

# Teoría de la Computación 2022

Lab 07

05.octubre.2022

1. Suponga que cada una de las expresiones siguientes da el tiempo de procesamiento  $T(n)$  gastado por un algoritmo para resolver un problema de tamaño  $n$ . Seleccione el(los) término(s) dominante(s) que tienen el aumento más pronunciado en  $n$  y especifique el Big-Oh de menor complejidad para cada algoritmo.

Expresión	Términos dominantes	$O(\cdot)$
$5 + 0.001n^3 + 0.025n$		
$500n + 100n^{1.5} + 50n \log_{10} n$		
$0.3n + 5n^{1.5} + 2.5n^{1.75}$		
$n^2 \log_2 n + n(\log_2 n)^2$		
$n \log_3 n + n \log_2 n$		
$3 \log_8 n + \log_2 \log_2 \log_2 n$		
$100n + 0.01n^2$		
$2^n + n^{0.5} + 0.5n^{1.25}$		
$0.01^n \log_2 n + n(\log_2 n)^2$		
$100n \log_3 n + n^3 + 100n$		
$0.003 \log_4 n + \log_2 \log_2 n$		

2. Calcular cuál es el tiempo promedio que requiere un computador para hacer una operación aritmética de tipo suma entre dos cantidades.

Sugerencia, hacer una simulación de sumar dos enteros  $a$  y  $b$ , y repetirla una cantidad grande  $N$  de veces. Sumar el tiempo que tarda estas repeticiones y luego calcular el tiempo promedio.

- ¿Hay diferencias al repetir varias veces este experimento?. Repita el experimento  $k = 10^5$  veces, y muestre una distribución de los tiempos promedio. Analice sus resultados.
- ¿Hay diferencias al incrementar el tamaño de las cantidades  $a$  y  $b$ . Pruebe con números de tamaños, 5 dígitos, 10 dígitos, 15 dígitos, y compare los tiempos de ejecución.
- ¿Hay diferencias según el tipo de dato? Experimente con números enteros, y luego repita los experimentos con variables de tipo float, y compare las diferencias entre tiempos de ejecución.
- ¿Hay diferencias según la operación aritmética? Sustituya el cálculo de  $a + b$  por  $a/b$ , y repita los experimentos con esta otra operación. Compare las diferencias entre tiempos de ejecución.

3. Muestre que en el algoritmo de reducción gaussiana (para resolver sistemas de ecuaciones):

**Algoritmo:** (Eliminación Gaussiana).

Inputs:  $A \in \mathbb{R}^{n \times n}$ , Outputs:  $L \in \mathbb{R}^{n \times n}$ ,  $U \in \mathbb{R}^{n \times n}$ .

Initialise  $U = A$ ,  $L = I$  (la matriz identidad de  $n \times n$ )

For  $k = 1$  to  $n-1$ :

    For  $i = k + 1$  to  $n$ :

$$L_{i,k} = 1 / U_{i,k}$$

    For  $j = k$  to  $n$ :

$$U_{i,j} = U_{i,j} - L_{i,k}U_{k,j}.$$

hay exactamente  $n^3 + n^2 - 2n$  operaciones aritméticas (sumas, restas, productos, cocientes, y asignaciones), de modo que el tiempo de ejecución es de orden  $O(n^3)$ .

4. Muestre que  $T(n) = a_0 + a_1n + a_2n^2 + a_3n^3$  es de orden  $O(n)$  usando la definición formal de la notación Big-Oh. Esto es, hallar una constante  $C > 0$ , y un umbral  $n_0 \in \mathbb{N}$  tales que

$$T(n) \leq Cn^3, \quad \text{para todo } n \geq n_0.$$

5. Proporcione un análisis del tiempo de ejecución (notación Big-Oh) para cada uno de los siguientes 4 fragmentos de programa. Tenga en cuenta que el tiempo de ejecución corresponde aquí al número de veces que se ejecuta la operación suma. *sqrt* es la función que devuelve la raíz cuadrada de un número dado.

a) `suma = 0`  
`For i in range(0, sqrt(n)/2):`  
`suma += 1`  
`For j in range(0, sqrt(n)/4):`  
`suma += 1`  
`For k in range(0, 8+j):`  
`suma += 1`

b) `suma = 0`  
`For i in range(0, sqrt(n)/2):`  
`For j in range(i, 8+i):`  
`For k in range(j, 8+j):`  
`suma += 1`

c) `suma = 0`  
`For i in range(0, 2*n):`  
`For j in range(0, i*i):`  
`For k in range(0, j):`  
`If (j % i == 1):`  
`suma += 1`

d) `suma = 0`  
`For i in range(0, 2*n):`  
`For j in range(0, i*i):`  
`For k in range(0, j):`  
`If (j % i):`  
`suma += 1`

6. Los algoritmos  $A$  y  $B$  gastan exactamente  $T_A(n) = 0.1n^2 \log_{10} n$  y  $T_B(n) = 2.5n^2 \mu s$ , respectivamente, para un problema de tamaño  $n$ .

Elija el algoritmos que es mejor, en el sentido de la notación Big-Oh, y encuentre un problema de tamaño  $n_0$  tal que para cualquier tamaño  $n \geq n_0$ , el algoritmo elegido siempre mejora al otro.

Si nuestro problema a resolver es de tamaño  $n \leq 10^9$ , ¿cuál algoritmos recomendaría usar?

7. Se debe elegir entre uno de los dos paquetes de software,  $A$  ó  $B$ , para procesar colecciones de datos, que contienen cada uno hasta  $10^9$  registros. El tiempo promedio de procesamiento del software  $A$  es  $T_A(n) = 0.001n$  ms y el tiempo promedio de procesamiento del software  $B$  es  $T_B(n) = 500\sqrt{n}$  ms.

¿Cuál algoritmo tiene un mejor rendimiento en el sentido Big-Oh? Encontrar las condiciones exactas en las que uno de estos software mejora al otro ( $C, n_0$ ).