



Primer examen opcional  
FUNDAMENTOS DE ANÁLISIS Y DISEÑO DE ALGORITMOS  
Grupo 80 - Duración: 2 horas

Carlos Andres Delgado S, Ing \*

05 de Junio de 2014

**Nota:** Por favor marcar todas las hojas del examen con su nombre y código de estudiante. Recuerde colocar los procedimientos debido a que tienen un gran valor en la calificación.

## 1. Computación iterativa [34 puntos]

Se requiere un algoritmo que permita calcular una función cuya entrada es un número entero  $n$  y salida es  $\sum_{i=0}^n (i^4 + 2)$

- [4 puntos.] Indique las salidas para las siguientes entradas:  $\{0, 1, 3, 5, 7\}$
- [30 puntos.] Diseñe un algoritmo **utilizando ciclos** que solucione el problema
  - [8 puntos.] Escriba el algoritmo diseñado.
  - [5 puntos.] ¿Cual es la complejidad de su algoritmo para una entrada de tamaño  $n$  en términos de  $O(f(n))$ .
  - [5 puntos.] ¿Como puede representar los estados de su solución?. ¿Cual sería el estado inicial?
  - [4 puntos.] ¿Cual es la transición de estados de su algoritmo?.

- [8 puntos.] ¿Cual es la invariante de ciclo de su algoritmo?.

## 2. Crecimiento de funciones y ecuaciones de recurrencia [16 puntos]

- [8 puntos] Suponiendo que  $f(n)$  está definida demuestre que no existe una función  $g(n)$  tal que cumple  $f(n) = o(g(n))$  y  $f(n) = \omega(g(n))$ .
- [8 puntos] Resuelva la ecuación  $T(n) = 6T(\frac{n}{6}) + 4n^2$  con  $T(1) = 0$  utilizando cualquiera de los métodos vistos en clase.

## 3. Estructuras de datos [50 puntos]

Indicar cuales de las siguientes expresiones son verdaderas y cuales son falsas, justifique su respuesta.

### 1. Estructura Pilas (Stack) y Colas (Queue)

- a) [5 puntos.] Una cola se puede considerar una estructura *FIFO* es decir, que el primer elemento encolado es el primero en ser desencolado.
- b) [5 puntos.] La complejidad de las operaciones **Push** y **Pop** en una pila es  $\Theta(n \log(n))$

---

\* carlos.andres.delgado@correounivalle.edu.co

## 2. Estructura Listas

- a) [5 puntos.] En la lista **simplemente enlazada** la operación  $LIST - INSERT(L, x)$  inserta  $x$  al inicio de la lista
- b) [5 puntos.] En una lista **doblemente enlazada** la complejidad de  $LIST - DELETE(L, x)$  será  $\mathcal{O}(1)$

## 3. Estructura Tablas Hash

- a) [5 puntos.] Un tabla hash sólo es útil cuando el numero de **slots m** es superior o igual al número de llaves a almacenar
- b) [5 puntos.] Si se utiliza una buena función hash en una tabla hash con un número de **slots m** y un universo de llaves de tamaño  $k$  la complejidad de una **búsqueda** (*exitosa o no*) es  $\Theta(n)$

## 4. Estructura Árboles de búsqueda binaria

- a) [5 puntos.] La complejidad de las operaciones en el peor caso es  $\mathcal{O}(n \lg n)$
- b) [5 puntos.] El recorrido inorden del árbol se obtiene una lista ordenada en forma descendente

## 5. Estructura Árboles rojinegros

- a) [5 puntos.] Un nodo rojo puede tener hijos rojos o negros
- b) [5 puntos.] En el peor caso la complejidad de las operaciones en un árbol rojinegro es  $\Theta(n)$

$$\blacksquare \sum_{k=0}^n ar^k = \frac{ar^{(n+1)} - a}{r-1} \text{ Si } r \neq 1$$
$$\blacksquare \sum_{k=0}^n ar^k = (n+1)a \text{ Si } r = 1$$

## Formulas solución método del maestro

Recuerde la forma  $T(n) = aT(\frac{n}{b}) + f(n)$

- Si  $f(n) = \mathcal{O}(n^{\log_b a - \epsilon})$  para algún  $\epsilon > 0$  entonces  $T(n) = \Theta(n^{\log_b a})$
- Si  $f(n) = \Theta(n^{\log_b a})$  entonces  $T(n) = \Theta(\log(n) * n^{\log_b a})$
- Si  $f(n) = \Omega(n^{\log_b a + \epsilon})$  para algún  $\epsilon > 0$  y  $af(\frac{n}{b}) \leq cf(n)$  con  $c < 1$  entonces  $T(n) = \Theta(f(n))$ .

## Ayudas

### Formulas de sumatorias

$$\blacksquare \sum_{k=1}^n c = cn$$
$$\blacksquare \sum_{k=1}^n k = \frac{n(n+1)}{2}$$
$$\blacksquare \sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6}$$