UNIVERSIDAD DEL VALLE DE GUATEMALA

Teoría de probabilidades

Sección 10

Laboratorio 8

Humberto Alexander de la Cruz Chanchavac - 23735

Daniel Oswaldo Juárez Herrera - 23709

GUATEMALA, 15 de octubre de 2025

Repositorio: https://github.com/hadelacruz/Lab8_Teoria.git

Video: https://youtu.be/mAVceSycYTs

Problema 1

```
void function(int n) {
    int i, j, k, counter = 0;
    for (i = n/2; i <= n; i++) {
        for (j = 1; j+n/2 <= n; j++) {
            for (k = 1; k <= n; k = k*2) {
                counter++;
            }
        }
    }
}
```

----------→ 1

----------→ n/2

----------→ n/2

----------→ log2 n

Problema 1

$$f(n) = 1 + \left(\frac{n}{2}\right)\left(\frac{n}{2}\right)(\log_2 n)$$

$$f(n) = \frac{n^2}{4}\log_2 n \qquad g(n) = n^2 \log_2 n$$

$$f(n) \leq c \cdot g(n)$$

$$\frac{1}{4} n^2 \log_2 n \leq c \cdot n^2 \log_2 n \quad ; \ c = 2$$

$$\frac{1}{4} n^2 \log_2 n \leq 2 n^2 \log_2 n$$

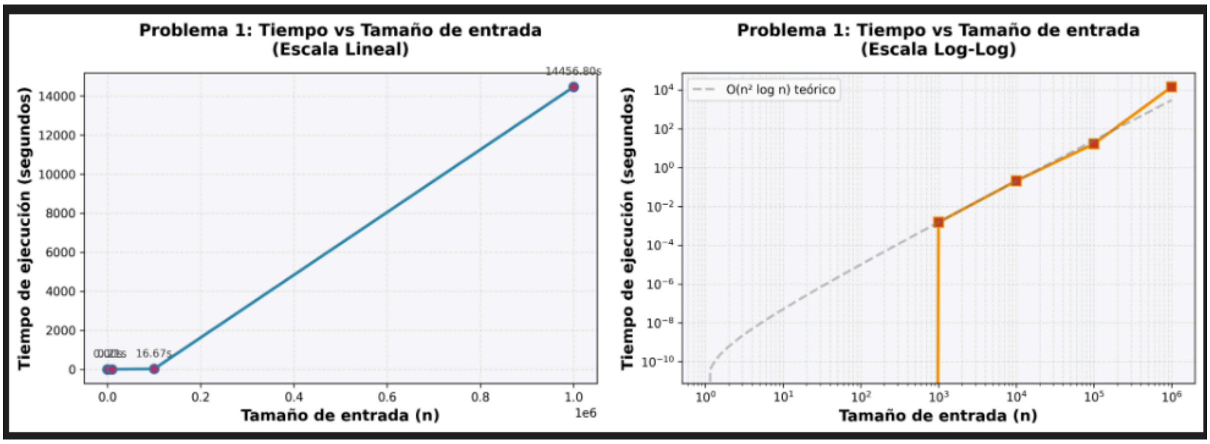
$$\therefore f(n) \leq c g(n) \quad ; \ c = 2 \ \text{y} \ n_0 = 1$$

$$f(n) = O(n^2 \log_2 n)$$

Tabla

n	tiempo (s)
1	0
10	0
100	0
1000	0.001
10000	0.13584
100000	16.669
1000000	14456.8

Gráfica



Problema 2

```c
void function(int n) {
    if (n <= 1) return;
    int i, j;
    for (i = 1; i <= n; i++) {
        for (j = 1; j <= n; j++) {
            printf("Sequence\n");
            break;
        }
    }
}
```

----------→ 1

----------→ n

----------→ break

Problema 2

$f(n) = n+1 \qquad g(n) = n$

$f(n) \leq c \cdot g(n)$

$n+1 \leq c \cdot n \quad ; \quad c = 2$

$n+1 \leq 2n$

$n \geq 1$

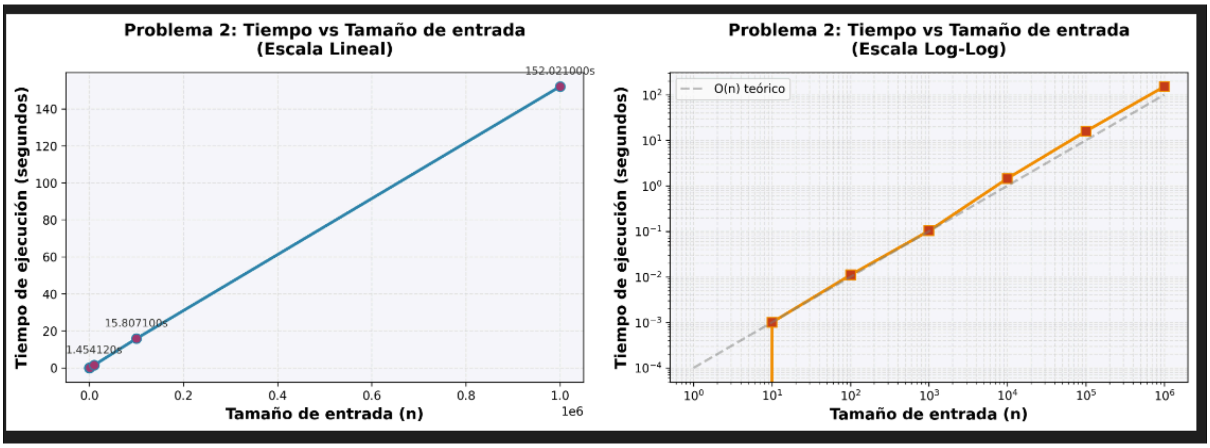
$\therefore \; f(n) \leq c g(n) \quad c; \; c = 2 \;$ y $\; n_0 = 1$

