



Primer examen opcional
FUNDAMENTOS DE ANÁLISIS Y DISEÑO DE ALGORITMOS
Grupo 80 - Duración: 1 hora 30 minutos

Carlos Andres Delgado S, Ing *

28 de Mayo 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. Calentamiento [10 puntos]

Dos algoritmos $A1$ y $A2$ para resolver el mismo problema toman n^2 horas y n^3 segundos, respectivamente, para dar una respuesta a entradas de tamaño n . ¿En la práctica, para que tamaño de las entradas escogería usted utilizaría $A1$ y no $A2$?. Justifique claramente su respuesta.

2. Computación iterativa [34 puntos]

Sea $A[1..n]$ un arreglo de n números distintos. Si $i < j$ y $A[i] > A[j]$, entonces el par (i, j) es llamado una inversión de A .

- [2 puntos.] Enumere todas las inversiones del arreglo $\langle 5, 9, 12, 10, 3 \rangle$
- [8 puntos.] Cual arreglo con elementos del conjunto $\{1, 2, \dots, n\}$ tiene el mayor numero de inversiones? ¿Cuántas tiene?.

- [24 puntos.] Diseñe un algoritmo utilizando ciclos que determine el **número de inversiones** en cualquier arreglo de longitud n .

- [6 puntos.] Escriba el algoritmo diseñado.
- [4 puntos.] ¿Cual es la complejidad de su algoritmo para una entrada de tamaño n en términos de $O(f(n))$.
- [4 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?.
- [6 puntos.] ¿Cual es la invariante de ciclo de su algoritmo?.

3. Crecimiento de funciones y ecuaciones de recurrencia [16 puntos]

- [8 puntos] Indique dos funciones $f(n)$ y $g(n)$ tal que se cumpla $f(n) = O(g(n))$ y $f(n)$ no es $\Theta(g(n))$.
- [8 puntos] Resuelva la ecuación $T(n) = 4T(\frac{n}{4}) + 2n + 1$ utilizando cualquiera de los métodos vistos en clase.

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

4. Estructuras de datos [40 puntos]

4.1. Listas, pilas y colas. [23 puntos]

Un palíndromo es una cadena de caracteres que se puede leer igual de izquierda a derecha y viceversa (ignorando los espacios entre las palabras). Escriba una función que reciba una cadena almacenada en una lista simplemente enlazada (un solo carácter por cada nodo), y que retorne verdadero o falso, según si la cadena recibida es o no un palíndromo respectivamente.

NOTA: Sólo serán válidas las soluciones que únicamente utilicen las estructuras listas, pilas y colas o un subconjunto de ellas, al igual que las operaciones vistas en clase para cada una de esas estructuras

4.2. Árboles binarios de búsqueda y árboles rojinegros [17 puntos]

- [7 puntos] ¿Cual es la ventaja que tienen los árboles rojinegros frente a los árboles de búsqueda binaria para garantizar que en el caso de los árboles rojinegros las operaciones en el peor de los casos tengan complejidad $O(n \log(n))$ frente a $O(n)$ del caso de los arboles de búsqueda binaria?.
- [10 puntos] Para el siguiente árbol de búsqueda binaria

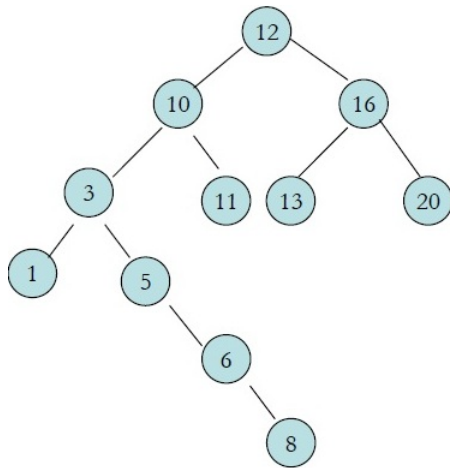


Figura 1: Arbol de ejemplo

- [4 puntos] Realice la operación $TREE - INSERT(4)$. Explique brevemente como se realiza el proceso. Dibuje el árbol resultante.
- [6 puntos] Realice la operación $TREE - DELETE(10)$. Explique brevemente como se realiza el proceso. Dibuje el árbol resultante.

Ayudas

Formulas de sumatorias

- $\sum_{k=1}^n c = cn$
- $\sum_{k=1}^n k = \frac{n(n+1)}{2}$
- $\sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6}$
- $\sum_{k=0}^n ar^k = \frac{ar^{(n+1)} - a}{r-1}$ Si $r \neq 1$
- $\sum_{k=0}^n ar^k = (n+1)a$ Si $r = 1$

Formulas solución método del maestro

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

- Si $f(n) = O(n^{\log_b a - \epsilon})$ 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 $T(n) = \Theta(f(n))$.