

Complejidad



Introducción al pensamiento computacional

¿Qué hace bueno a un algoritmo?

Que haga uso **eficiente** de los **recursos** que necesita:

- El **tiempo** que necesita para **correr**.
- El **espacio** que necesita para **correr**.

¿Cómo podemos medir el tiempo?

Comúnmente medimos el tiempo en segundos, minutos, horas, etc. Pero esto tiene un problema:

Esta medida depende del computador que se use



vs



**¡El tiempo se
mide en
operaciones!**

¿Qué es una operación?

Es un **proceso** que es **relevante** para el **problema** que estamos resolviendo:

- Adivinar el número: Preguntar si mi número es el correcto.
- Dibujar una cuadrícula: Trazar un segmento de línea recta.

El número de operaciones suele ser una medida de tiempo independiente del computador que esté ejecutando un algoritmo

Problemas con el número de operaciones

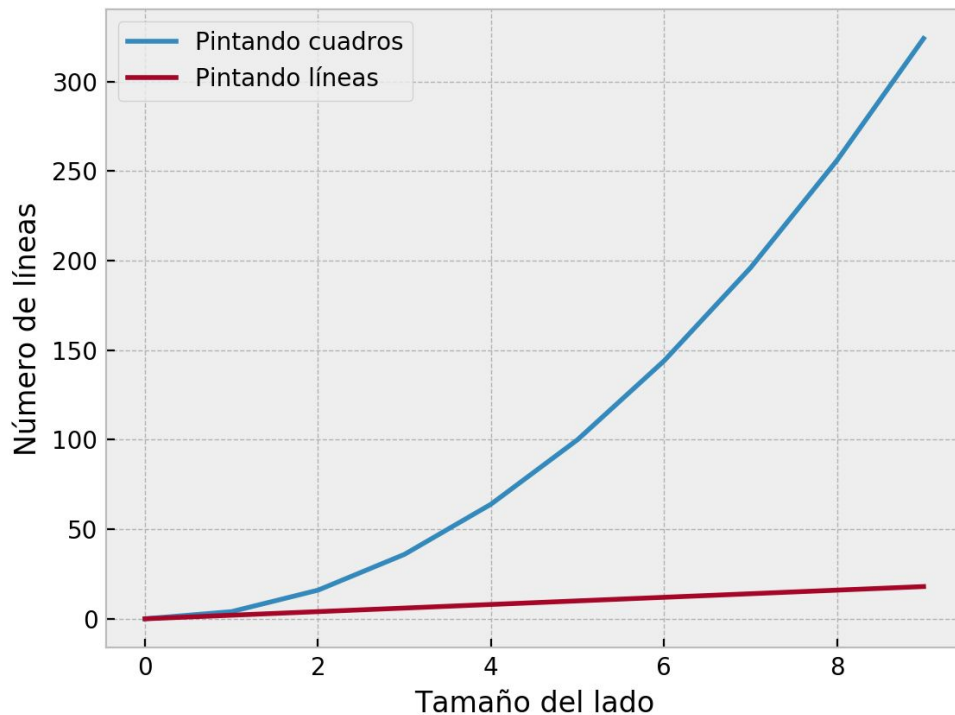
El número de operaciones no es una medida perfecta. Depende de:

- El tamaño de la entrada:
 - Es más rápido adivinar un número entre 1 y 100 que entre 1 y 1 000 000.
 - Es más rápido dibujar una cuadrícula de 4 x 4 que una de 1000 x 1000.
- La entrada del algoritmo:
 - Búsqueda binaria: Adivinar 14 entre 1 y 100 requiere más operaciones que 50.

Número de pasos vs. Tamaño

Pensemos el número de pasos como una función del tamaño de la entrada:

Lado	Líneas (Cuadros)	Líneas (Líneas)
1	4	4
2	16	6
3	36	8
4	64	10
5	100	12



Casos

- Un algoritmo puede tomar un número distinto de operaciones para un mismo tamaño de entrada.
- Para esto solemos contar el número de operaciones en 3 casos distintos:
 - El mejor caso.
 - El caso promedio.
 - El peor caso.

¿Cuántas operaciones se necesitan para encontrar un número entre 1 y 100 preguntando los números desde 1 hasta 100 en orden?

Casos

- Un algoritmo puede tomar un número distinto de operaciones para un mismo tamaño de entrada.
- Para esto solemos contar el número de operaciones en 3 casos distintos:
 - El mejor caso: 1 operación (cuando el número es 1)
 - El caso promedio: 51 operaciones (en promedio)
 - El peor caso: 100 operaciones (cuando el número es 100)

¿Cuántas operaciones se necesitan para encontrar un número entre 1 y 100 preguntando los números desde 1 hasta 100 en orden?

Complejidad en tiempo

El **número de operaciones** que necesita un algoritmo en **función del tamaño** de la entrada para el **peor caso** se conoce como su **complejidad en tiempo**.

Complejidad: Búsqueda binaria



Notación O-Grande (Big-O)

- Nos interesa cómo crece el tiempo de ejecución y no su valor exacto.
- Para esto introducimos la notación O-grande:

$$n + 1 = O(n)$$

$$2n = O(n)$$

$$n^2 + n = O(n^2)$$

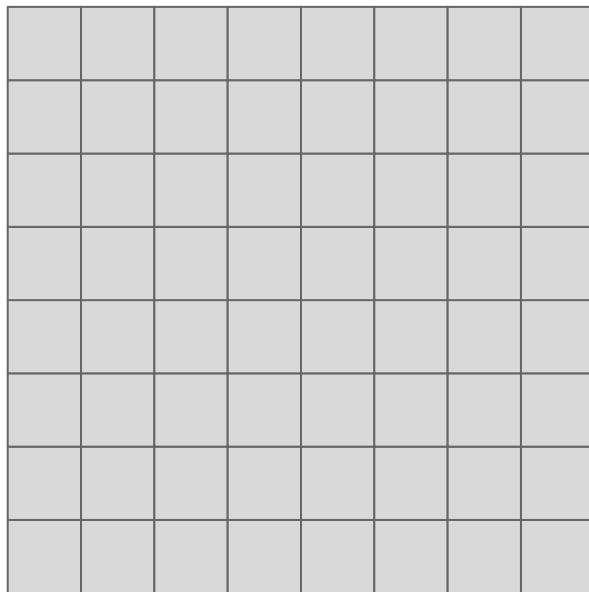
$$n + \log(n) = O(n)$$

$$\log(n) + 34 = O(\log(n))$$

$$42 = O(1)$$

**Tomamos la “potencia”
de n dominante**

Complejidad: Pintar una cuadrícula



Ejercicio

¿Cuál es la complejidad en tiempo para “pintar” una cuadrícula de $n \times n$ usando dobleces en vez de un lápiz?
