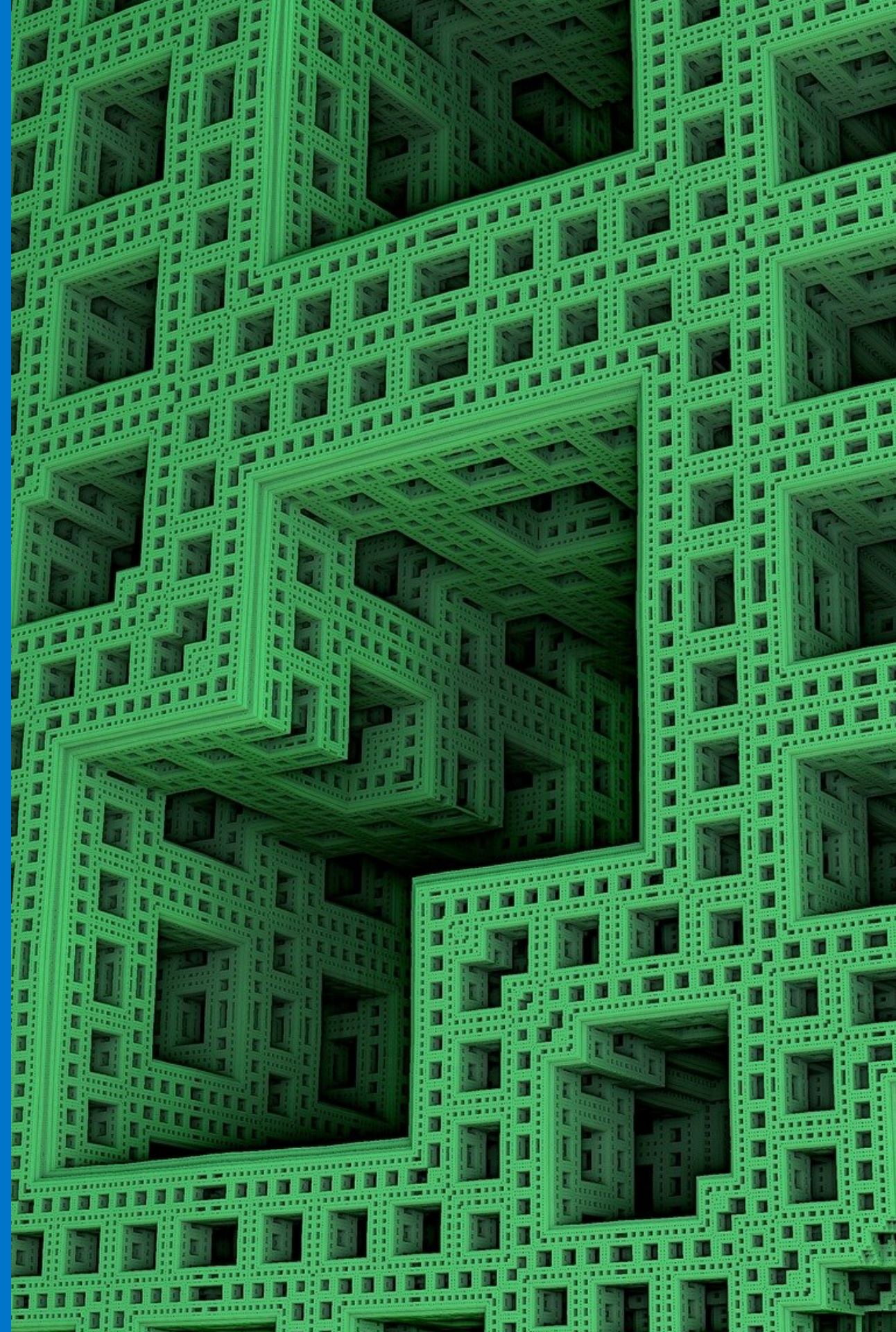


Algoritmos:
Eficiencia, avances y
la peor forma de
ganar un millón de
dolares.

Prof. Arturo Merino

UOH Universidad
de O'Higgins

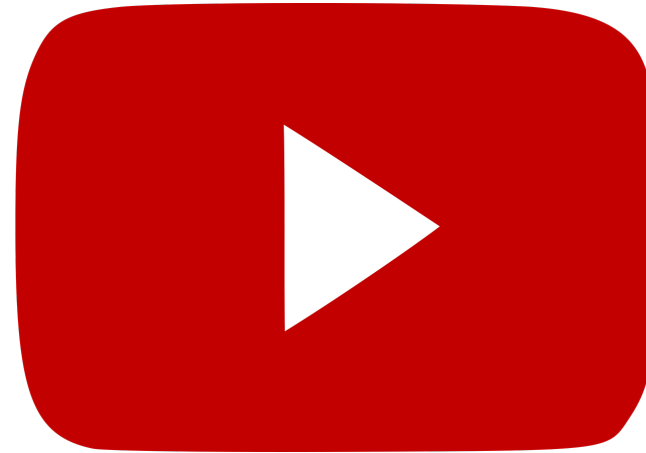


■ ¿Qué es un algoritmo?

- ¿Qué son?

¿Qué es un algoritmo?

- ¿Qué son?



¿Qué es un algoritmo?

- ¿Qué son?
- **Algoritmo:** secuencia precisa de pasos para resolver un *problema computacional*.

¿Qué es un algoritmo?

- ¿Qué son?
- **Algoritmo:** secuencia precisa de pasos para resolver un *problema computacional*.
- **Problema computacional:** recibe datos como *entrada* y retorna datos *como salida*.

¿Qué es un algoritmo?

- ¿Qué son?
- **Algoritmo:** secuencia precisa de pasos para resolver un *problema computacional*.
- **Problema computacional:** recibe datos como *entrada* y retorna datos *como* salida.
 - Receta de cocina



Algoritmo



¿Qué es un algoritmo?

- ¿Qué son?
- **Algoritmo:** secuencia precisa de pasos para resolver un *problema computacional*.
- **Problema computacional:** recibe datos como *entrada* y retorna datos *como* salida.
 - Receta de cocina
 - Ordenar una lista



¿Qué es un algoritmo?

- ¿Qué son?
- **Algoritmo:** secuencia precisa de pasos para resolver un *problema computacional*.
- **Problema computacional:** recibe datos como *entrada* y retorna datos *como* salida.
 - Receta de cocina
 - Ordenar una lista
 - Multiplicar dos números

5034

2301

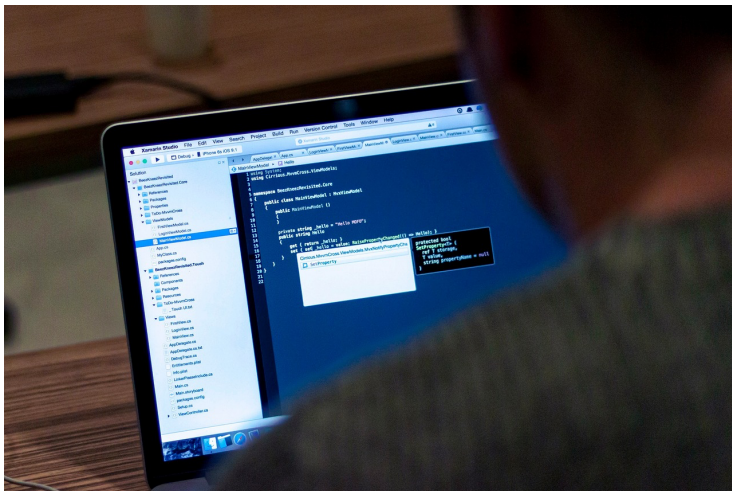
Algoritmo



11583234

¿Qué es un algoritmo?

- ¿Qué son?
- **Algoritmo:** secuencia precisa de pasos para resolver un *problema computacional*.
- **Problema computacional:** recibe datos como *entrada* y retorna datos *como* salida.
 - Receta de cocina
 - Ordenar una lista
 - Multiplicar dos números
 - Recomendar videos

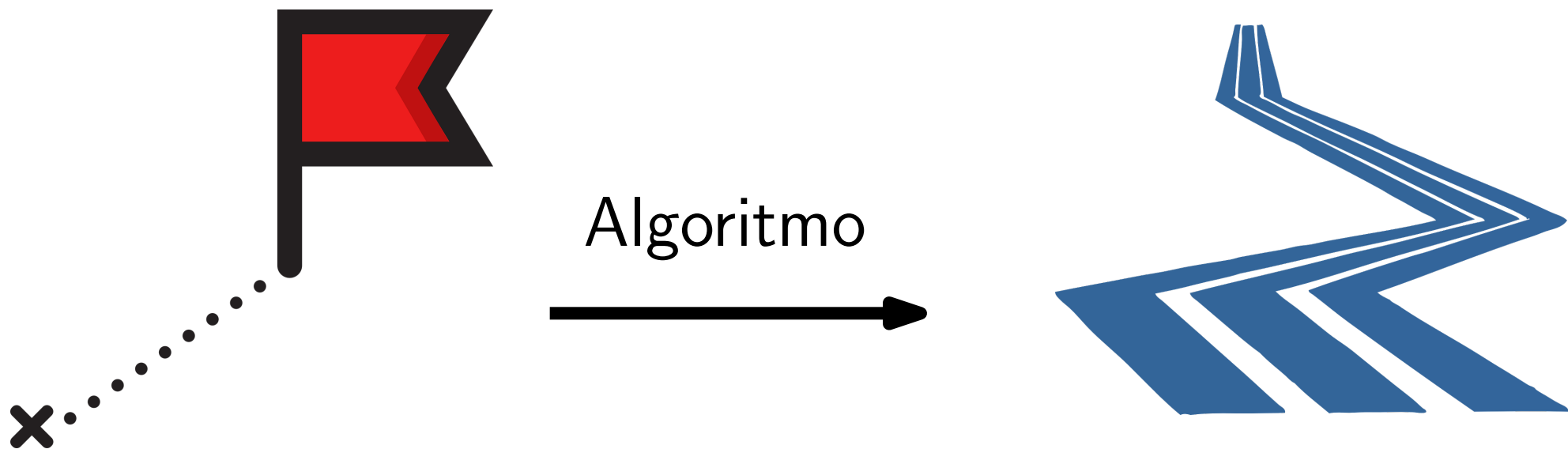


Algoritmo



¿Qué es un algoritmo?

- ¿Qué son?
- **Algoritmo:** secuencia precisa de pasos para resolver un *problema computacional*.
- **Problema computacional:** recibe datos como *entrada* y retorna datos *como* salida.
 - Receta de cocina
 - Ordenar una lista
 - Multiplicar dos números
 - Recomendar videos
 - Encontrar la ruta más rápida



¿Qué es un algoritmo?

- ¿Qué son?
- **Algoritmo:** secuencia precisa de pasos para resolver un *problema computacional*.
- **Problema computacional:** recibe datos como *entrada* y retorna datos *como* salida.
 - Receta de cocina
 - Ordenar una lista
 - Multiplicar dos números
 - Recomendar videos
 - Encontrar la ruta más rápida
 - Mucho, mucho más



Algoritmo



¿Qué es un algoritmo?

- ¿Qué son?
- **Algoritmo:** secuencia precisa de pasos para resolver un *problema computacional*.
- **Problema computacional:** recibe datos como *entrada* y retorna datos *como* salida.
 - Receta de cocina
 - Ordenar una lista
 - Multiplicar dos números
 - Recomendar videos
 - Encontrar la ruta más rápida
 - Mucho, mucho más
- Mismo problema, varios algoritmos.

¿Qué es un algoritmo?

- ¿Qué son?
- **Algoritmo:** secuencia precisa de pasos para resolver un *problema computacional*.
- **Problema computacional:** recibe datos como *entrada* y retorna datos *como* salida.
 - Receta de cocina
 - Ordenar una lista
 - Multiplicar dos números
 - Recomendar videos
 - Encontrar la ruta más rápida
 - Mucho, mucho más
- Mismo problema, varios algoritmos.
- Opciones, decisiones.

■ Algoritmos como objeto de estudio

- **Diseño y análisis de algoritmos:**

Algoritmos como objeto de estudio

- **Diseño y análisis de algoritmos:**
 - ¿Qué tan eficiente pueden ser los algoritmos que diseñamos?

Algoritmos como objeto de estudio

- **Diseño y análisis de algoritmos:**
 - ¿Qué tan eficiente pueden ser los algoritmos que diseñamos?
- **Eficiencia:** Lograr resolver el problema, usando la menor cantidad de recursos.

