

Análisis de Algoritmos

lunes, 10 de junio de 2024 15:02

1.- Ordene las siguientes funciones: \sqrt{n} , n , 3^n , n^2 , cte, 2^n , $\log_2^2(n)$, $\log_3(n)$, $\log_2(n)$ según su velocidad de crecimiento.

De menor a mayor

- cte

- \sqrt{n}

- $\log_3(n)$, $\log_2(n)$, $\log_3^2(n)$

- n

- n^2

- 2^n

2.- Exprese de qué orden es el siguiente fragmento de código

```
for (int j = 4; j < n; j=j+2) {  $O(n)$   
    val = 0;  
    for (int i = 0; i < j; ++i) {  $O(n)$   
        val = val + i * j;  
        for (int k = 0; k < n; ++k) {  $O(n)$   
            val++;  
        }  
    }  
}
```

(a) $O(n \log n)$

(b) $O(n^2)$

(c) $O(n^2 \log n)$

(e) $O(n^3)$

Opción e $O(n^3)$ ✓

3.- Suponga que dispone de un algoritmo A, que resuelve un problema de tamaño n , y su función de tiempo de ejecución es $T(n) = n * \log(n)$. Este algoritmo se ejecuta en una computadora que procesa **10.000 operaciones** por segundo. Determine el tiempo que requerirá el algoritmo para resolver un problema de tamaño $n=1024$.

Algoritmo A $\rightarrow T(n) = n * \log(n)$

Computador \rightarrow procesa 1000 operaciones por segundo

$$\begin{aligned} T(1024) &= 1024 * \log(1024) \\ &= 1024 * 10 \\ &= 10240 \text{ op} \end{aligned}$$

$$\text{Tiempo} = \frac{10240 \cancel{\text{ops}}}{10000 \cancel{\text{ops}}/\text{seg}} = 10,24 \text{ seg}$$

4.- ¿Cuál es el resultado de la siguiente sumatoria?

$$\sum_{i=3}^8 n \cdot i =$$

- a) $(8-3+1) \cdot n$
- b) $(8-3+1) \cdot i \cdot n$
- c) $33n$
- d) $5n$
- e) $8 \cdot i$
- f) Ninguna de las otras opciones

$$n \sum_{i=3}^8 i = (8-3+1) \cdot i \cdot n \quad \checkmark$$

5.- ¿Cuál de las siguientes sentencias es correcta, según la definición vista en clase?

- (a) n^2 es $O(n^2)$
- (b) n^2 es $O(n^3)$
- (c) n^2 es $O(n^2 \log n)$
- (d) Opciones a y b
- (e) Opciones a, b y c
- (f) Ninguna de las otras opciones

Opción a y b

6.- Dado el siguiente algoritmo

```
void ejercicio5 (int n) {
    if (n >= 2) {
        2 * ejercicio5 (n/2);
        n = n/2;
        ejercicio5 (n/2);
    }
}
```

i) Indique el $T(n)$ para $n \geq 2$

- (a) $T(n) = d + 3 \cdot T(n/2)$
- (b) $T(n) = d + 2 \cdot T(n/2) + T(n/4)$
- (c) $T(n) = d + T(n/2) + T(n/4)$
- (d) $T(n) = d + T(n/2) + T(n/2)$
- (e) $T(n) = d + T(n/2) + T(n/2) + T(n/4)$

Opción b

7.- Dada la recurrencia

$$T(n) = \begin{cases} 1 & \text{para } n \leq 1 \\ T(n/3) + c & \text{para } n > 1 \end{cases}$$

i) ¿Cómo se reemplaza $T(n/3)$, considerando $n/3 > 1$?

- (a) $T(n/3) + c$
- (b) Ninguna de las otras opciones
- (c) $T(n/3) + 1$
- (d) $T(n/3/3) + c$
- (e) $T(n/3/3) + 1$

ii) Desarrolle la función $T(n)$

$$i. T(n/3/3) + c$$

$$ii. P_1 = T(n/3) + c$$

$$P_2 = [T(n/3/3) + c] + c = T(n/9) + 2c$$

$$P_3 = [T(n/9/3) + c] + 2c = T(n/27) + 3c$$

$$P_k = T(n/3^k) + kc$$

$$n/3^k = 1$$

$$n = 3^k$$

$$\log_3(n) = \log_3(3^k)$$

$$\log_3(n) = k$$

$$T(n/3^{\log_3(n)}) + \log_3(n) * c = 1 + \log_3(n) * c = O(\log_3(n))$$

8.- Considere el siguiente fragmento de código:

```
int count = 0; int n = a.length;
for (int i = 0; i < n; i += n/2) {
    for (int j = 0; j < n; j++) {
        a[j]++;
    }
}
```

$O(n)$
 $O(n)$ } $O(n^2)$

Este algoritmo se ejecuta en una computadora que procesa 100.000 operaciones por cada segundo. Determine el tiempo aproximado que requerirá el algoritmo para resolver un problema de tamaño $n=1000$.

