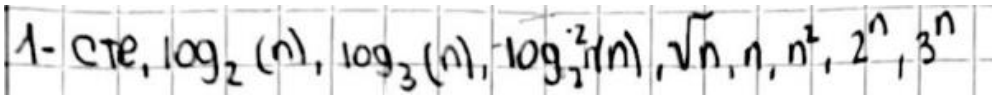
Algoritmos y Estructuras de Datos Cursada 2024
Análisis de Algoritmos

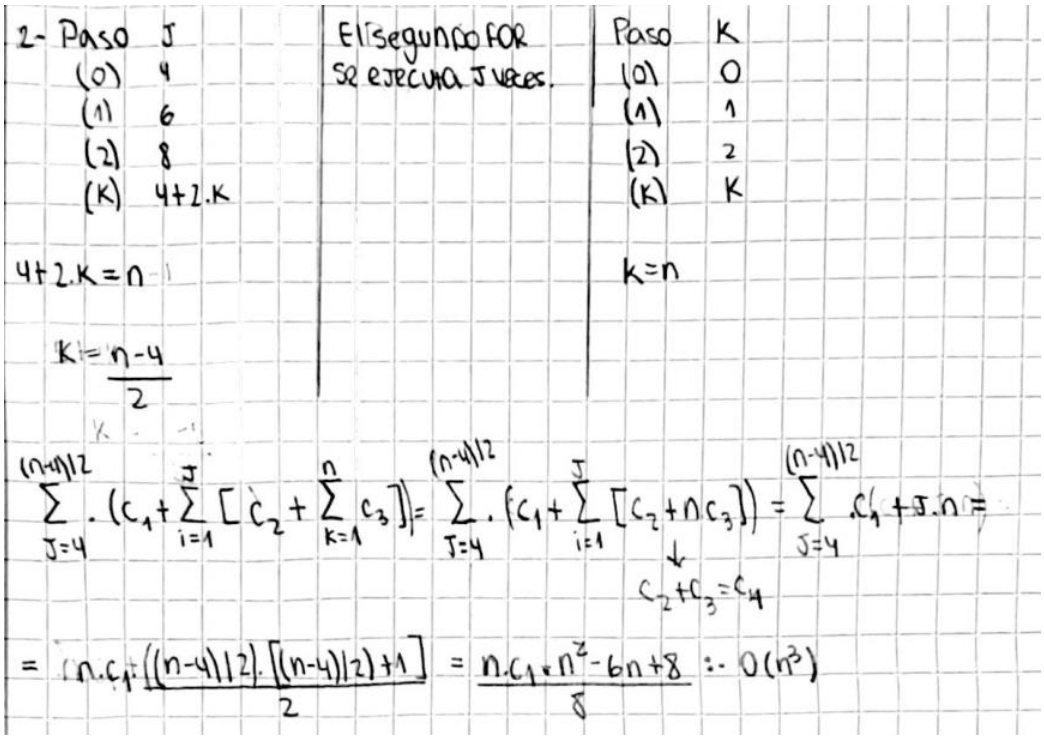
1.- Ordene las siguientes funciones: $\sqrt{n}$, n, 3^n, n_2, cte, 2^n, $\log_2{}^2(n)$, $\log_3(n)$, $\log_2(n)$ según su velocidad de crecimiento.



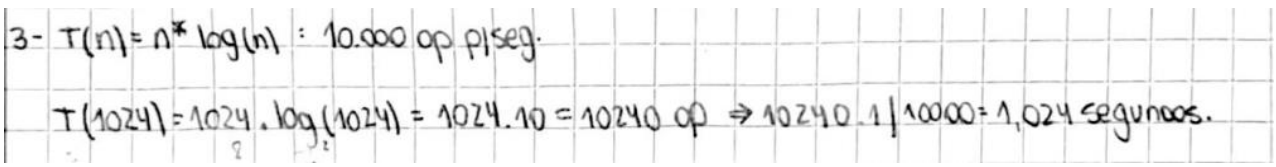
2.- Exprese de qué orden es el siguiente fragmento de código

