

# Orden de complejidad de algoritmos

# ¿Qué es un algoritmo?

Un algoritmo es un conjunto ordenado de instrucciones bien definidas, no ambiguas y finitas, que permite resolver un determinado problema computacional.

**Ejemplo:** Diseñar un algoritmo para, dado un número entero  $A$  mayor a 1, mostrar sus divisores diferentes de 1 y  $A$

# ¿Qué es un algoritmo?

Un corolario de la definición anterior es que un determinado problema computacional puede ser resuelto por diversos algoritmos diferentes.

*“Si deseas ser un buen diseñador de algoritmos debes preguntarte siempre a manera de mantra: ¿se puede hacer mejor?”*

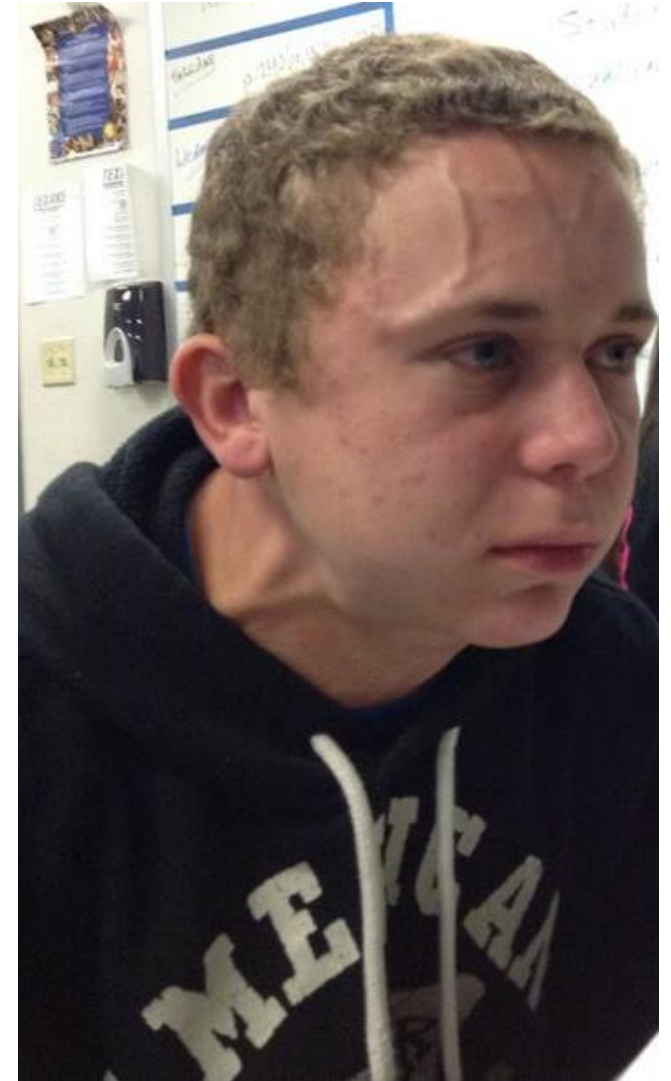
- Roughgarden, T. (2012) Algorithms design and analysis part 1

*“Quizá el principio más importante para el buen diseñador de algoritmos es negarse a estar satisfecho”*

- Aho et al. (1974) The design and analysis of computer algorithms

*“La primera solución que se nos ocurre o la solución más obvia generalmente no es la mejor*

-Yo

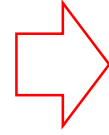


# ¿Cómo se mide un algoritmo?

Una manera objetiva de determinar que tan “bueno” es un algoritmo es contando número de operaciones que este debe realizar para resolver un problema cuya entrada tiene un tamaño  $N$ . Es decir, calcular un  $f(N)$

# Solución 1

```
read A
for i = 2 to A-1:
    if A % i = 0:
        print i
```

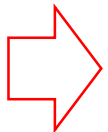


Si A es 100, imprimiría:

2, 4, 5, 10, 20, 25, 50

Número de operaciones:

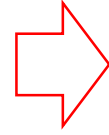
$$1 + (A - 2)(1 + 1') \rightarrow f(N) = 2N - 3$$



Si A es 1.000.000, haría 1.999.997 operaciones

# Solución 2

```
read A
for i = 2 to A/2:
    if A % i = 0:
        print i
```

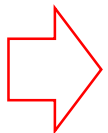


Si A es 100, imprimiría:

2, 4, 5, 10, 20, 25, 50

Número de operaciones:

$$1 + \left(\frac{A}{2} - 1\right) (1 + 1') \rightarrow f(N) = N - 1$$



Si A es 1.000.000, haría 999.999 operaciones

# Solución 3

```
read A
for i = 2 to  $\sqrt{A}$ :
    if A % i = 0:
        if A/i  $\neq$  i:
            print i, A/i
        else:
            print i
```

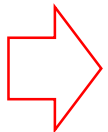


Si A es 100, imprimiría:

2, 50, 4, 25, 5, 20, 10

Número de operaciones:

$$1 + (\sqrt{A} - 1)(1 + 1' + 1'') \rightarrow f(N) = 3\sqrt{N} - 2$$



Si A es 1.000.000, haría 2.998 operaciones