

# Primer examen parcial

## Fundamentos de análisis y diseño de algoritmos

Carlos Andres Delgado S, Msc  
carlos.andres.delgado@correounivalle.edu.co

27 de octubre 2022

### 1. Complejidad computacional e iterativa [50 puntos]

Para el siguiente algoritmo:

```

1 //Suponga n par
2 int algoritmo1(int n){
3     int a = 0;
4     int i = 0;
5     while(i <= n){
6         i = i + 1;
7         int j = 0;
8         int t = 0;
9         while(j <= 2*n){
10            t = t + 2;
11            j = j + 2;
12        }
13        a = a + t;
14    }
15    return a;
16 }

```

①  
 $n=2$   
 0, 2, 4, 6  $\frac{4+2}{2} + 1 = 4 \checkmark$   
 $n=4$   
 0, 2, 4, 6, 8, 10  $\frac{8+2}{2} + 1 = 5 \checkmark$

$O(n^2)$

- (20 puntos) Calcule la complejidad total del algoritmo. Muestre el procedimiento línea por línea. Finalmente, indique la complejidad total en términos de  $O(f(n))$  siendo  $f(n)$  la cota más pequeña posible.
- (15 puntos) Para el ciclo interno
  - (5 puntos) Forma de estado, estado inicial, transformación de estados y estado final
  - (10 puntos) Invariante de ciclo y su demostración
- (15 puntos) Para el ciclo externo
  - (5 puntos) Forma de estado, estado inicial, transformación de estados y estado final
  - (10 puntos) Invariante de ciclo y su demostración

Interna

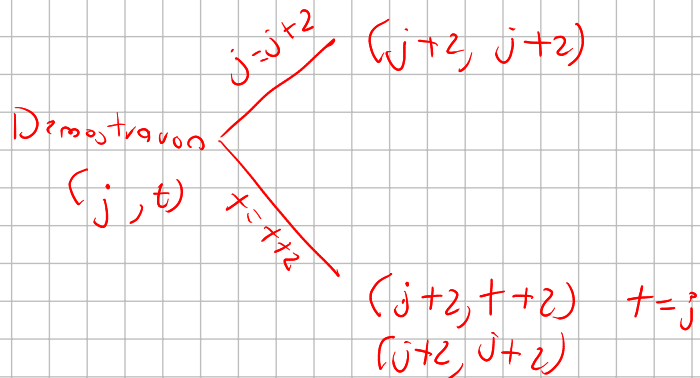
$(j, t)$

$(0, 0)$

$(j, t) \rightarrow (j+2, t+2)$  Transformation

$(0, 0) \Rightarrow (2, 2) \rightarrow (4, 4) \rightarrow \dots \rightarrow (2n, 2n) \rightarrow (2n+2, 2n+2)$

Invariante  $(j, j)$



Externa

$q = a + 2n + 2$

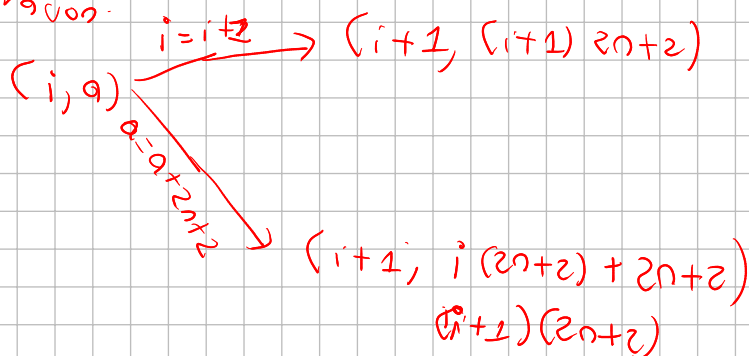
Estado  $(i, q)$

Transformation

$(0, 0) \rightarrow (1, 0 + 2n + 2) \rightarrow (2, 0 + 2(2n + 2))$

Invariante  $(i, i(2n+2))$

Demonstracion



## 2. Relaciones de recurrencia [20 puntos]

Dada la siguiente R.R

$$T(n) = 3T\left(\frac{n}{3}\right) + \frac{n}{3}, T(1) = O(1)$$

$$T(n) = 3^i T\left(\frac{n}{3^i}\right) + 3^i \frac{n}{3^i} + \frac{3^{i-2}}{3^{i-2}} \dots$$

$$T(n) = 3^{\log_3(n)} T(1) + \sum_{i=0}^{\log_3(n)-1} \frac{n}{3}$$

$$T(n) = n + \log(n) \frac{n}{3}$$

$$O(n \log n)$$

- (15 puntos) Resuelva la RR con arboles o el método de expansión.
- (5 puntos) Solucione con el método del maestro y compare la cota con la solución obtenida en el punto 1, explique si estas coinciden.

## 3. Estructuras de datos [30 puntos]

Dada un lista enlazada bidimensional (matriz) estime la complejidad del siguiente algoritmo:

```

1 /*
2  *random retorna un número aleatorio entre 0 y 1
3  *ListaEnlazada es una clase que nos provee una lista enlazada
4  */
5 int algoritmo2(int n){
6     ListaEnlazada matriz = new ListaEnlazada();
7     //Creación de la matriz
8     for(int i=0; i<n; i++){
9         matriz.insert(new ListaEnlazada());
10        for(int j=0; j<n; j++){
11            matriz[i].insert(random());
12        }
13    }
14    //Recorrido de la matriz
15    for(int i=0; i<n; i++){
16        for(int j=0; j<n; j++){
17            suma += matriz[i][j];
18        }
19    }
20 }
21
22 return suma;
23 }

```

Doble	Simple
1	1
n+1	n+1
n	1+2+3... = $\frac{n(n+1)}{2}$
$n(n+1)$	$n(n+1)$
$n^2$	$n^2 \left( \frac{n(n+1)}{2} \right) \leftarrow (n^4)$
$n+1$	$n+1$
$n(n+1)$	$n(n+1)$
$n \left( \frac{n(n+1)}{2} \right)$	$n \left( \frac{n(n+1)}{2} \right)$
1	

1+2+3+4+...+n  $\leftarrow$  Cota inferior

**Pista** Una lista enlazada bidimensional, es una lista enlazada cuyos elementos son listas enlazadas de números.

- (15 puntos) Estime la complejidad de este algoritmo si matriz es una lista doblemente enlazada
- (15 puntos) Estime la complejidad de este algoritmo si matriz es una lista simplemente enlazada

# Ayudas

## Sumatorias

$$\begin{aligned}\sum_{k=1}^n c &= cn & \sum_{k=1}^n k &= \frac{n(n+1)}{2} & \sum_{k=1}^n k^2 &= \frac{n(n+1)(2n+1)}{6} \\ \sum_{k=1}^n k^3 &= \frac{n^2(n+1)^2}{4} & \sum_{k=0}^n ar^k &= \frac{ar^{(n+1)} - a}{r-1} \text{ Si } r \neq 1 & \sum_{k=0}^n ar^k &= (n+1)a \text{ Si } r = 1\end{aligned}$$

## Potencias y logaritmos

- $a^{\log_b(n)} = n^{\log_b(a)}$
- $\frac{1}{a} = a^{-1}$
- $\frac{a^i}{b^i} = \left(\frac{a}{b}\right)^i$
- $\log_a(b) = \frac{\log_c(a)}{\log_c(b)}$

## Formulas solución método del maestro

Aplica para R.R de 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 existe un  $c < 1$  tal que  $af(\frac{n}{b}) \leq cf(n)$  entonces  $T(n) = \Theta(f(n))$ .