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

(a) O(n logn) (b) O(n^2) (c) O(n^2 log n) (e) O(n^3)



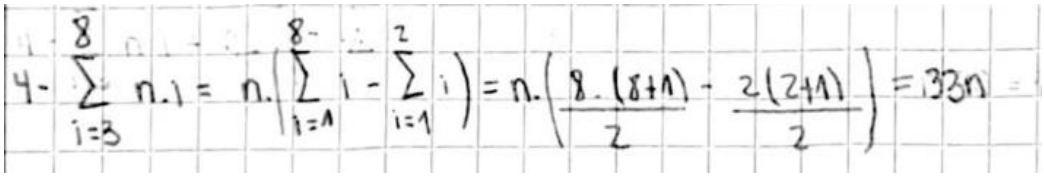
3.- Suponga que dispone de un algoritmo A, que resuelve un problema de **tamaño n**, y su función de tiempo de ejecución es **T(n) = n * log(n)**. Este algoritmo se ejecuta en una computadora que procesa **10.000 operaciones** por segundo. Determine el **tiempo** que requerirá el algoritmo para resolver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



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

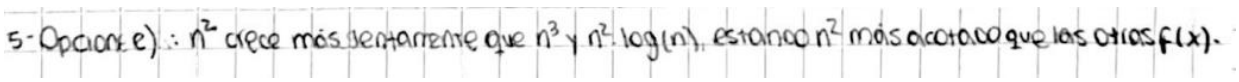
$$\sum_{i=3}^{8} n * i =$$

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



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



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>=2

(a) T(n) = d + 3 * T(n/2)
(b) T(n) = d + 2 * 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)

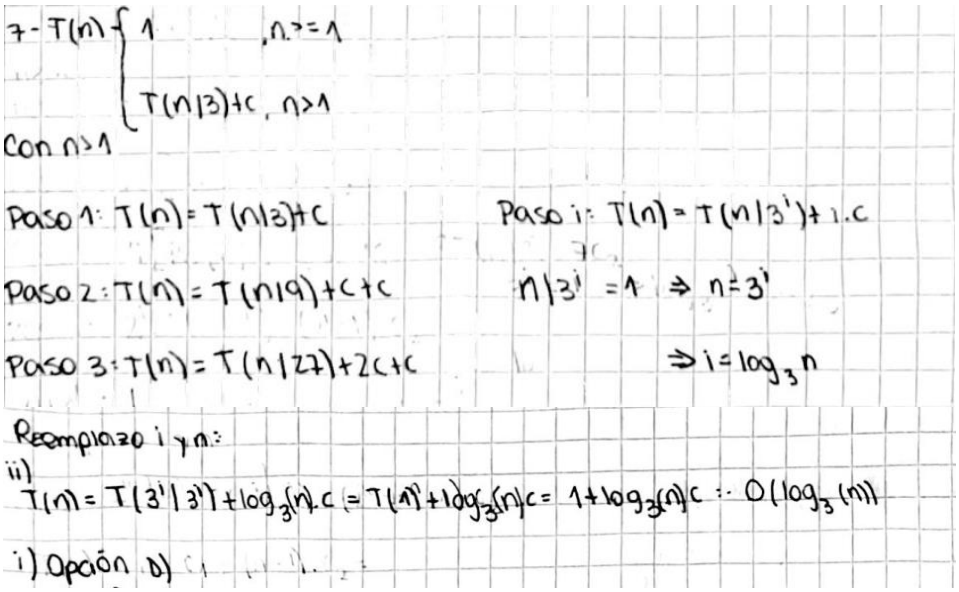
7.- Dada la recurrencia

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

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)

