$  un algoritmo que recibe como entrada una matriz de adyacencia  $M$  que representa a un grafo  $G$ , y retorna **true** si el grafo  $G$  es desconexo, y **false** en caso contrario. Demuestre que  $\mathcal{A}$  en el peor caso debe realizar al menos  $\frac{n^2}{4}$  accesos a la matriz  $M$  que recibe como parámetro.
4. [1.5 puntos] Dada una lista  $L$  de largo  $n$  y un número  $k \in \{1, \dots, n\}$ , definimos la  $k$ -rotación de  $L$  como una lista que resulta de mover  $k$  posiciones a la derecha los elementos de  $L$  considerando a  $L$  como una lista circular. Por ejemplo, si  $L = [2, 5, 9, 1, 9, 4]$ , entonces la 1-rotación de  $L$  es la lista  $[4, 2, 5, 9, 1, 9]$ , la 3-rotación de  $L$  es la lista  $[1, 9, 4, 2, 5, 9]$  y la 6-rotación de  $L$  es la lista  $[2, 5, 9, 1, 9, 4]$ .

Una lista  $L$  de largo  $n$  se dice rotación-ordenada si existe una lista ordenada  $L'$  de largo  $n$  y un número  $k \in \{1, \dots, n\}$  tal que  $L$  es la  $k$ -rotación de  $L'$ . En esta pregunta usted debe desarrollar un algoritmo **BuscarRO**( $L, a$ ) que recibe como parámetro una lista  $L$  que es rotación-ordenada y un elemento  $a$ , y verifica si existe una posición  $i$  tal que  $L[i] = a$  (si existe tal posición la debe retornar, si no debe retornar **false**). El algoritmo **BuscarRO** en el peor caso debe ser  $O(\log_2 n)$ , donde  $n$  es el largo de la lista de entrada.