$f(n) = n$

Tabla

n	tiempo (s)
1	0
10	0.001002
100	0.011002
1000	0.10263
10000	1.45412
100000	15.8071
1000000	152.021

Gráfica



Problema 3

```c
void function(int n) {
    int i, j;
    for (i = 1; i <= n/3; i++) {
        for (j = 1; j <= n; j += 4) {
            printf("Sequence\n");
        }
    }
}
```

----------→ 1

----------→ 1/3 n

----------→ 1/4 n

Problema 3

$f(n) = \frac{n^2}{12} + 1 \qquad g(n) = n^2$

$\frac{n^2}{12} + 1 \leq c \cdot n^2 \; ; \; c = 1$

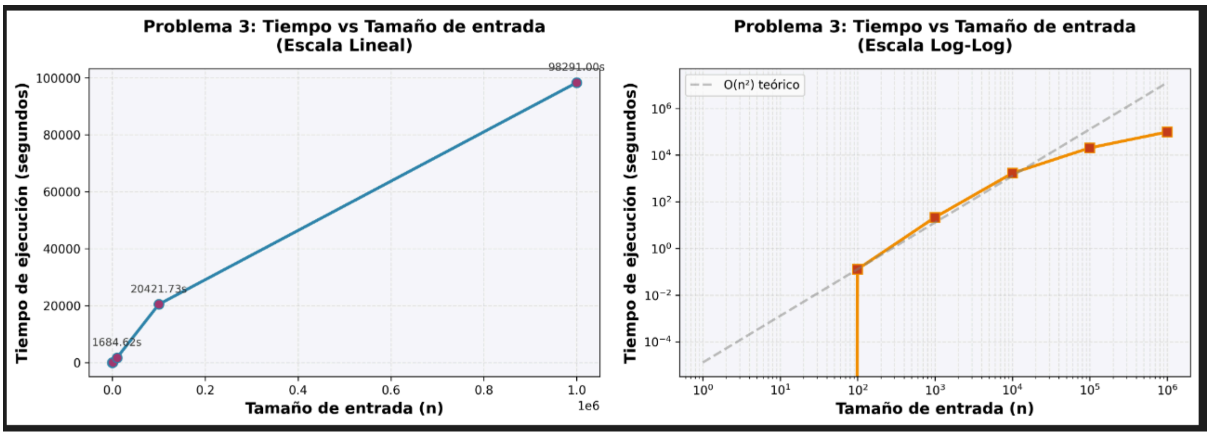
$\therefore \; f(n) \leq c\,g(n) \; ; \; c = 1$ y $n_0 = 1$

$f(n) = n^2$

Tabla

n	tiempo (s)
1	0
10	0
100	0.128443
1000	21.523
10000	16684.87
100000	20421.73
1000000	98291.0

Gráfica



Problema 4

Problema 4

Busqueda Lineal

Recorre secuencialmente un arreglo de n elementos, comparando cada una

```
BL (Array a, int x):
    n= len(a)
    for i in range (0,n):
        if a[1] == x:
            return i
    return -1
```

Mejor Caso	Complejidad
El elemento buscado se encuentra en la primera posición	$O(1)$

Caso Promedio	
Elemento en posición aleatoria	$O(\frac{n}{2}) \to O(n)$

Peor Caso	
El elemento no encontrado	$O(n)$

Problema 5

Problema 5

a) Si $f(n) = \Theta(g(n))$ y $g(n) = \Theta(h(n))$, entonces $h(n) = \Theta(f(n))$

Por definición:

$f(n) = \Theta(g(n))$ implica que existe constantes positivas c_1, c_2, n_0

$c_1 g(n) \leq f(n) \leq c_2 g(n) \quad \forall n \geq n_0$

$g(n) = \Theta(h(n))$ implica que existen constantes positivas c_3, c_4, n_1

$c_3 h(n) \leq g(n) \leq c_4 h(n) \quad \forall n \geq n_1$

$\therefore \; n \geq \max(n_0, n_1)$

$h(n) = \Theta(f(n))$ Verdadero

b) Si $f(n) = O(g(n))$ y $g(n) = O(h(n))$ entonces $h(n) = \Omega(f(n))$

Por definición

$f(n) = O(g(n))$ implica que existen constantes positivas c, n_0

$f(n) \leq c \cdot g(n) \quad \forall n \geq n_0$

$g(n) = O(h(n))$ implica que existen constantes positivas d, n_1

$g(n) \leq d \cdot h(n) \quad \forall n \geq n_1$

$\therefore \; h(n) \geq \frac{1}{c \cdot d} f(n) \quad \forall n \geq \max(n_0, n_1)$

$h(n) = \Omega(f(n))$ Verdadero

c) $F(n) = \Theta(n^2)$

El numero total de iteraciones es

$$\sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} 1 = \frac{n(n-1)}{2} = O(n^2)$$

Pero la suma total es

$$\sum_{i=0}^{n-1} \frac{(n-i-1)(n-i)}{2} = O(n^3)$$

Ya aque que en cada iteración se crea una nueva tupla

$\therefore$ la ejecución es $O(n^3)$

Falso