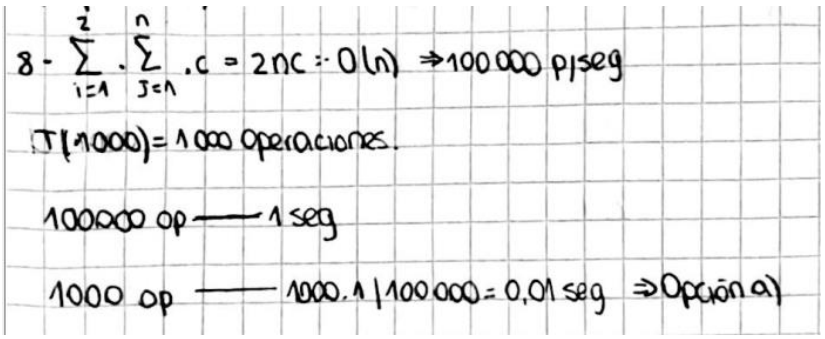


8.- Considere el siguiente fragmento de código:

```
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]++;
        }
}
```

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.

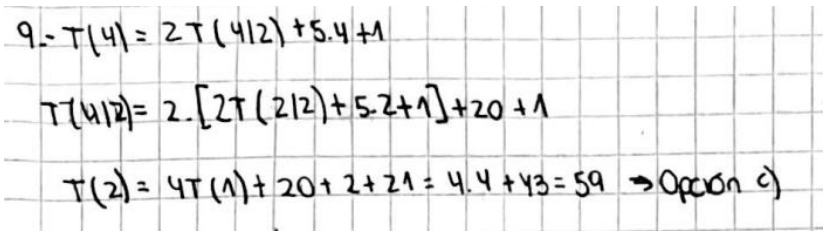


9.- Considere la siguiente recurrencia:

T(1) = 4
T(n) = 2 T(n/2) + 5n + 1 (n ≥ 2)

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

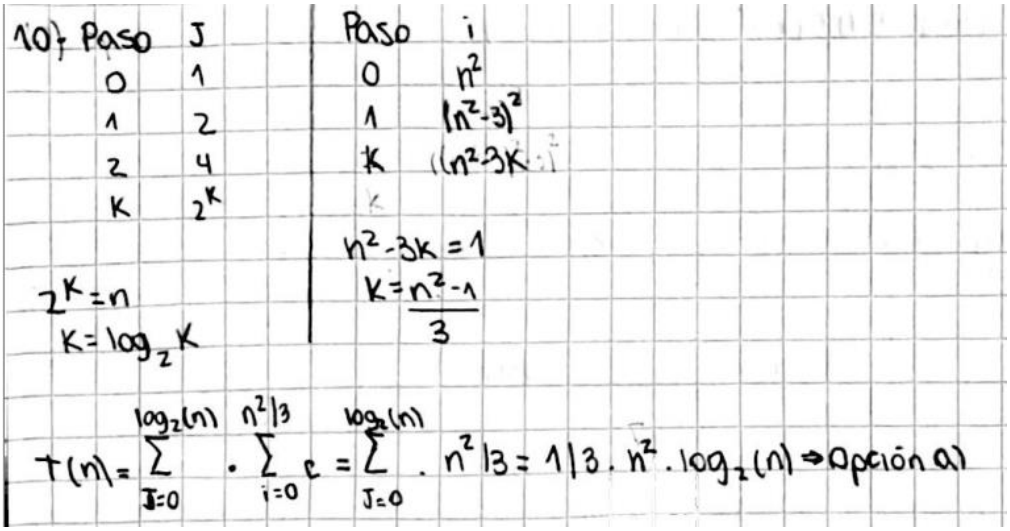
(a) 51
(b) 38
(c) 59
(d) 79
(e) Ninguna de las opciones anteriores