Algoritmos como objeto de estudio

- **Diseño y análisis de algoritmos:**
 - ¿Qué tan eficiente pueden ser los algoritmos que diseñamos?
- **Eficiencia:** Lograr resolver el problema, usando la menor cantidad de recursos.
 - Tiempo

Algoritmos como objeto de estudio

- **Diseño y análisis de algoritmos:**
 - ¿Qué tan eficiente pueden ser los algoritmos que diseñamos?
- **Eficiencia:** Lograr resolver el problema, usando la menor cantidad de recursos.
 - Tiempo
 - Energía

Algoritmos como objeto de estudio

- **Diseño y análisis de algoritmos:**
 - ¿Qué tan eficiente pueden ser los algoritmos que diseñamos?
- **Eficiencia:** Lograr resolver el problema, usando la menor cantidad de recursos.
 - Tiempo
 - Energía
 - Memoria

Algoritmos como objeto de estudio

- **Diseño y análisis de algoritmos:**
 - ¿Qué tan eficiente pueden ser los algoritmos que diseñamos?
- **Eficiencia:** Lograr resolver el problema, usando la menor cantidad de recursos.
 - Tiempo
 - Energía
 - Memoria
 - Servidores

Algoritmos como objeto de estudio

- **Diseño y análisis de algoritmos:**
 - ¿Qué tan eficiente pueden ser los algoritmos que diseñamos?
- **Eficiencia:** Lograr resolver el problema, usando la menor cantidad de recursos.
 - Tiempo
 - Energía
 - Memoria
 - Servidores

■ ¿Cómo medir la eficiencia?

- **Sencillo:** Medir con un reloj cuánto demora.

■ ¿Cómo medir la eficiencia?

- **Sencillo:** Medir con un reloj cuánto demora.
- **Problema:** Muy sensible. Depende de

■ ¿Cómo medir la eficiencia?

- **Sencillo:** Medir con un reloj cuánto demora.
- **Problema:** Muy sensible. Depende de
 - Computador

¿Cómo medir la eficiencia?

- **Sencillo:** Medir con un reloj cuánto demora.
- **Problema:** Muy sensible. Depende de
 - Computador
 - Lenguaje de programación

¿Cómo medir la eficiencia?

- **Sencillo:** Medir con un reloj cuánto demora.
- **Problema:** Muy sensible. Depende de
 - Computador
 - Temperatura
 - Lenguaje de programación

¿Cómo medir la eficiencia?

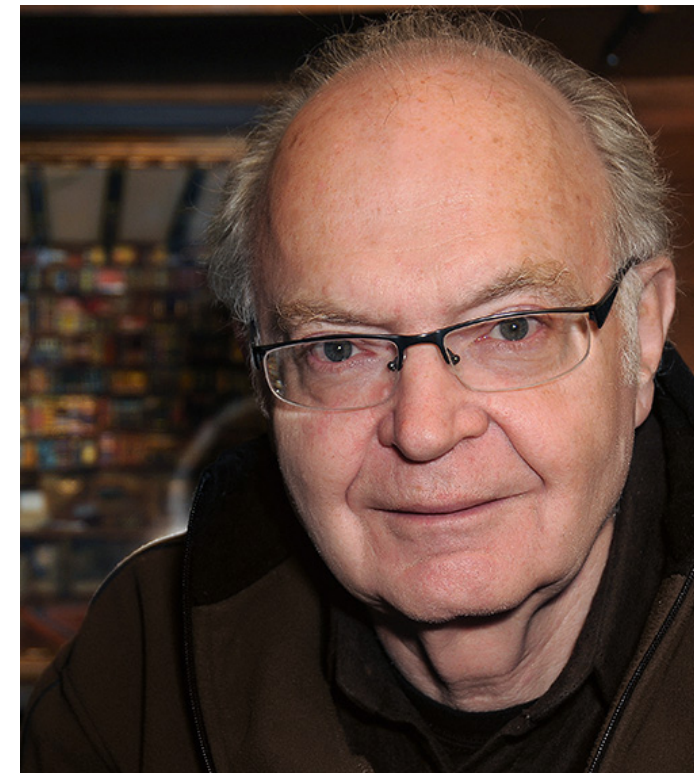
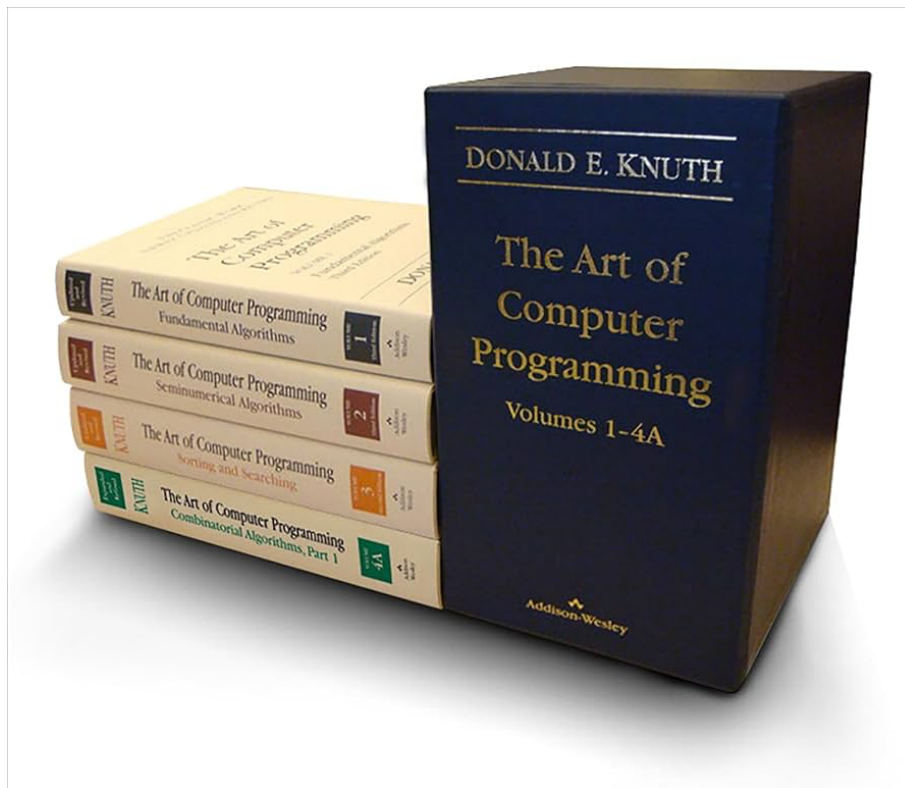
- **Sencillo:** Medir con un reloj cuánto demora.
- **Problema:** Muy sensible. Depende de
 - Computador
 - Temperatura
 - Lenguaje de programación
 - Implementación

¿Cómo medir la eficiencia?

- **Sencillo:** Medir con un reloj cuánto demora.
- **Problema:** Muy sensible. Depende de
 - Computador
 - Temperatura
 - Lenguaje de programación
 - Implementación
- **Análisis de algoritmos:** Número de operaciones.

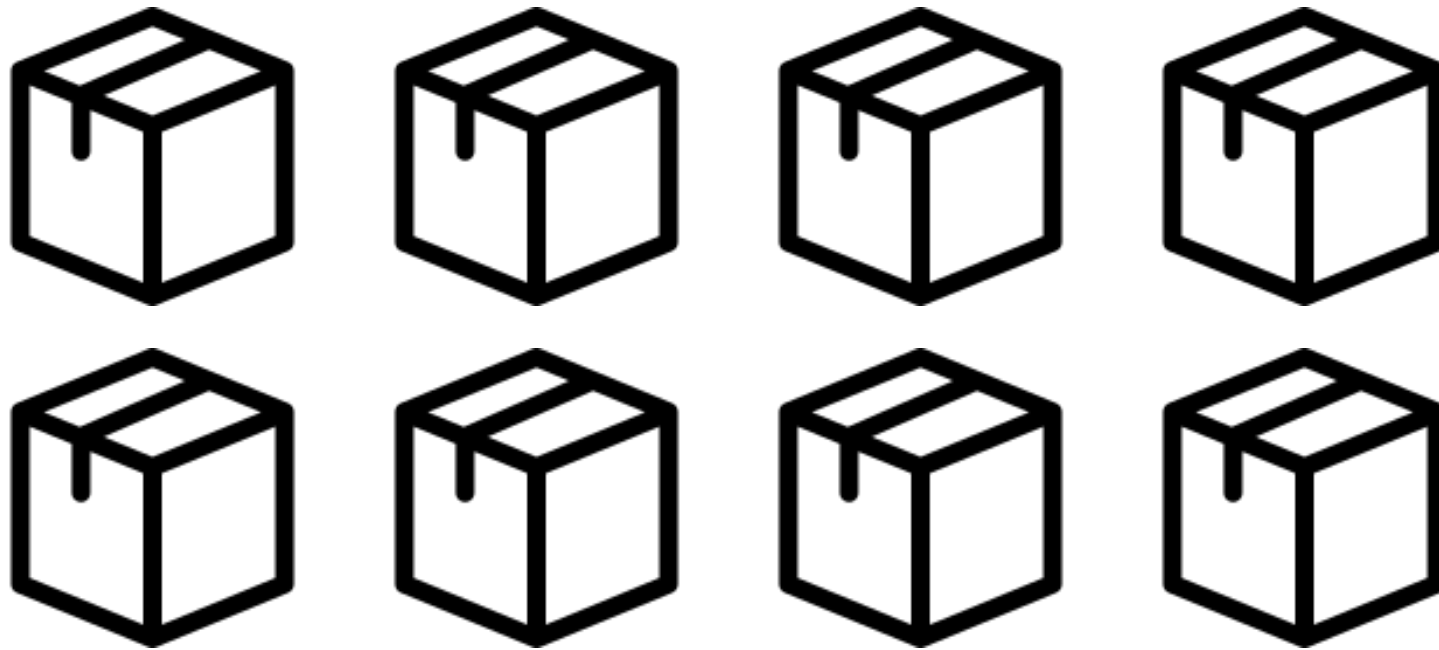
¿Cómo medir la eficiencia?

- **Sencillo:** Medir con un reloj cuánto demora.
- **Problema:** Muy sensible. Depende de
 - Computador
 - Temperatura
 - Lenguaje de programación
 - Implementación
- **Análisis de algoritmos:** Número de operaciones.



