

# Examen 1

## Análisis y Diseño de Algoritmos

28 de Abril del 2020

**Ejercicio 1 (3 ptos).** Pruebe que  $n^2 - 100n - 200\sqrt{n} = \Theta(n^2)$ .

**Ejercicio 2 (4 ptos).** Resuelva la recurrencia **de manera explícita y exacta**, usando los métodos vistos en clase

$$T(n) = \begin{cases} 1 & n = 1 \\ T(n) = 2T(\lfloor n/2 \rfloor) + n \lg n & \text{caso contrario} \end{cases}$$

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

**Ejercicio 3 (3 ptos).** Considere la siguiente recurrencia sobre los números naturales

$$T(n) = \begin{cases} 0 & n = 0 \\ 0 & n = 1 \\ T(n) = T(n-1) + T(n-2) + 1 & \text{caso contrario.} \end{cases}$$

Pruebe por inducción que  $T(n) = \Omega((3/2)^n)$ .

**Ejercicio 4 (2 ptos).** Considere el siguiente problema. Entrada: un arreglo  $A[1..n]$  de números, donde  $n$  es una potencia de 3. Salida: el valor de un elemento máximo en  $A$ . Un algoritmo de división y conquista para dicho problema divide el arreglo en tres partes del mismo tamaño. Luego, busca el máximo elemento en cada una de estas partes y compara estos tres máximos encontrados. Escriba el pseudocódigo del algoritmo anterior. Analise el tiempo de ejecución de este algoritmo como fue visto en clase. Escriba una recurrencia para el desempeño de este algoritmo. Indique el crecimiento  $\Theta$  de esta recurrencia **usando teorema maestro**.

**Ejercicio 5 (2 ptos).** Demuestre que, cuando  $n$  es potencia de 2,  $\sum_{i=0}^{\lg n} (n/2^i) = 2n - 1$ .

**Ejercicio 6 (2 ptos).** ¿Cuántos nodos a profundidad  $k$  tiene un árbol binario completo? Pruebe por inducción.

**Ejercicio 7 (4 ptos).** Tareas de evaluación continua