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

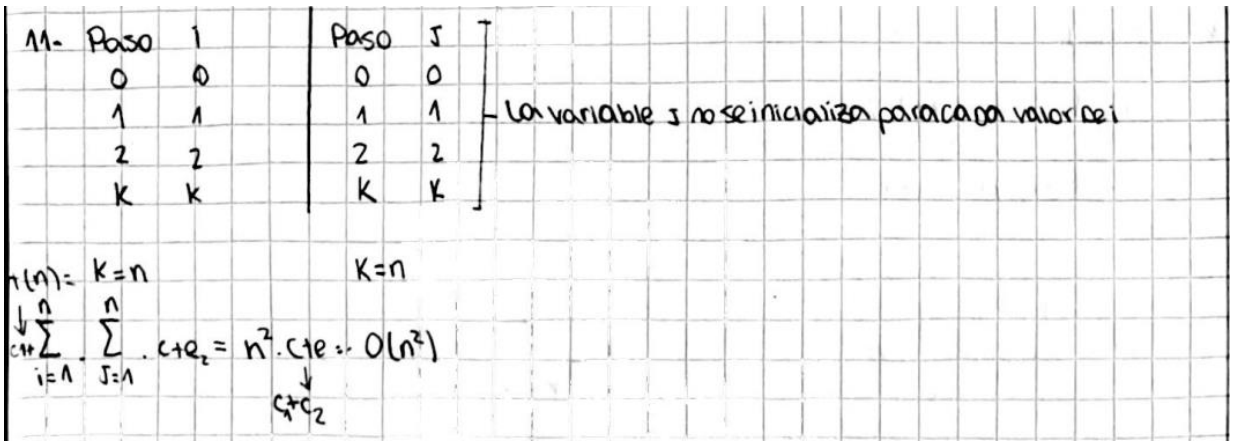
```
public static void ejercicio (int n) {
        int x = 0;
        int j = 1;
        while ( j <= n ) {
                for ( int i = n*n ; i >=1 ; i = i - 3 )
                        x = x + 1 ; j = j * 2 ;
        }
}
```

(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) $T(n) = (1/3) * n^2 * \log_2(n) + \log_2(n)$



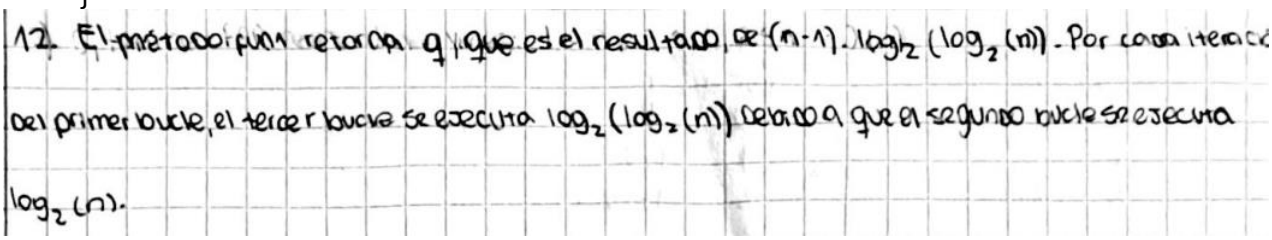
11.- ¿Cuál es el tiempo de ejecución del siguiente método?