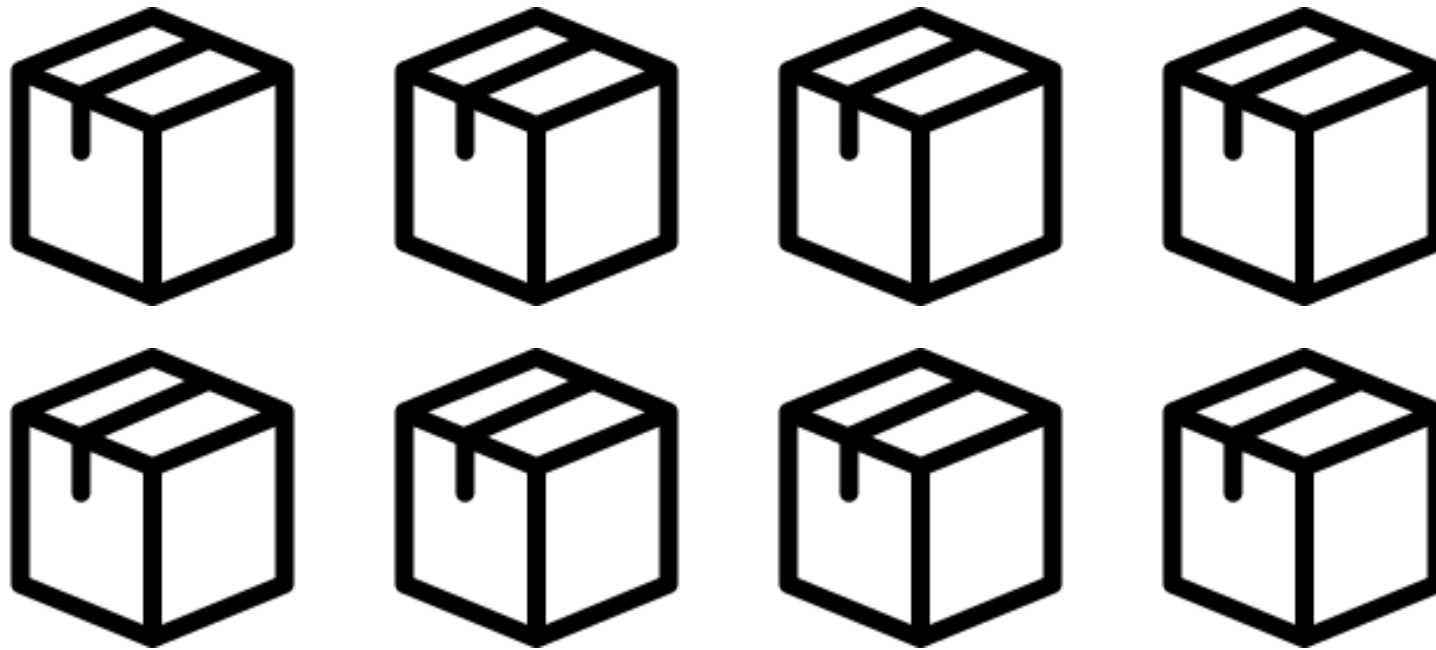
Un primer problema algorítmico

- **Entrada:** 8 cajas con pesos secretos adentro



Un primer problema algorítmico

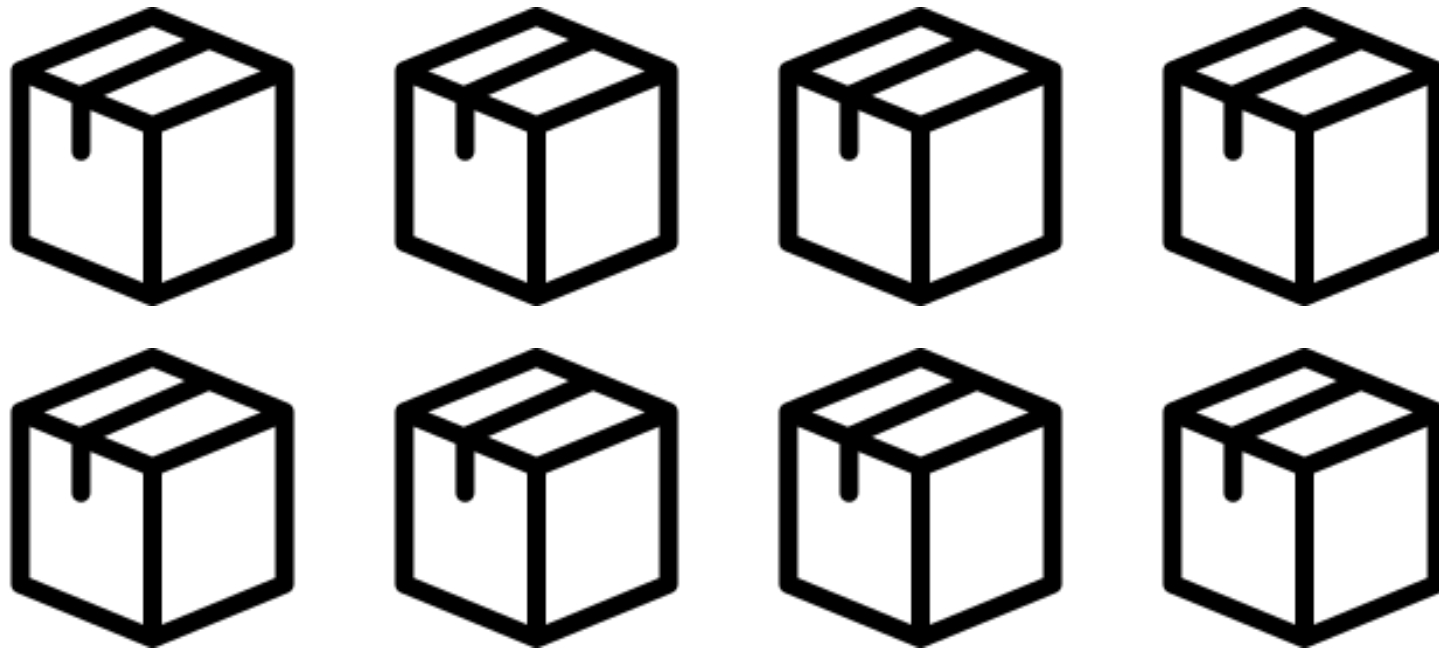
- **Entrada:** 8 cajas con pesos secretos adentro



- **Salida:** Elegir la caja de peso máximo.

Un primer problema algorítmico

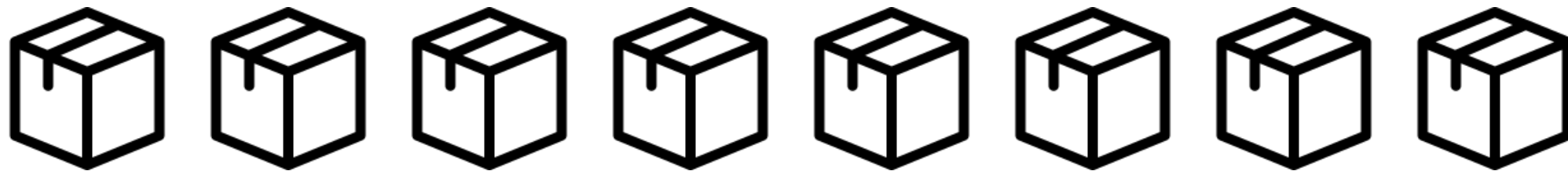
- **Entrada:** 8 cajas con pesos secretos adentro



- **Salida:** Elegir la caja de peso máximo.
- **Operaciones:** Comparar el peso de dos cajas.

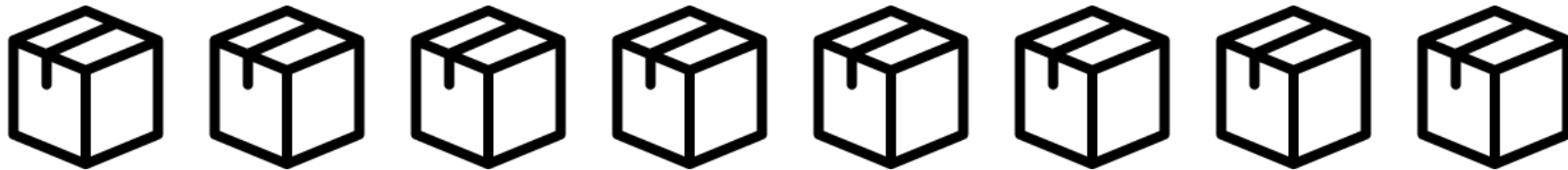


Solución problema 1



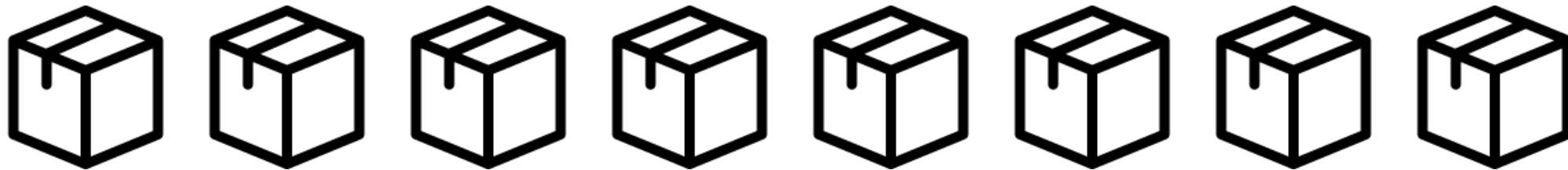
Solución problema 1

- Comparar las primeras dos cajas, quedarse con la mejor.



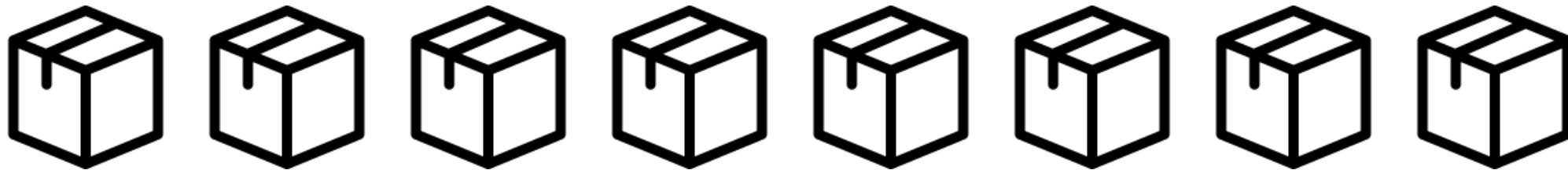
Solución problema 1

- Comparar las primeras dos cajas, quedarse con la mejor.
- Comparar la mejor actual con la siguiente, quedarse con la mejor.



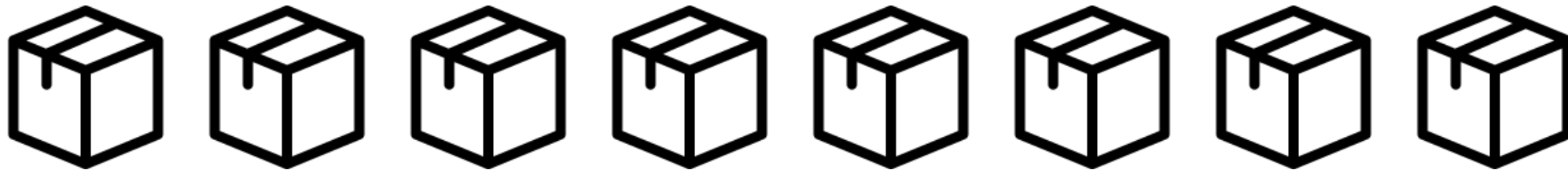
Solución problema 1

- Comparar las primeras dos cajas, quedarse con la mejor.
- Comparar la mejor actual con la siguiente, quedarse con la mejor.
- Repetir el paso 2.



Solución problema 1

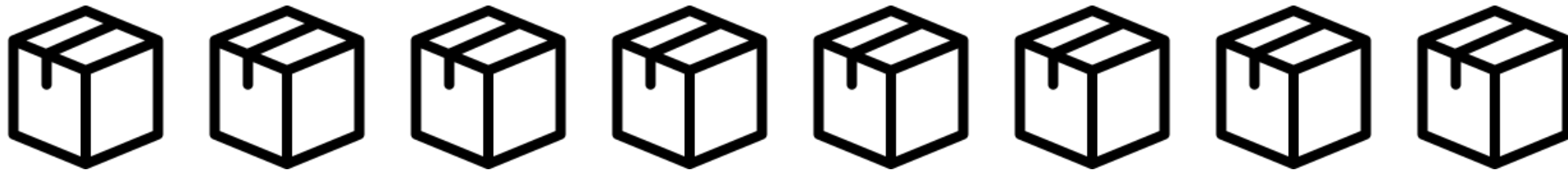
- Comparar las primeras dos cajas, quedarse con la mejor.
- Comparar la mejor actual con la siguiente, quedarse con la mejor.
- Repetir el paso 2.



- ¿Cuántas comparaciones?

Solución problema 1

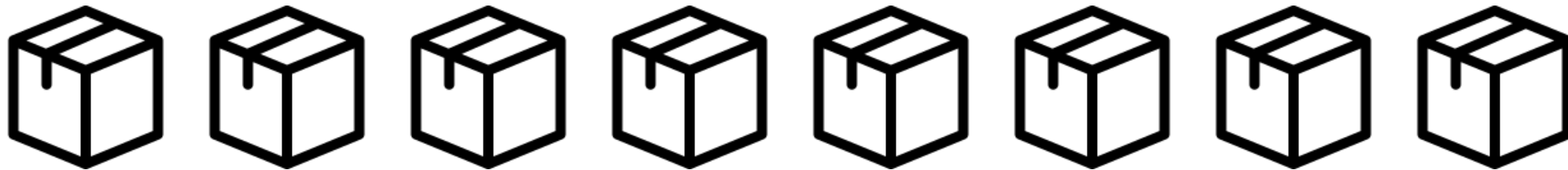
- Comparar las primeras dos cajas, quedarse con la mejor.
- Comparar la mejor actual con la siguiente, quedarse con la mejor.
- Repetir el paso 2.



- ¿Cuántas comparaciones?
 - R: 7.

Solución problema 1

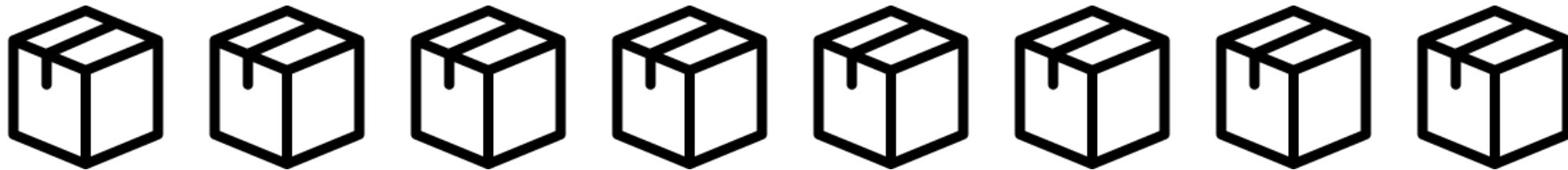
- Comparar las primeras dos cajas, quedarse con la mejor.
- Comparar la mejor actual con la siguiente, quedarse con la mejor.
- Repetir el paso 2.



- ¿Cuántas comparaciones?
 - R: 7.
- ¿ n cajas?