- (a) 0,01 seg
- (b) 0,1 seg
- (c) 1 seg
- (d) Ninguna de las opciones anteriores.

$$T(n) = 2n \text{ cte}$$

$$T(1000) = 2 \times 1000 \times cte = 2000$$

$$Tiempo = \frac{2000}{100000} = 0,02$$

Opción d.

9.- Considere la siguiente recurrencia:

$$T(1) = 4$$

$$T(n) = 2 T(n/2) + 5n + 1 \quad (n \geq 2)$$

¿Cuál es el valor de $T(n)$ para $n = 4$?

- (a) 51
- (b) 38
- (c) 59
- (d) 79
- (e) Ninguna de las opciones anteriores.

$$T(4) = 2 T(4/2) + 5 \times 4 + 1 = 2 T(2) + 21$$

$$T(2) = 2 [2 T(2/2) + 5 \times 2 + 1] + 21 = 4 T(1) + 43$$

$$T(1) = 4 [4] + 43 = 59$$

10.- Expresar la función $T(n)$ del siguiente segmento de código:

```
public static void ejercicio (int n) {
    int x = 0;
    int j = 1;
    while (j <= n) {
        for (int i = n*n; i >= 1; i = i - 3)
            x = x + 1;
        j = j * 2;
    }
}
```

- (a) $T(n) = (1/3) * n^2 + \log_2(n)$
- (b) $T(n) = n^2 + (1/3) * \log_2(n)$
- (c) $T(n) = (1/3) * \log_2(n)$
- (d) $T(n) = (1/3) * n^2 * \log_2(n) + \log_2(n)$

$$T(n) = \log_2(n) * \left(\left(\frac{n^2 - 1}{3} + 1 \right) + cte \right)$$

$$T(n) = \log_2(n) * n^2 * \frac{1}{3} + \log_2(n)$$

Opción d

11.- ¿Cuál es el tiempo de ejecución del siguiente método?

```
void fun(int n, int arr[])
{
    int i = 0, j = 0;
    for (; i < n; ++i)
        while (j < n && arr[i] < arr[j])
            j++;
}
```

$$T(n) = CTe_1 + n * CTe_2$$

12.- ¿Cuál es el valor que retorna el método fun1?

```
int fun1 (int n)
{
    int i, j, k, p, q = 0;
    for (i = 1; i < n; ++i)
    {
        p = 0;
        for (j = n; j > 1; j = j/2)
```

```
        ++p;
        for (k = 1; k < p; k = k*2)
            ++q;
    }
    return q;
}
```

N: \log_2

13.- ¿Cuál es el tiempo de ejecución del siguiente código?

```
void fun(int n)
{
    for (int i = 0; i < n / 2; i++)
        for (int j = 1; j + n / 2 <= n; j++)
            for (int k = 1; k <= n; k = k * 2)
                System.out.print("AyED");
}

int main()
{
    int n=8;
    fun(3);
}
```

$$\left. \begin{array}{l} O(n) \\ O(n) \\ O(\log_2(n)) \end{array} \right\} O(n^2 \cdot \log_2(n))$$

$$T(n) = CTe_1 + n/2 * n/2 * (\log_2(n) * CTe_2) = \left(\frac{n}{2}\right)^2 * (\log_2(n) * CTe_2) + CTe_1$$

14.- ¿Cuál es el tiempo de ejecución del siguiente código?

```
void fun(int a, int b)
{
    // Consider a and b both are positive integers
    while (a != b) {
        if (a > b)
            a = a - b;
        else
            b = b - a;
    }
}
```

El peor caso es lo sumo se ejecuta

$\text{Max}(a, b)$

15.- ¿Cuál es el tiempo de ejecución del siguiente código?

```
void fun(int n)
{
    for(int i=0; i<n; i++)
        System.out.print("AyED");
}
```

$T(n) = (\sqrt{n} + 1) * C_{te}$