Examen 1

Análisis y Diseño de Algoritmos

28 de Abril del 2020

Ejercicio 1. (6 ptos) Considere el siguiente problema.

Entrada: Un arreglo $A[1..n]$ de números enteros positivos diferentes (no necesariamente ordenado).

Salida: Un índice k tal que $A[1], \ldots, A[k-1] < A[k]$ y $A[k] < A[k+1], \ldots, A[n]$. Si no existe tal índice k, devolver -1. Por ejemplo, si $A = [1, 3, 2, 4, 8, 7, 5, 6]$, el algoritmo debe devolver el índice 3.

(a) (4 puntos) Diseñe un algoritmo de división y conquista que consuma tiempo $\Theta(n)$ en el peor caso. Puede asumir que n es potencia de 2.

(b) (1 puntos) Escriba la recurrencia que define el peor caso en el tiempo de ejecución del algoritmo. Puede asumir que todas las constantes asociadas valen 1.

(c) (1 puntos) Resuelva la recurrencia anterior usando teorema maestro

Ejercicio 2. (7 ptos) Pruebe las siguientes afirmaciones.

(a) $2n^2 - 5n - 8 = \Omega(n^2)$

(b) $\sum_{k=1}^{n} k^{99} = \Theta(n^{100})$

Ejercicio 3. (7 ptos)

(a) (3 puntos)

$$T(n) = \begin{cases} 1 & n = 1 \\ T(n) = 2T(n-1) + 4n - 25 & \text{caso contrário} \end{cases}$$

Pruebe por inducción que $T(n) = \Omega(2^n)$.

(b) (4 puntos) Resuelva la recurrencia por los métodos vistos en clase

$$T(n) = \begin{cases} 1 & n = 1 \\ T(n) = 8T(\lceil n/2 \rceil) + n & \text{caso contrário} \end{cases}$$

Puede suponer que es conocido que $T(n)$ es una función creciente.