Solución problema 1

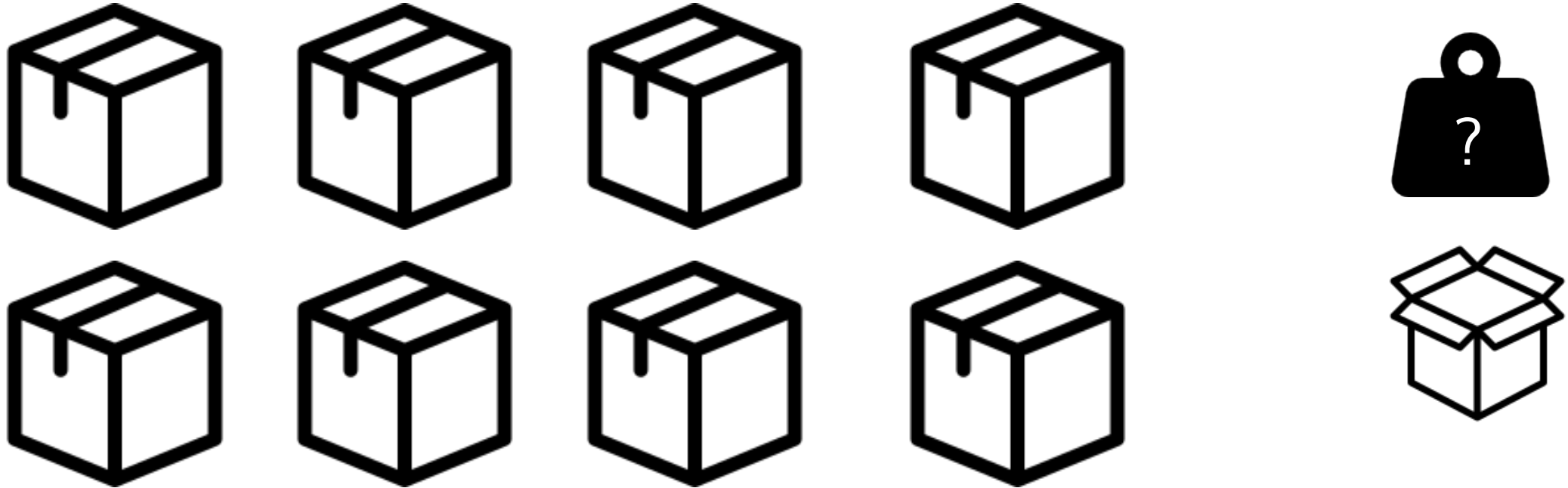
- Comparar las primeras dos cajas, quedarse con la mejor.
- Comparar la mejor actual con la siguiente, quedarse con la mejor.
- Repetir el paso 2.



- ¿Cuántas comparaciones?
 - R: 7.
- ¿ n cajas?
 - R: $n - 1$.

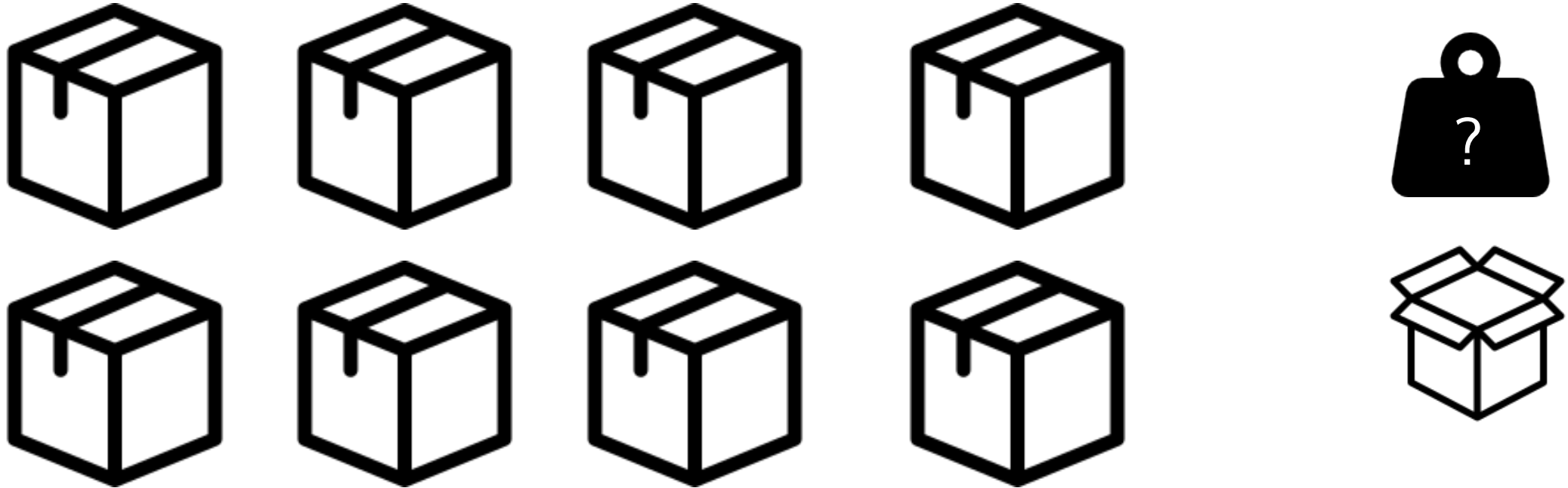
Segundo problema: Ordenar

- **Entrada:** 8 cajas con contenidos secretos adentro



Segundo problema: Ordenar

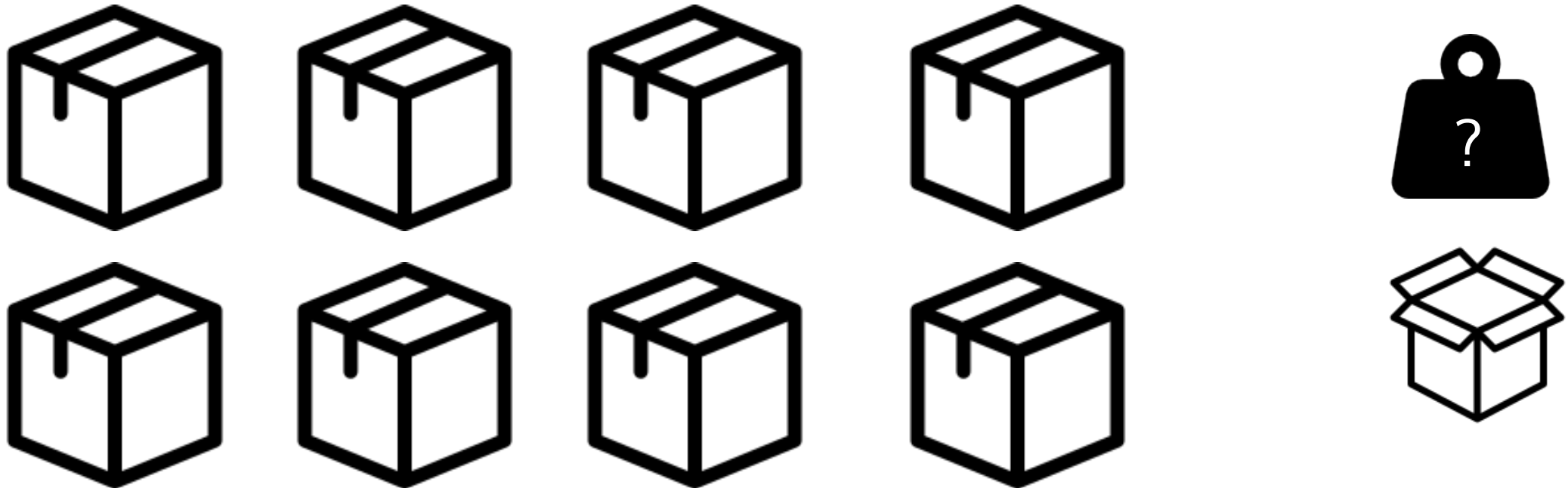
- **Entrada:** 8 cajas con contenidos secretos adentro



- **Salida:** Ordenar las cajas de mayor a menor peso.

Segundo problema: Ordenar

- **Entrada:** 8 cajas con contenidos secretos adentro

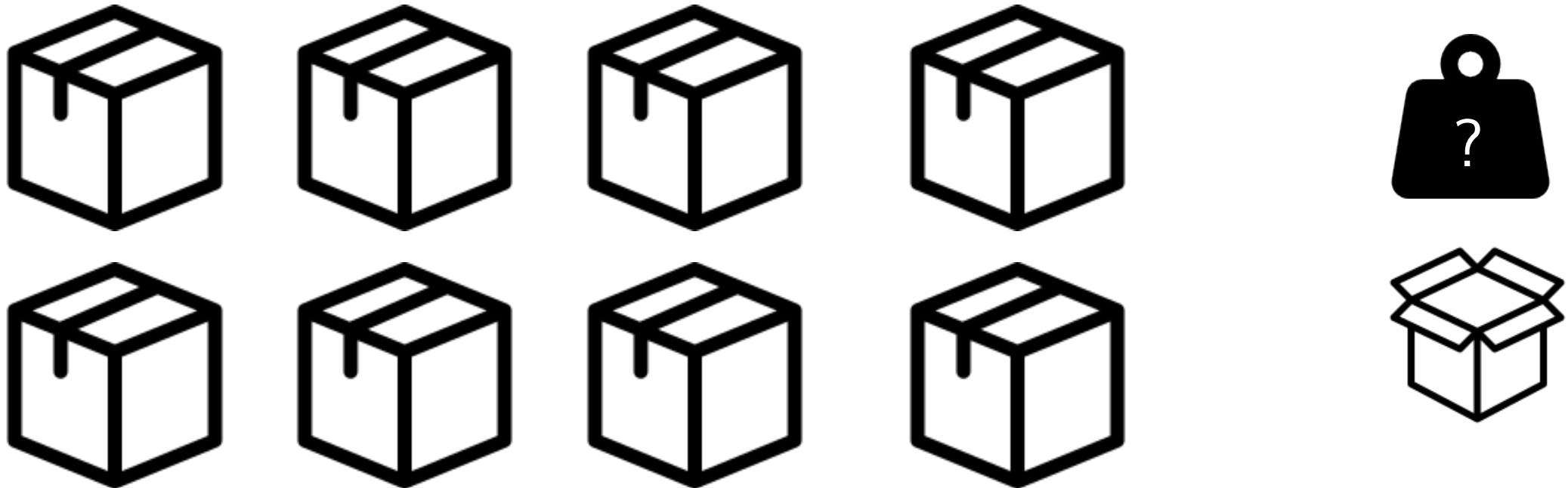


- **Salida:** Ordenar las cajas de mayor a menor peso.
- **Operaciones:** Comparar el peso de dos cajas.



Segundo problema: Ordenar

- **Entrada:** 8 cajas con contenidos secretos adentro

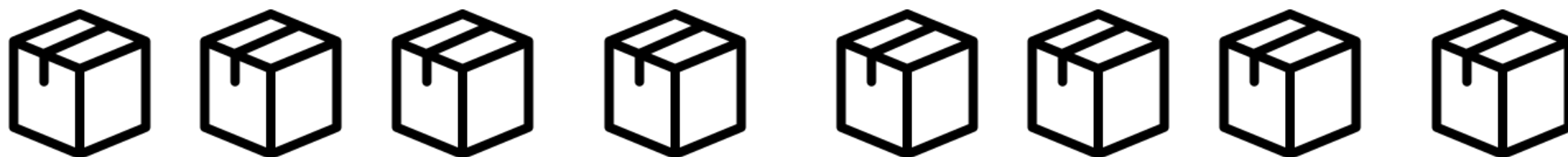


- **Salida:** Ordenar las cajas de mayor a menor peso.
- **Operaciones:** Comparar el peso de dos cajas.



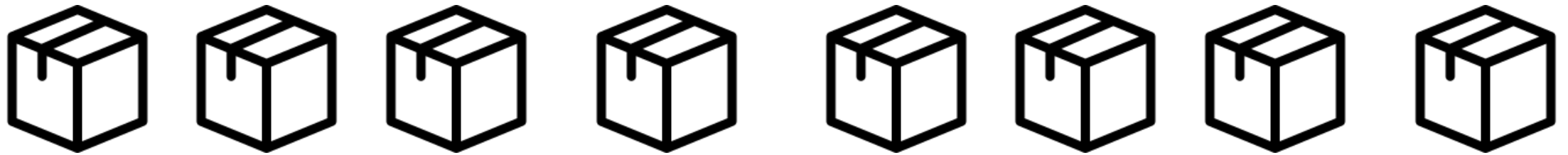
- ¡Problema real que resuelven los computadores todos los días!

Solución problema 2



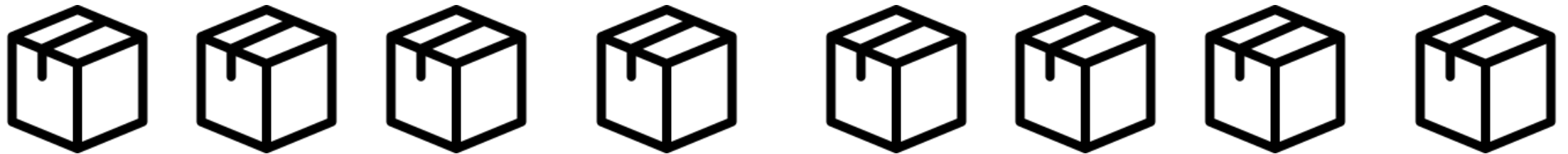
Solución problema 2

- Encontrar el máximo, poner la caja en su lugar.