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



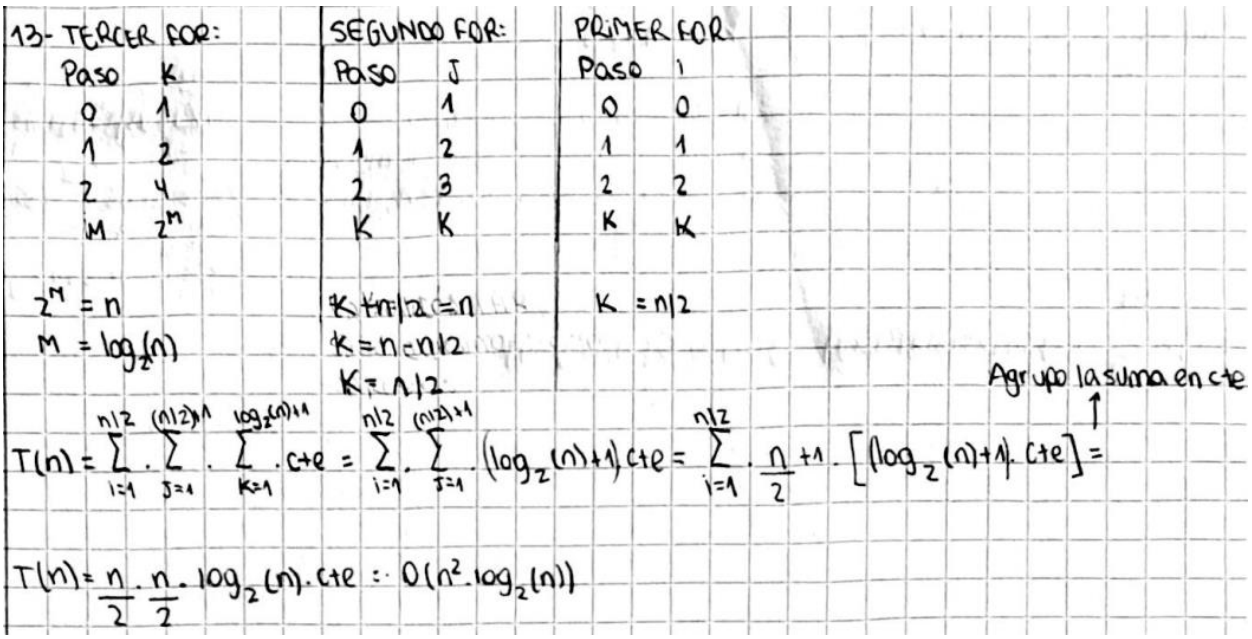
12.- ¿Cuál es el valor que retorna el método fun1?

```
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;
}
```



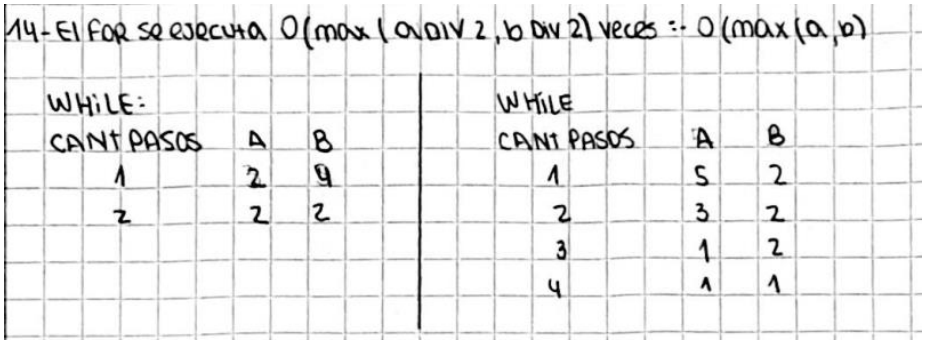
13.- ¿Cuál es el tiempo de ejecución del siguiente código?

```
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);
}
```



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;
        }
}
```



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

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

15- Paso | i
0 | 0
1 | 1
2 | 2
k | k

$K.K = n$
$K^2 = n$
$K = \sqrt{n}$

$\Rightarrow T(n) = \sum_{i=1}^{\sqrt{n}} .cte = \sqrt{n}.cte \therefore O(\sqrt{n})$

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

```
int fun(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j < n; j += i) {
            // Some O(1) task
        }
    }
}
```

Nota: Tenga en cuenta que (1/1 + 1/2 + 1/3 + … + 1/n) se puede acotar con O

16- PRIMER FOR:
Paso | i
0 | 1
1 | 2
k | k+1

El FOR se ejecuta n+1 veces
$K = n+1$

SEGUNDO FOR:
Paso | j
0 | 1
1 | 1+i
2 | 1+2i
3 | 1+3i
k | 1+ki

El FOR se ejecuta con i=1, n veces
El FOR se ejecuta con i=2, n/2 veces
El FOR se ejecuta con i=n, n/n veces

$\sum_{i=1}^{n} n/i = 1/2 \sum_{i=1}^{n} 1/i$

Serie armónica es acotada por n:
↳ se resume en log(n)

$T(n) = \sum_{i=1}^{n+1} .\log(n).cte = (n+1).\log(n) \therefore O(n.\log(n))$