

Tarea 1: Propuesta de ejercicios

Jose Antonio Barrientos Sanchez

18 de agosto de 2025

Ejercicio 1

Ordene la siguiente lista de funciones de menor a mayor según su tasa de crecimiento asintótico. Es decir, si una función $f(n)$ precede a $g(n)$, entonces $f(n) = O(g(n))$.

$$\begin{array}{ccccccc} \sqrt{n} & n \log_2(n) & 2^n & n^2 & \log_2(n) \\ n! & n & 1000 & n^3 & \log_2(\log_2(n)) \end{array}$$

Ejercicio 2

Determine la complejidad temporal del siguiente fragmento de código.

```
1 def funcion_ejemplo(n):
2     suma = 0
3     i = 1
4     while i <= n:
5         suma += i
6         i = i * 2
7     return suma
```

Listing 1: Bucle simple

Ejercicio 3

Usando la definición formal de Big O, demuestre que $5n^3 + 3n + 2$ es $O(n^3)$.

Ejercicio 4

Demuestre que para todo $k \geq 0$ y todo conjunto de constantes reales $\{a_k, a_{k-1}, \dots, a_1, a_0\}$, se cumple que:

$$a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n + a_0 = O(n^k)$$

Ejercicio 5

Muestre que para cualesquiera constantes reales a y b , con $b > 0$, se cumple que:

$$(n + a)^b = \Theta(n^b)$$