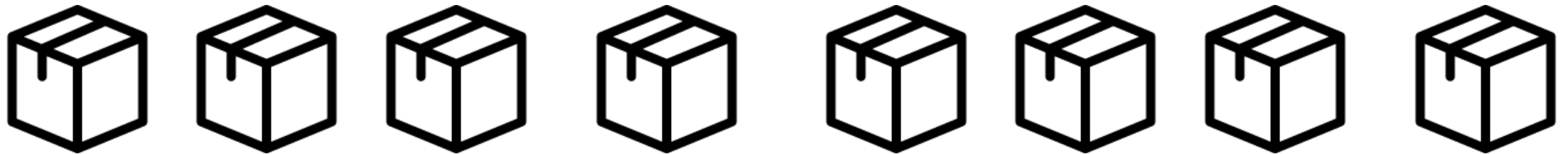
Solución problema 2

- Encontrar el máximo, poner la caja en su lugar.
- Encontrar el máximo de las cajas restantes, poner la caja que queda en el siguiente lugar.



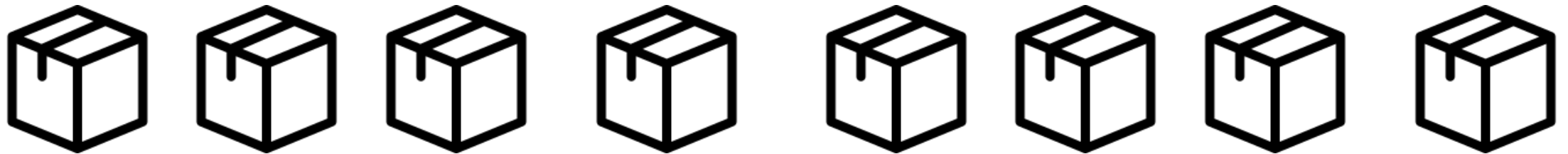
Solución problema 2

- Encontrar el máximo, poner la caja en su lugar.
- Encontrar el máximo de las cajas restantes, poner la caja que queda en el siguiente lugar.
- Repetir el paso 2.



Solución problema 2

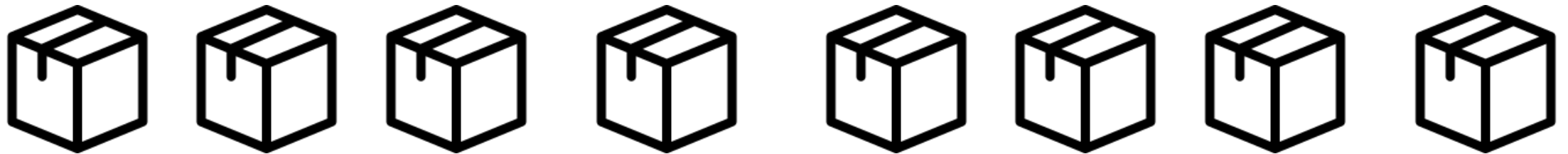
- Encontrar el máximo, poner la caja en su lugar.
- Encontrar el máximo de las cajas restantes, poner la caja que queda en el siguiente lugar.
- Repetir el paso 2.



- ¿Cuántas comparaciones?

Solución problema 2

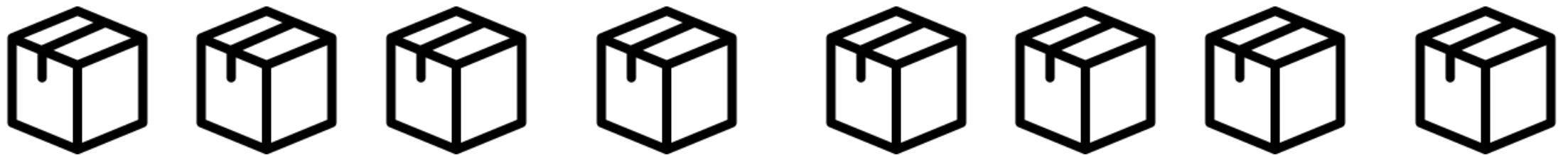
- Encontrar el máximo, poner la caja en su lugar.
- Encontrar el máximo de las cajas restantes, poner la caja que queda en el siguiente lugar.
- Repetir el paso 2.



- ¿Cuántas comparaciones?
 - R: $7 + 6 + 5 + 4 + 3 + 2 + 1 = 28$.

Solución problema 2

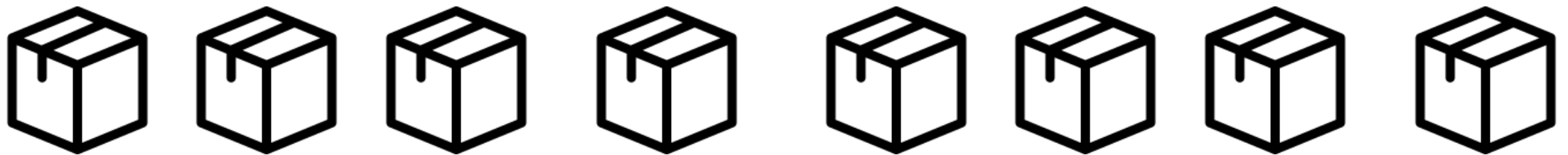
- Encontrar el máximo, poner la caja en su lugar.
- Encontrar el máximo de las cajas restantes, poner la caja que queda en el siguiente lugar.
- Repetir el paso 2.



- ¿Cuántas comparaciones?
 - R: $7 + 6 + 5 + 4 + 3 + 2 + 1 = 28$.
- ¿ n cajas?

Solución problema 2

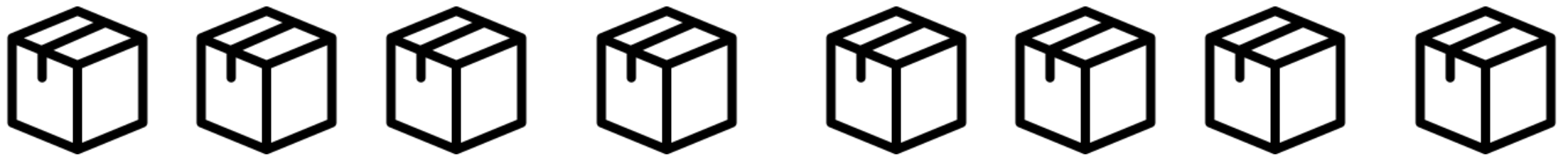
- Encontrar el máximo, poner la caja en su lugar.
- Encontrar el máximo de las cajas restantes, poner la caja que queda en el siguiente lugar.
- Repetir el paso 2.



- ¿Cuántas comparaciones?
 - R: $7 + 6 + 5 + 4 + 3 + 2 + 1 = 28$.
- ¿ n cajas?
 - R: $\frac{(n-1)(n-2)}{2} \approx \frac{n^2}{2}$.

Solución problema 2

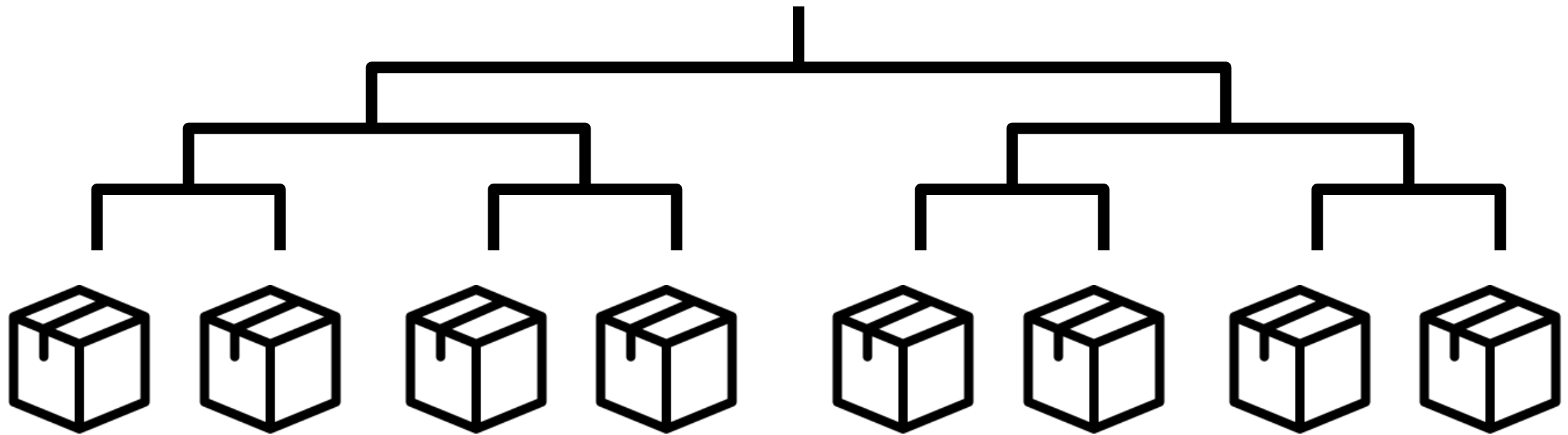
- Encontrar el máximo, poner la caja en su lugar.
- Encontrar el máximo de las cajas restantes, poner la caja que queda en el siguiente lugar.
- Repetir el paso 2.



- ¿Cuántas comparaciones?
 - R: $7 + 6 + 5 + 4 + 3 + 2 + 1 = 28$.
- ¿ n cajas?
 - R: $\frac{(n-1)(n-2)}{2} \approx \frac{n^2}{2}$.
- Esto se conoce como SelectionSort.

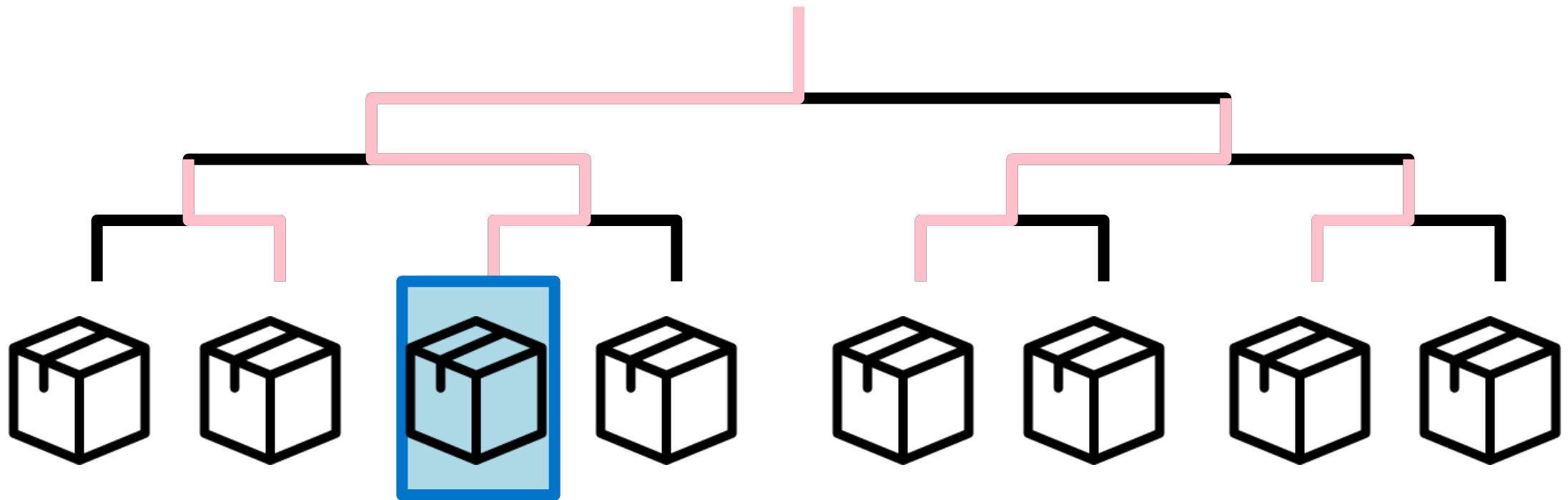
¿Menos comparaciones?

