



Primer examen parcial  
FUNDAMENTOS DE ANÁLISIS Y DISEÑO DE ALGORITMOS  
Grupo 02

Duración: 2 horas  
Carlos Andres Delgado S, Ing<sup>\*</sup>  
26 de Marzo de 2015

**1. Computación iterativa y complejidad algoritmos [45 puntos]**

Para el siguiente algoritmo:

```
1 Algoritmo(int N)
2 {
3     int i, res;
4     i = -3;
5     res = 0;
6
7     while(i < N){
8
9         res = res + i;
10
11         for(int j=0; j <= (2N); j++)
12         {
13             res = res + 3;
14         }
15         i++;
16     }
17     System.out.println("Resultado=" + Res);
18 }
```

**1.1. Entendimos el problema [8 puntos]**

- (3 puntos) Determine las salidas para las siguientes entradas  $\{-1, 1, 2, 3\}$
- (5 puntos) Para un número  $n \geq -3, n \in \mathbb{N}$  escriba una expresión que permita determinar la salida este algoritmo.

**1.2. Analicemos el algoritmo [37 puntos]**

- (15 puntos) Muestre cuantas veces se ejecuta cada línea del código en términos de  $n$  y dé el total de ejecuciones del algoritmo en términos de  $n$ . Indique la complejidad del algoritmo en términos de  $O(f(n))$ .
- (5 puntos) ¿Cómo puede representar los estados del algoritmo?. ¿Cual es el estado inicial?.
- (6 puntos) ¿Cómo es la transición de estados del algoritmo?.

- (9 puntos) ¿Cual es la invariante de ciclo del algoritmo?.

**2. Crecimiento de funciones [20 puntos]**

- (8 puntos) Demuestre que  $n^3 - n$  es  $O(n^3)$ .
- (12 puntos) Indique si existen funciones  $f(n)$  y  $g(n)$  tales que  $f(n)$  es  $O(g(n))$  y  $g(n)$  es  $\theta(f(n))$ . En caso de existir dé un ejemplo de funciones  $f(n)$  y  $g(n)$

**3. Ecuaciones de recurrencia [20 puntos]**

Para las siguientes preguntas asuma que  $T(2) = 2$ .

- (10 puntos) Con el método de iteración solucione  $T(n) = 2T(\frac{n}{2}) + n^5 + \frac{n}{4}$ . Expresé en forma de sumatorias.
- (10 puntos) Con el método del maestro determine la solución a la siguiente ecuación de recurrencia  $T(n) = 16T(\frac{n}{2}) + n^2$ .

**4. Estructuras de datos [15 puntos]**

- (7 puntos) Se tiene un array denominado como  $S$  inicialmente vacío de tamaño 7 para construir una cola, muestre como se encuentra el array y el valor de la variable  $S.head$  y  $S.tail$  después de ejecutar las siguientes instrucciones

1. ENQUEUE( $S, 2$ )
2. ENQUEUE( $S, 3$ )
3. DEQUEUE( $S$ )
4. ENQUEUE( $S, 7$ )
5. DEQUEUE( $S$ )
6. ENQUEUE( $S, 8$ )
7. DEQUEUE( $S$ )
8. ENQUEUE( $S, -3$ )

- (8 puntos) Para el siguiente árbol binario de búsqueda: Aplique la operación DELETE(20). Muestre el procedimiento que realizó y el árbol resultante.

<sup>\*</sup>carlos.andres.delgado@correounivalle.edu.co

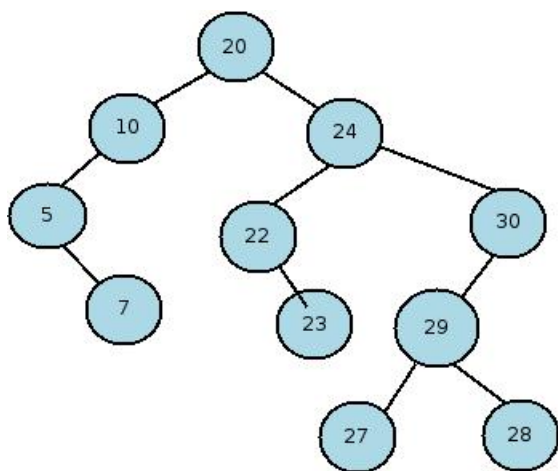


Figura 1: Arbol Binario de Búsqueda

## 5. Punto extra [+0.5 parcial]

Construya un árbol rojinegro insertando los siguientes elementos en este orden:  $\{2, 10, 12, -3, 1, 3, 19\}$  muestre el procedimiento al insertar cada elemento.

## 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  $\epsilon > 0$ ,  $c < 1$  y  $af(\frac{n}{b}) \leq cf(n)$  entonces  $T(n) = \Theta(f(n))$ .

Recuerde colocar los procedimientos realizados, ya que estos tienen un gran valor en la calificación de cada punto.

## Inserción árboles rojinegros

- Caso 1: x(rojo) es un hijo de un padre rojo y el tío de x es rojo, se pintan de negro padre y tío de x, el abuelo de x queda entonces de rojo. x es ahora el abuelo de x
- Caso 2: x(rojo) es un hijo derecho de un padre rojo y el tío de x, y, es ahora negro. Se rota a la izquierda p[x]. x ahora es el padre de x
- Caso 3: x(rojo) es el hijo izquierdo de un padre rojo y el tío es negro. Se cambian los colores de p[x] y p[p[x]]. Se rota a la derecha p[x].

Figura 2: Rotaciones rojinegros

