

AyED 2023. Módulo 2 - 18.11.2023 - Tema 1

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

Ejercicio 1 (5 puntos). Dado el siguiente algoritmo desarrolle e indique el $T(n)$ correspondiente

```
public int metodo (int n, int a){
    int j = 1; int x= 1;
    while ( j < n){
        for (int i = n/2; i < 2* n; i= i +3)
            { algo de O(1); }
        j = j * 2;
    }
    for (int i=a; i <= 100; i++)
        for (int j=1; j < i*n; j++)
            x = x + j * i;

    return x;
}
```

Aclaración: El valor de "a" es fijo mientras esté en ejecución el método.

Ejercicio 2 (1 punto). Calcular el $O(n)$ y justifique usando la definición de Big Oh.

$$T(N) = 10^5 n^3 + n^3 \log(n^2) - n^2$$

Ejercicio 3 (2 puntos). En el siguiente segmento de código tenga en cuenta que el tiempo de ejecución corresponde, aquí, al número de veces que se ejecuta la operación **sum ++**. **sqrt** es la función que devuelve la raíz cuadrada de un número dado.

```
sum :=0
for (i=0; i<sqrt(n)/2; i++)
    for (j=i; j<8+i; j++)
        for (k=j; k<8+j; k++)
            sum++;
```

Si se necesitan 10 ms para ejecutar este código para $n = 100$, ¿cuánto tardará en ejecutarse para $n = 400$?

Ejercicio 4 (2 puntos).

1) ¿Cuál es la complejidad de insertar/borrar al comienzo de un arreglo?

- (a) $O(n)$ (b) $O(\log n)$ $O(1)$ $O(n \log n)$

2) Se tienen cuatro algoritmos que usan un único loop y dentro del loop se ejecuta el mismo conjunto de sentencias. Considere los siguientes loops:

- A) `for(i = 0; i < n; i++)`
B) `for(i = 0; i < n; i += 2)`
C) `for(i = 1; i < n; i *= 2)`
D) `for(i = n; i <= n; i /= 2)`

Si el tamaño de la entrada es n (positivo), ¿cuál algoritmo es más eficiente (sin considerar el objetivo del algoritmo)?

- (a) A (b) B (c) C (d) D

3) Considere las siguientes sentencias:

- I. $(n + k)^m = O(n^m)$ donde k y m son constantes
II. $2^{n+1} = O(2^n)$
III. $2^{(2n+1)} = O(2^n)$

¿Cuál es verdadera según la definición de Big-Oh?

- (a) I y II
(b) I y III
(c) II y III
(d) Todas son verdaderas

4) Dado el siguiente algoritmo

```
int recur (int n) {  
    if ( n ≥ 5 ) {  
        n = n-3;  
        int resultado = 2 + recur(n) + 4 * recur(n-1);  
        return resultado + 3 * recur(n - 2) + recur(n-2);  
    }  
    else  
        return 1;  
}
```

Indique el $T(n)$ para $n \geq 5$

- (a) $T(n) = c + T(n) + 4 * T(n-1) + 3T(n-2) + T(n-2)$
(b) $T(n) = c + T(n) + T(n-1) + 4T(n-2)$
(c) $T(n) = c + T(n-3) + 4 * T(n-4) + 3T(n-5) + T(n-5)$
(d) $T(n) = c + T(n-3) + T(n-4) + 2T(n-5)$
(e) $T(n) = c + T(n) + 4 * T(n-4) + 4T(n-5)$
(f) $T(n) = c + 2T(n-3) + 4 * T(n-4) + 3T(n-2) + T(n-2)$