- Otra manera de encontrar el máximo



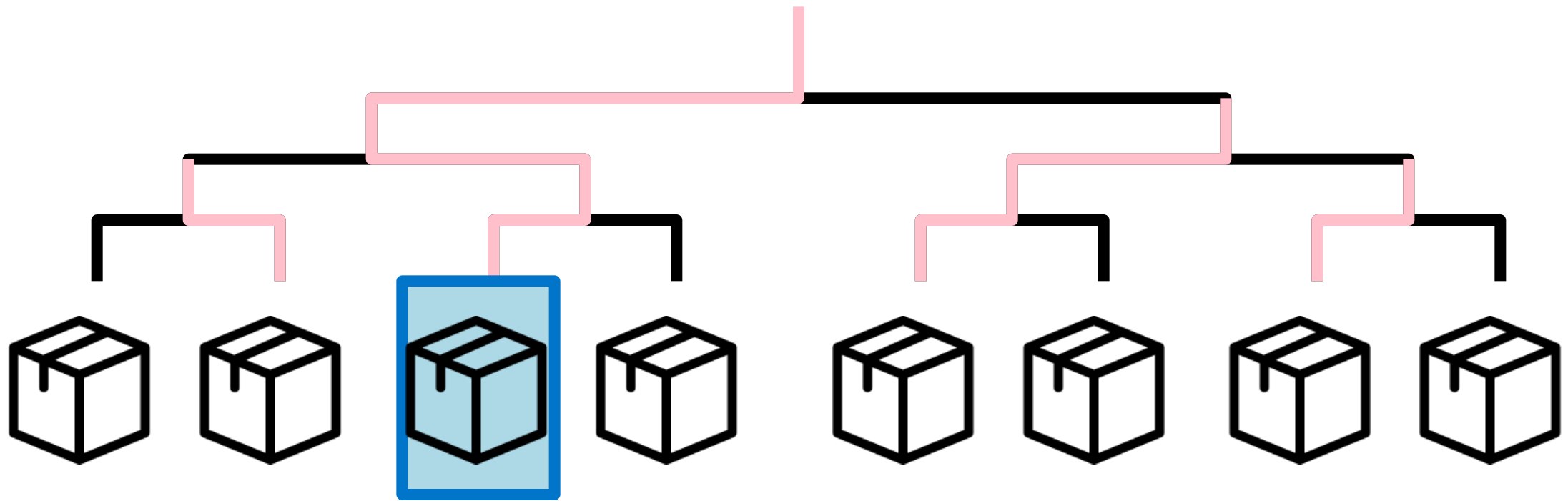
¿Menos comparaciones?

- Otra manera de encontrar el máximo



¿Menos comparaciones?

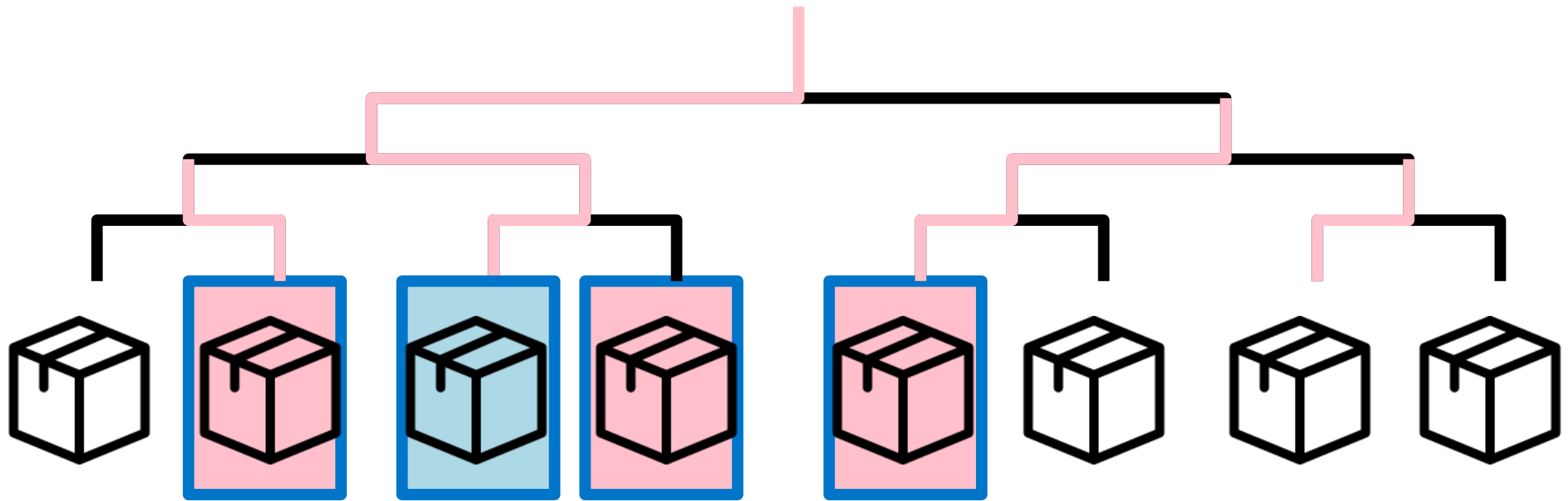
- Otra manera de encontrar el máximo



- Se puede *reusar* información para encontrar la segunda caja.

¿Menos comparaciones?

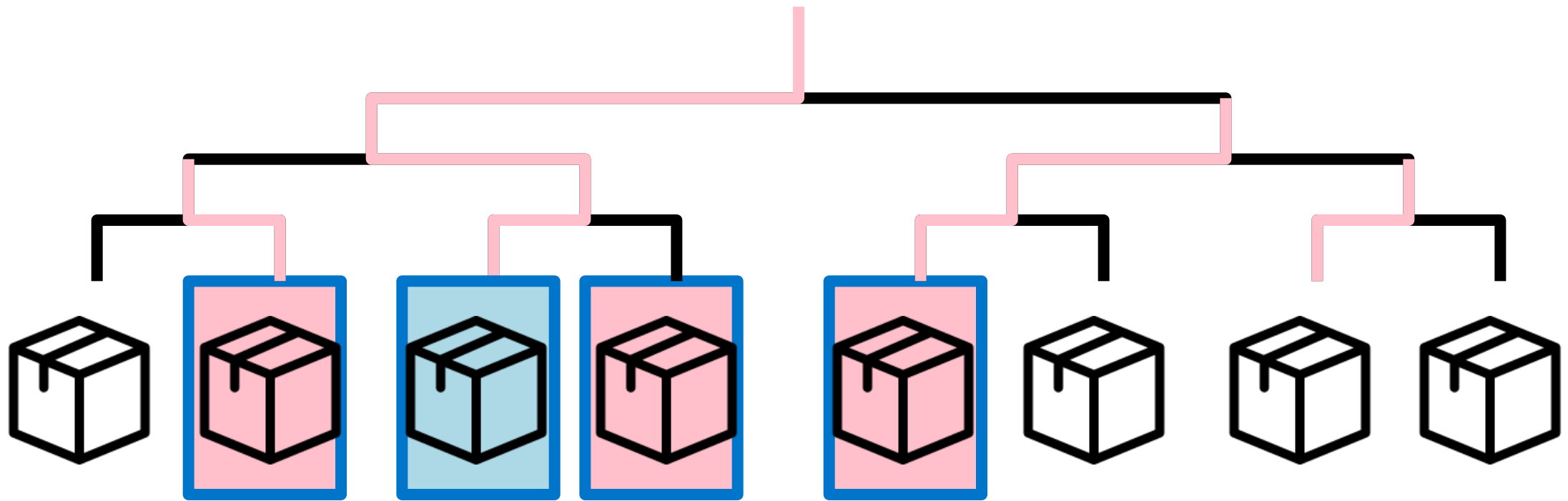
- Otra manera de encontrar el máximo



- Se puede *reusar* información para encontrar la segunda caja.

¿Menos comparaciones?

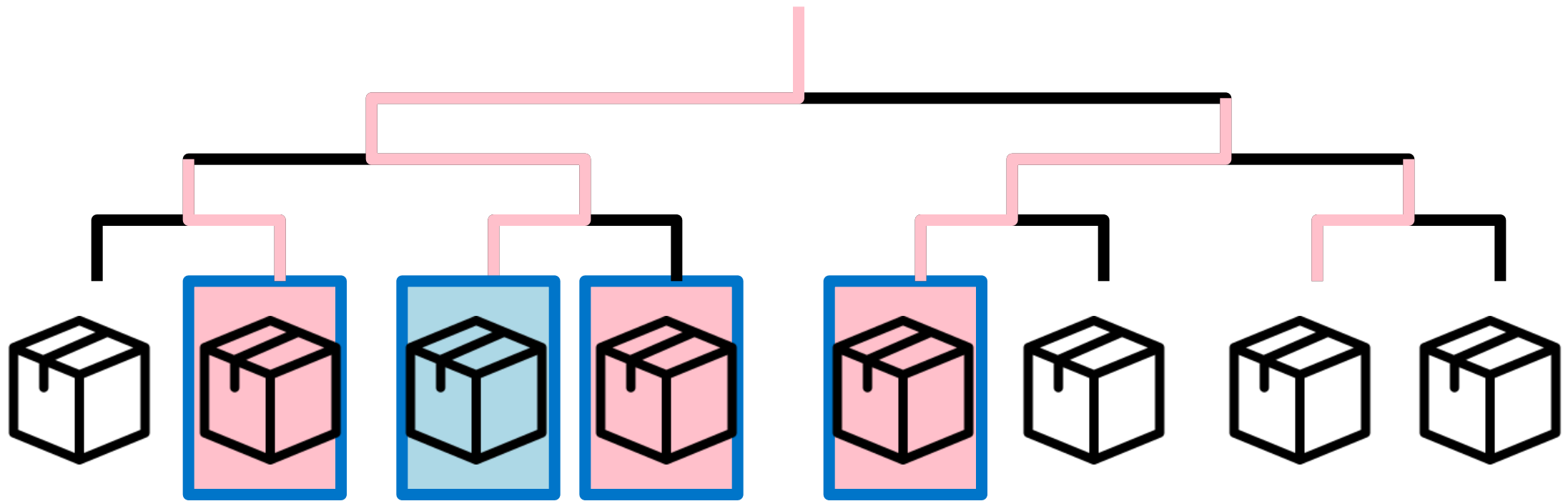
- Otra manera de encontrar el máximo



- Se puede *reusar* información para encontrar la segunda caja.
- Sólo 2 comparaciones para la segunda caja, en vez de 6.
- Estos razonamientos llevan a HeapSort.

¿Menos comparaciones?

- Otra manera de encontrar el máximo



- Se puede *reusar* información para encontrar la segunda caja.
- Sólo 2 comparaciones para la segunda caja, en vez de 6.
- Estos razonamientos llevan a HeapSort.
- ¿Con n cajas? Usa approx. $n \log_2 n$ comparaciones.

■ Comparando los metodos

- ¿Cuánto afecta esto al tiempo que demora en un computador?

Comparando los metodos

- ¿Cuánto afecta esto al tiempo que demora en un computador?



Leftraru, el supercomputador
más rápido de Chile

$\sim 7 \cdot 10^{13}$ comparaciones por
segundo

Comparando los metodos

- ¿Cuánto afecta esto al tiempo que demora en un computador?



v/s

Leftraru, el supercomputador
más rápido de Chile

$\sim 7 \cdot 10^{13}$ comparaciones por
segundo

Comparando los metodos

- ¿Cuánto afecta esto al tiempo que demora en un computador?



Leftraru, el supercomputador más rápido de Chile

$\sim 7 \cdot 10^{13}$ comparaciones por segundo

v/s



Playstation 5

$\sim 1 \cdot 10^9$ comparaciones por segundo

Comparando los metodos

- Al ordenar 100 millones de elementos. Leftraru con SelectionSort ($n^2/2$) toma:

$$\frac{(10^8)^2/2 \text{ [comparaciones]}}{7 \cdot 10^{13} \text{ [comparaciones/s]}} = 0,7 \cdot 10^3 \text{ [s]} \approx 4 \text{ horas}$$

Comparando los metodos

- Al ordenar 100 millones de elementos. Leftraru con SelectionSort ($n^2/2$) toma:

$$\frac{(10^8)^2/2 \text{ [comparaciones]}}{7 \cdot 10^{13} \text{ [comparaciones/s]}} = 0,7 \cdot 10^3 \text{ [s]} \approx 4 \text{ horas}$$

- Playstation 5 con HeapSort ($n \log_2 n$) toma:

$$\frac{10^8 \log_2(10^8) \text{ [comparaciones]}}{1 \cdot 10^9 \text{ [comparaciones/s]}} \approx 2,6 \cdot 10^0 \text{ [s]} = 2,6 \text{ segundos}$$

Comparando los metodos

- Al ordenar 100 millones de elementos. Leftraru con SelectionSort ($n^2/2$) toma:

$$\frac{(10^8)^2/2 \text{ [comparaciones]}}{7 \cdot 10^{13} \text{ [comparaciones/s]}} = 0,7 \cdot 10^3 \text{ [s]} \approx 4 \text{ horas}$$

- Playstation 5 con HeapSort ($n \log_2 n$) toma:

$$\frac{10^8 \log_2(10^8) \text{ [comparaciones]}}{1 \cdot 10^9 \text{ [comparaciones/s]}} \approx 2,6 \cdot 10^0 \text{ [s]} = 2,6 \text{ segundos}$$

- La Playstation 5 con HeapSort es más rápida desde approx. el millón y medio de elementos.

