

AyED 2023. Módulo 2 - Recuperatorio - 08.07.2023 - Tema 2

Apellido	Nombre	Legajo	Corrigió	Total
Ejercicio 1:	Ejercicio 2:	Ejercicio 3:	Ejercicio 4:	

Ejercicio 1 (4 puntos). Dado el siguiente algoritmo desarrolle e indique el $T(n)$ correspondiente

```

public static int rec2(int n){
    int resultado = 1;
    if (n <= 1)
        resultado = 2;
    else {
        for (int i=0; i < 2; i++)
            resultado = resultado * rec2(n/2);
    }
    return resultado;
}
    
```

$$T(n) = \begin{cases} c_1 & n \leq 1 \\ 2T\left(\frac{n}{2}\right) + c_2 & n > 1 \end{cases}$$

Paso 1 $2T\left(\frac{n}{2}\right) + c_2$

Paso 2 $2\left[2T\left(\frac{n}{2}\right) + c_2\right] + c_2 =$
 $2^2 T\left(\frac{n}{2^2}\right) + 3c_2$

Paso 3 $2^2 \left[2T\left(\frac{n}{2}\right) + c_2\right] + 3c_2 =$
 $2^3 T\left(\frac{n}{2^3}\right) + 7c_2$

Paso i $2^i T\left(\frac{n}{2^i}\right) + (2^i - 1)c_2$

$2^{\log_2 n} T\left(\frac{n}{2^{\log_2 n}}\right) + (2^{\log_2 n} - 1)c_2 =$

$n T\left(\frac{n}{n}\right) + (n-1)c_2 =$
 $(nc_1 + nc_2 - c_2)$

$\begin{matrix} n_0 = 1 \\ 2^i \\ i = \log_2 n \end{matrix}$

Ejercicio 2 (2 puntos). Calcular el $O(n)$ y justifique usando la definición de Big Oh.

$$T(N) = 2n + 10 + n * \log(n)$$

$O(n * \log(n))$

Ejercicio 3 (2 puntos). Se cuenta con un algoritmo ALGO1 que tiene un $T(n)=\log_{10}(n)$ y en una hora puede procesar un problema de tamaño $n=100$. Si se triplica el tiempo con el que se cuenta, ¿cuál es el tamaño del problema que va a poder procesar?

Respuesta:

$$T(n)=\log_{10}(n)$$

1 hora ----- tamaño $n=100$

1h 2 operaciones

3 hs..... 6 operaciones

El tamaño de la entrada va a ser $n= 1.000.000$

Ejercicio 4 (2 puntos).

1) Considere la siguiente recurrencia:

$$\begin{aligned} T(n) &= 1 && \text{si } n=1 \\ T(n) &= 5 * T(n/4) + n && \text{si } n \geq 2 \end{aligned}$$

¿Cuál es el valor de $T(n)$ para $n = 16$?

- (a) 41
- (b) 116
- (c) 100
- (d) 61**
- (e) 126

2) ¿Cuál es el orden del siguiente segmento de código?

```
int i = 1;
int x = 0;
while (i < Math.pow(2, n)) {
    int j = n*n;
    while (j > 0) {
        x++;
        j = j / 2;
    }
    i = i + 1;
}
```

- (a) $O(n^2 * \log n^2)$
- (b) $O(2^n * \log n)$

- (c) $O(2^n * n^2)$
- (d) $O(n^2 * \text{Log } n)$
- (e) $O(2^n * \text{Log } n^2)$
- (f) $O(n^{\log n})$

3) ¿Cuál de las siguientes sentencias es **verdadera** de acuerdo a la **definición formal de Big-Oh**?

- (a) $(n^3 + \log^3 n)$ es $O(n^3 \log n)$
- (b) $(4n^2 \log n + 4n^2)$ es $O(n^2)$
- (c) $(\log^5 n + 8n)$ es $O(n)$
- (d) $(n \log n + \log n)$ es $O(n^2)$
- (e) a y b
- (f) a, c y d
- (g) b y d

4) ¿Qué orden posee la siguiente función de tiempo de ejecución?

$$\begin{array}{ll} T(n) = 1 & \text{si } n=1 \\ T(n) = 4T(n/4) + n & \text{si } n \geq 2 \end{array}$$

- (a) $O(n \log_4 n)$
- (b) $O(\log_4 n)$
- (c) $O(n^4)$
- (d) $O(4^n)$
- (e) $O(n^4 \log_4 n)$