

Análisis de Algoritmos y Estructura de Datos

**Refuerzo/ejercicios
Complejidad de algoritmos**

Prof. Violeta Chang C

Semestre 2 – 2023



Complejidad de algoritmos

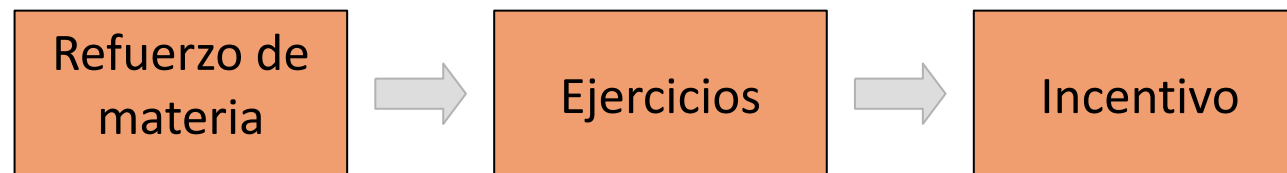
- **Contenidos:**

- Complejidad de algoritmos iterativos
- Complejidad de algoritmos recursivos

- **Objetivos:**

- Explicar mecanismos de cálculo de complejidad de algoritmos iterativos y recursivos
- Calcular complejidad de algoritmos iterativos y recursivos

Ruta de sesión





Mapa Conceptual

ALGORITMO

¿Qué es?

- Secuencia finita de pasos que resuelve un problema bien definido

Características

- Preciso
- Finito
- Definido

¿Cómo se representa?

- Diagrama de flujo
- Seudocódigo

¿Cómo comparar uno con otro?

- Empíricamente
- Teóricamente

Tipos según complejidad

- Constante
- Logarítmico
- Lineal
- Cuadrático
- Cúbico
- Exponencial



Complejidad Algorítmica

1

Conocer el tiempo de ejecución teórico de cada instrucción.

2

Anotar la suma de todos los tiempos.

3

Técnica algoritmos iterativos:

- Resolver y acotar la función.
- Identificar n_0 , k y $f(n)$
- $O(f(n))$

Técnica algoritmos recursivos

- Visualizar estructura general
- Identificar la ecuación de recurrencia
- Resolver usando reducción por sustracción o reducción por división.



Complejidad de algoritmos recursivos

- Reducción por sustracción:

- se realizan a llamadas recursivas
- tamaño del problema se reduce una **cantidad constante b** en cada llamada
- Operaciones parte no recursiva tienen $O(n^k)$

$$T(n) = aT(n - b) + O(n^k), \text{ si } n \geq b$$

$$O(\quad) = \begin{cases} n^k & \text{si } a < 1 \\ n^{k+1} & \text{si } a = 1 \\ a^{\frac{n}{b}} & \text{si } a > 1 \end{cases}$$

- Reducción por división:

- se realizan a llamadas recursivas
- tamaño del problema se reduce en una **proporción b (n/b)** en cada llamada
- Operaciones parte no recursiva tienen $O(n^k)$

$$T(n) = aT\left(\frac{n}{b}\right) + O(n^k), \text{ si } n \geq b$$

$$O(\quad) = \begin{cases} n^k & \text{si } a < b^k \\ n^k \log n & \text{si } a = b^k \\ n^{\log_b a} & \text{si } a > b^k \end{cases}$$



Actividad en parejas - 20 minutos

1. Escribir un algoritmo **iterativo** que calcule la potencia de un número entero. Calcular la complejidad del algoritmo propuesto
2. Escribir un algoritmo **recursivo** que calcule la potencia de un número entero. Calcular la complejidad del algoritmo propuesto



Actividad en parejas - 15 minutos

Un índice de equilibrio en un arreglo A de tamaño n es un índice X tal que la suma de los elementos del subarreglo $A(1..X)$ es igual a la suma de los elementos del subarreglo $A(X+1..n)$.

Por ejemplo, el arreglo $[-3 \ 5 \ 2 \ -3 \ -1 \ 3 \ -1 \ -2]$ tiene a 5 como su índice de equilibrio, mientras que el arreglo $[-3 \ 5 \ 3 \ -3 \ -1 \ 3 \ -1 \ -2]$ no tiene índice de equilibrio.

Escribir un algoritmo, en pseudocódigo, para encontrar un índice de equilibrio si existe, o que reporte que no existe. No se puede utilizar arreglo auxiliar. Calcular la complejidad del algoritmo propuesto, y justificar la complejidad obtenida.



Incentivo - individual - 15 minutos

Un elemento de mayoría en un arreglo A de tamaño n es un elemento que aparece más de $n/2$ veces (por lo tanto a lo más hay uno de esos elementos). Por ejemplo, el arreglo $[3\ 3\ 4\ 2\ 4\ 4\ 2\ 4\ 4]$ tiene un elemento de mayoría (4), mientras que el arreglo $[3\ 3\ 4\ 2\ 4\ 4\ 2\ 4]$ no tiene. Escribir un algoritmo, en pseudocódigo, para encontrar un elemento de mayoría si existe, o que reporte que no existe. No se puede utilizar arreglo auxiliar. Calcular y justificar la complejidad del algoritmo propuesto.



Co-evaluación de incentivo - 5 minutos

I.	El algoritmo resuelve correctamente el problema planteado. (SI 1 punto / NO 0 punto)	
II.	El algoritmo está escrito en pseudocódigo ordenado (SI 1 punto / NO 0 punto)	
III.	El algoritmo está escrito en el formato establecido (SI 1 punto / NO 0 punto)	
IV.	El algoritmo identifica entradas correctamente (SI 1 punto / NO 0 punto)	
V.	El algoritmo identifica y declara salidas de manera correcta. (SI 1 punto / NO 0 punto)	
VI.	Calcula correctamente la complejidad (SI 1 punto / NO 0 punto)	
VII.	Justifica la complejidad del algoritmo propuesto. (SI 1 punto / NO 0 punto)	
	TOTAL PUNTOS	

Actividad de cierre



- Ir a menti.com e ingresar código 1328 8245



<div>  Octubre 2023 </div>						
Domingo	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
1	2	3 	4 	5	6 	7
8	9 	10	11	12 	13 	14
15	16 <small>Encuentro de Dos Mundos</small>	17	18	19 	20 	21
22	23	24	25 	26 	27  <small>Día Nacional de las Iglesias Evangélicas y Protestantes</small>	28

Receso