Conclusiones comparación

- **Ambas Importantes:**
Buenos algoritmos y buen hardware

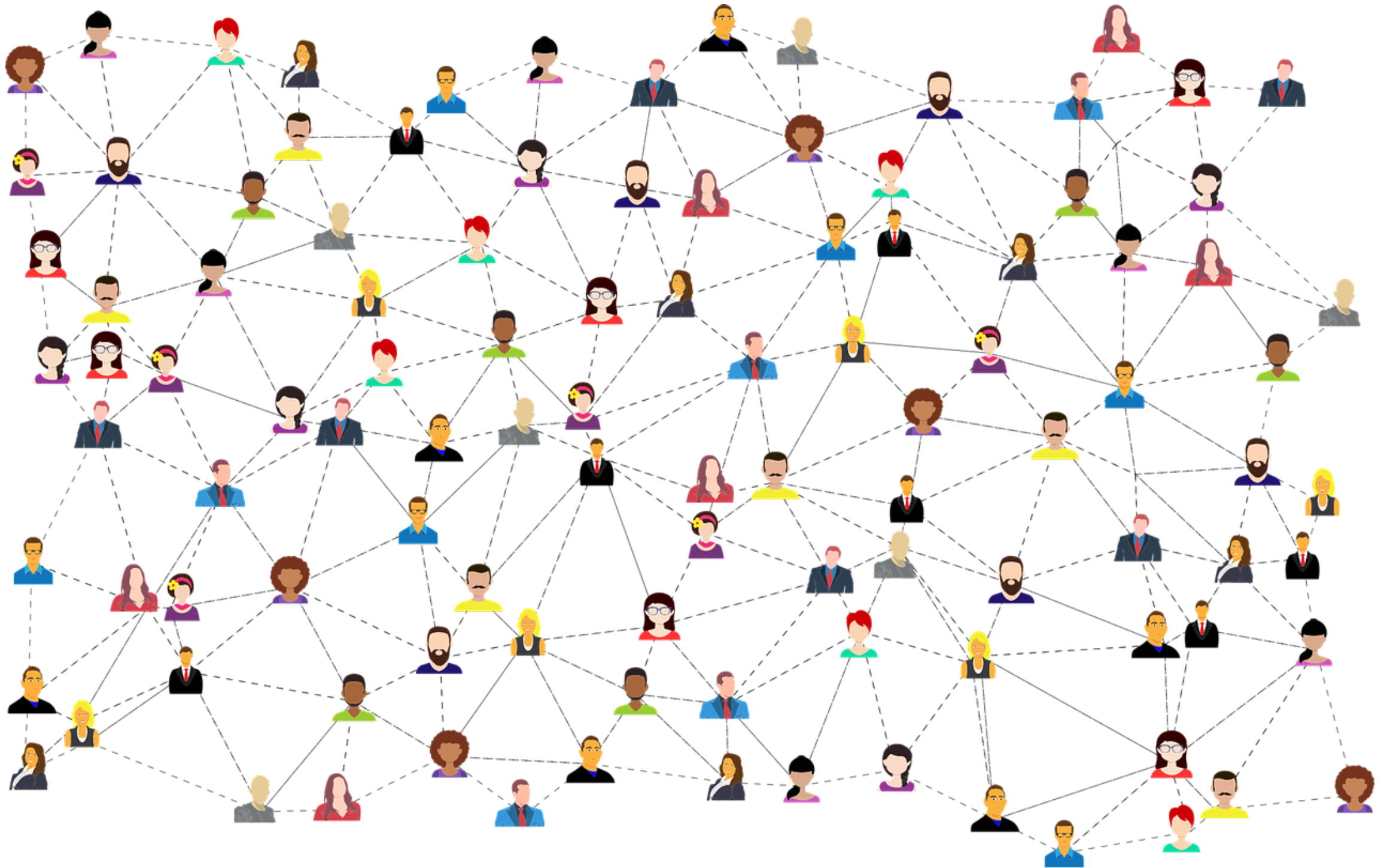
Conclusiones comparación

- **Ambas Importantes:**
Buenos algoritmos y buen hardware
- **Pero...**
Buenos algoritmos pesan más con datos más grandes.

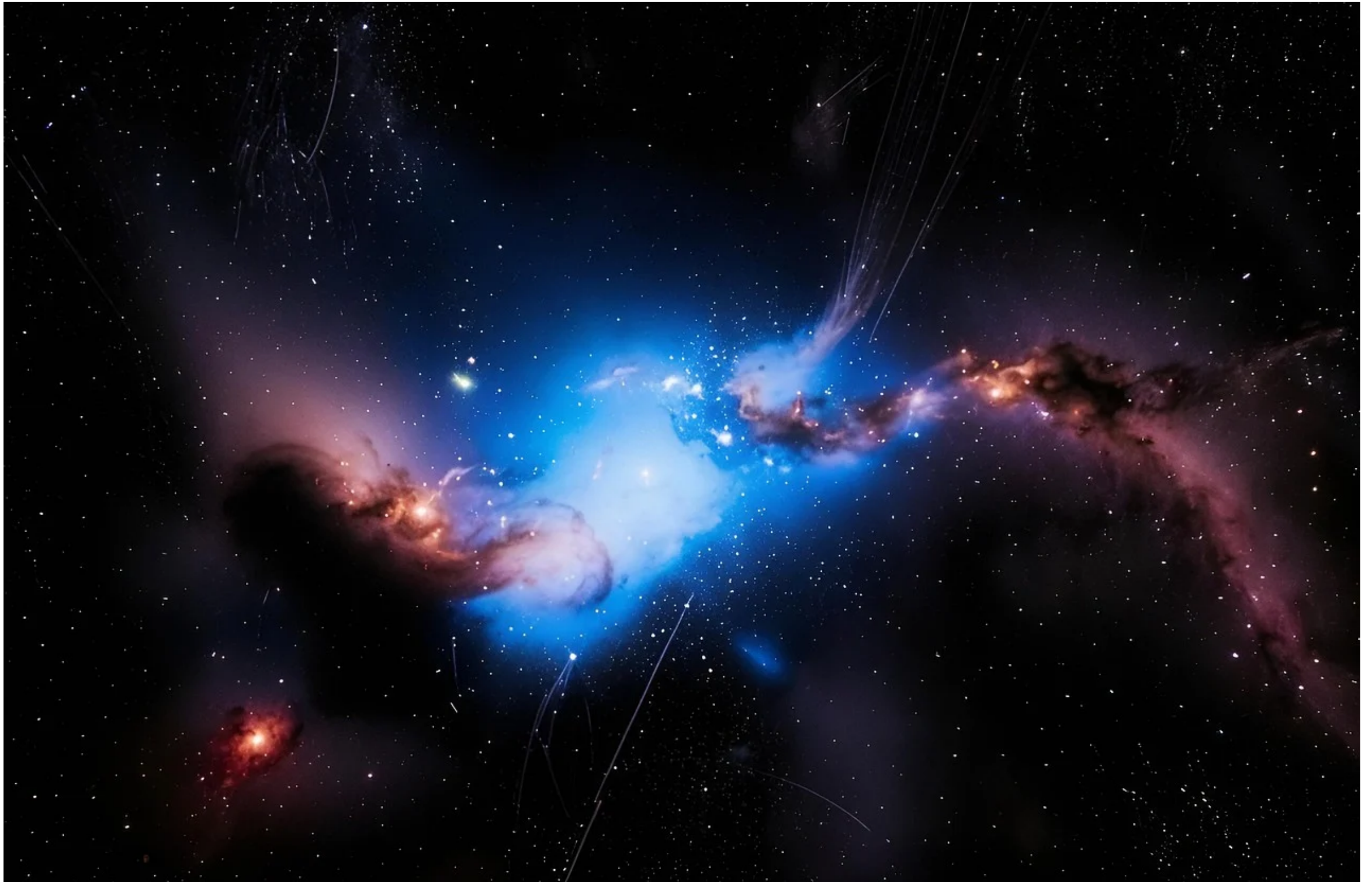
Conclusiones comparación

- **Ambas Importantes:**
Buenos algoritmos y buen hardware
- **Pero...**
Buenos algoritmos pesan más con datos más grandes.
- ¿Cuándo se necesita trabajar con muchos datos?

Redes sociales



Astronomía



Bioinformatics



Inteligencia artificial/aprendizaje de máquinas



Conclusiones comparación

- **Ambas Importantes:**
Buenos algoritmos y buen hardware
- **Pero...**
Buenos algoritmos pesan más con datos más grandes.
- ¿Cuándo se necesita trabajar con muchos datos?

Conclusiones comparación

- **Ambas Importantes:**
Buenos algoritmos y buen hardware
- **Pero...**
Buenos algoritmos pesan más con datos más grandes.
- ¿Cuándo se necesita trabajar con muchos datos?
 - ¡Muchas partes! Especialmente hoy en día.

Conclusiones comparación

- **Ambas Importantes:**
Buenos algoritmos y buen hardware
- **Pero...**
Buenos algoritmos pesan más con datos más grandes.
- ¿Cuándo se necesita trabajar con muchos datos?
 - ¡Muchas partes! Especialmente hoy en día.
- ¿Mejorar ordenar más allá de $n \log_2 n$?

Conclusiones comparación

- **Ambas Importantes:**
Buenos algoritmos y buen hardware
- **Pero...**
Buenos algoritmos pesan más con datos más grandes.
- ¿Cuándo se necesita trabajar con muchos datos?
 - ¡Muchas partes! Especialmente hoy en día.
- ¿Mejorar ordenar más allá de $n \log_2 n$?
 - Imposible. (¡HeapSort es el mejor algoritmo!)

Algunos avances algorítmicos

- Multiplicar números de n dígitos.

Algunos avances algorítmicos

- Multiplicar números de n dígitos.



$$n^2$$

Algunos avances algorítmicos

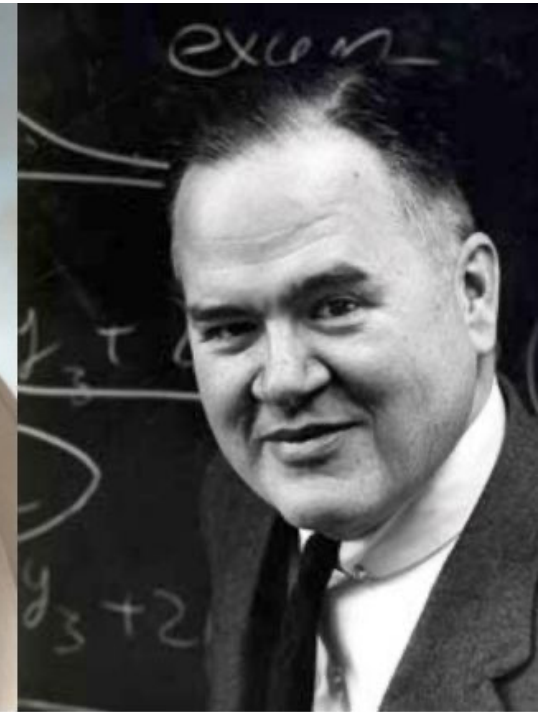
- Multiplicar números de n dígitos.



$$n^2$$



James William Cooley
(1926-)



John Wilder Tukey
(1915-2000)

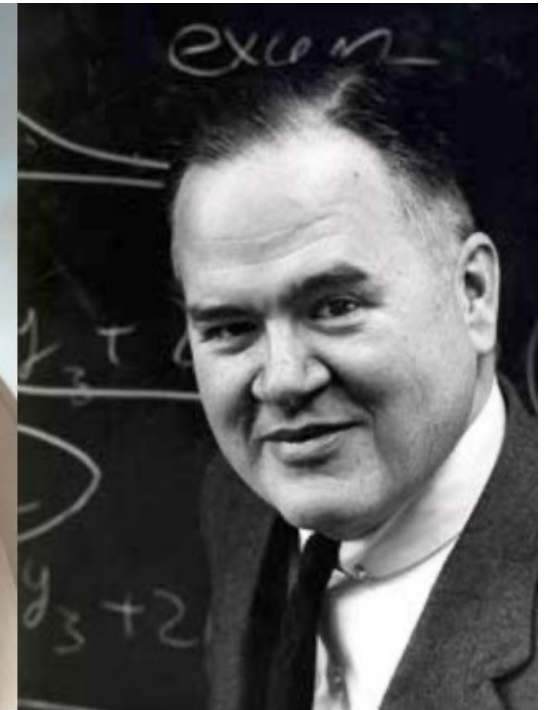
$$n \log_2 n \text{ (1965)}$$

Algunos avances algorítmicos

- Multiplicar números de n dígitos.



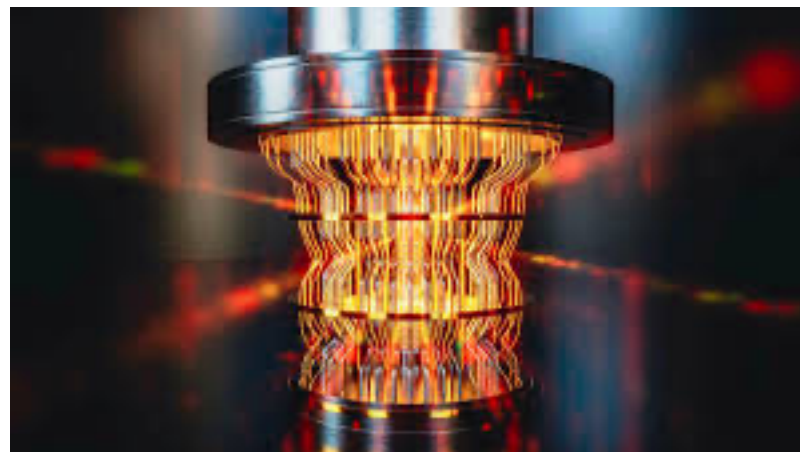
James William Cooley
(1926-)



John Wilder Tukey
(1915-2000)

$$n^2$$

$$n \log_2 n \text{ (1965)}$$



Algunos avances algorítmicos

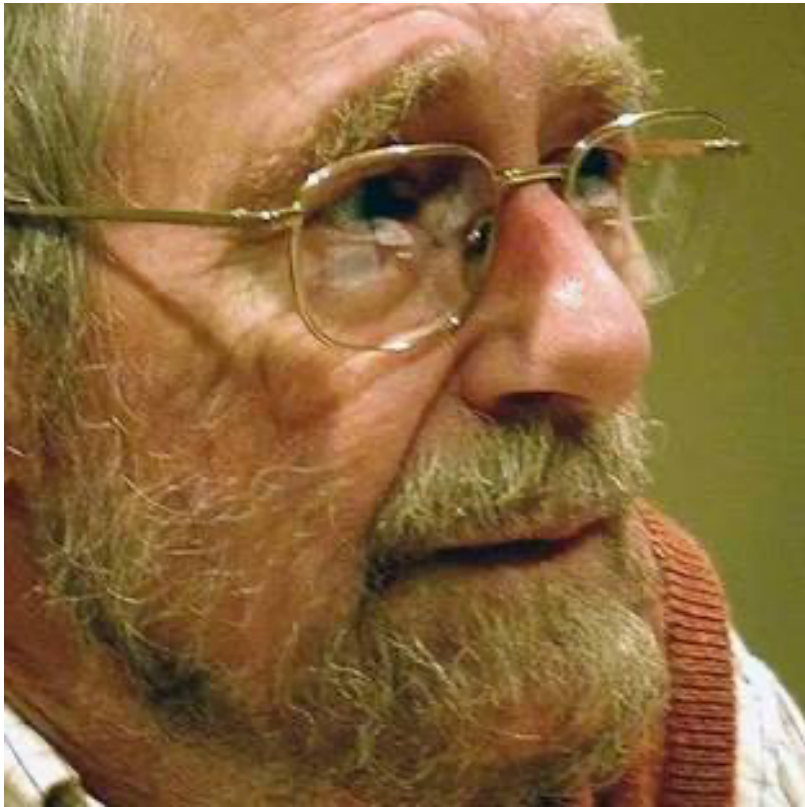
- Multiplicar números de n dígitos.
- Encontrar el camino más corto de un punto a otro.

?

Algunos avances algorítmicos

- Multiplicar números de n dígitos.
- Encontrar el camino más corto de un punto a otro.

?

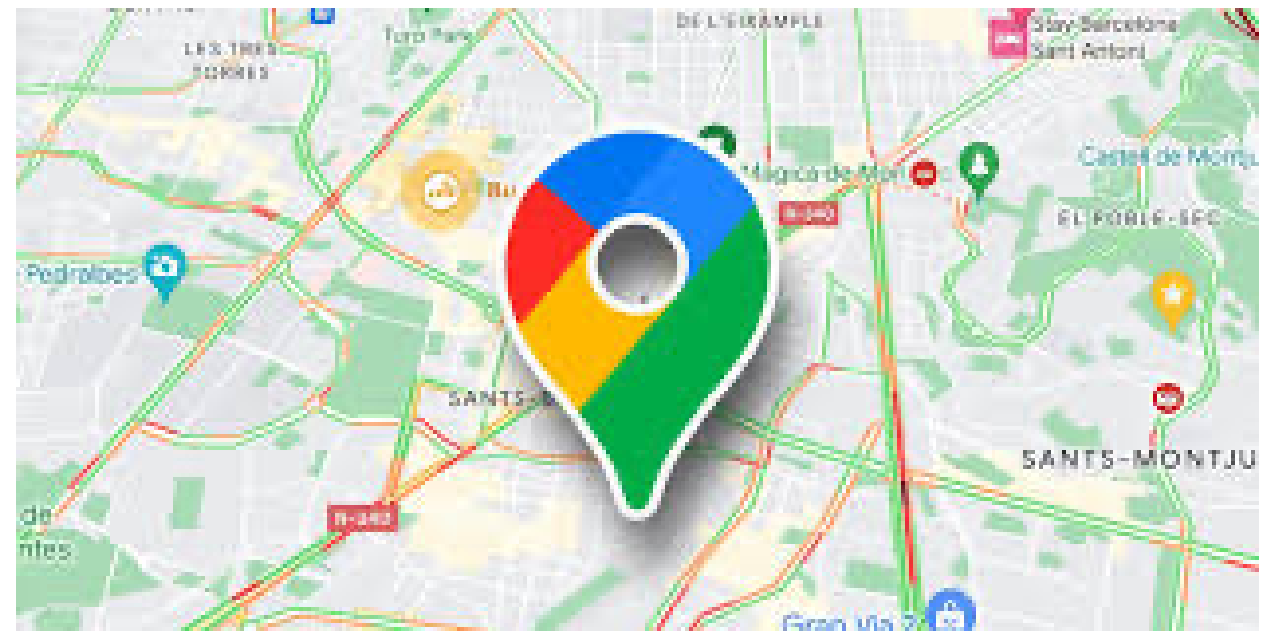
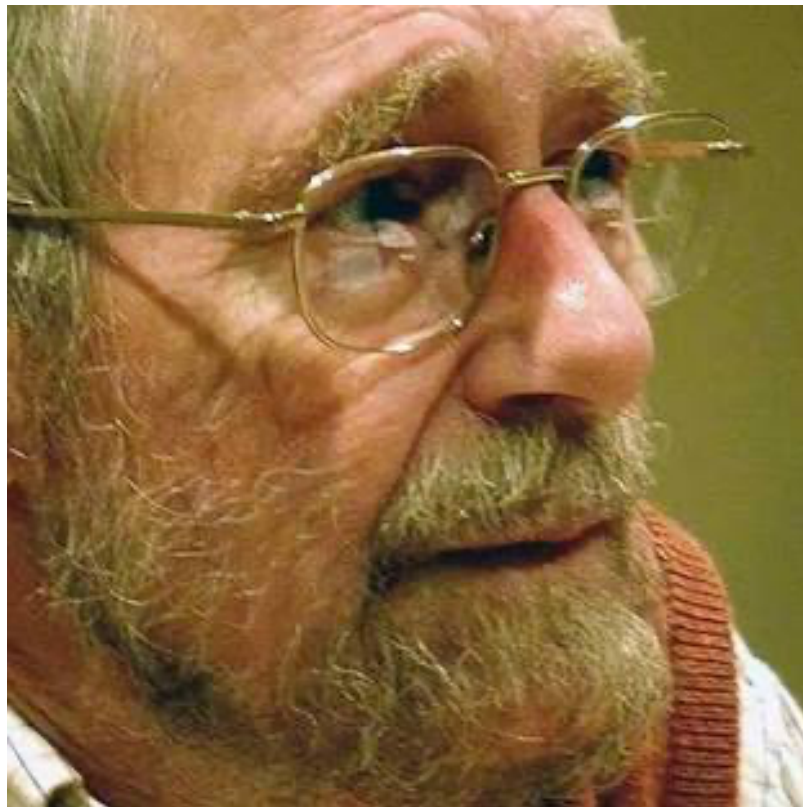


$n \log_2 n$ (Dijkstra 1959)

Algunos avances algorítmicos

- Multiplicar números de n dígitos.
- Encontrar el camino más corto de un punto a otro.

?



$n \log_2 n$ (Dijkstra 1959)

Algunos avances algorítmicos

- Multiplicar números de n dígitos.
- Encontrar el camino más corto de un punto a otro.
- Álgebra lineal



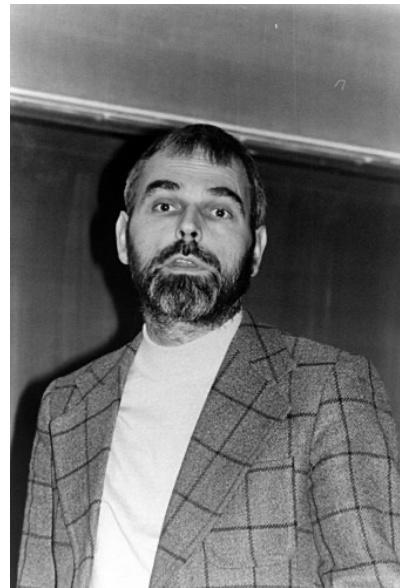
n^3 (Gauß app. 1800)

Algunos avances algorítmicos

- Multiplicar números de n dígitos.
- Encontrar el camino más corto de un punto a otro.
- Álgebra lineal



n^3 (Gauß app. 1800)



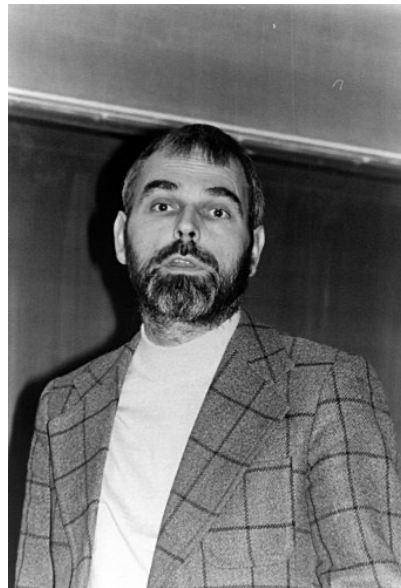
$n^{2.8}$ (Strassen 1969)

Algunos avances algorítmicos

- Multiplicar números de n dígitos.
- Encontrar el camino más corto de un punto a otro.
- Álgebra lineal



n^3 (Gauß app. 1800)



$n^{2.8}$ (Strassen 1969)



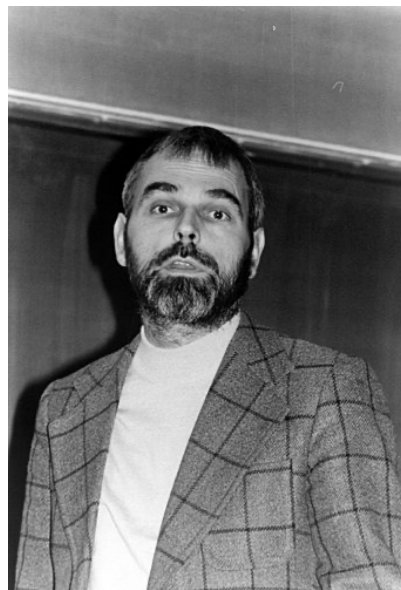
$n^{2.3}$ (Vassilevska 2012)

Algunos avances algorítmicos

- Multiplicar números de n dígitos.
- Encontrar el camino más corto de un punto a otro.
- Álgebra lineal



n^3 (Gauß app. 1800)



$n^{2.8}$ (Strassen 1969)



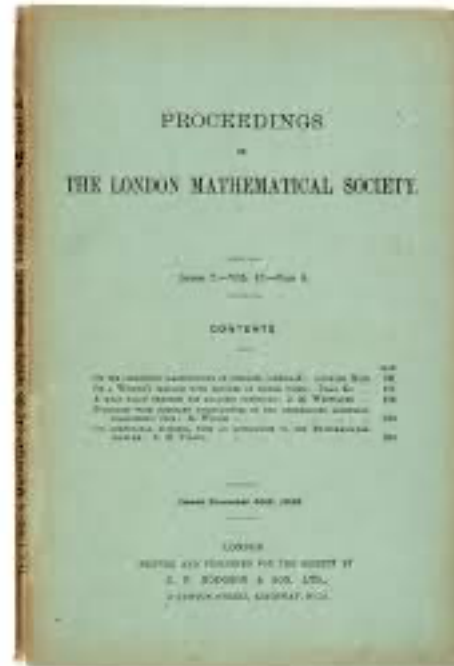
$n^{2.3}$ (Vassilevska 2012)



■ ¿Todos los problemas tienen algoritmos eficientes?

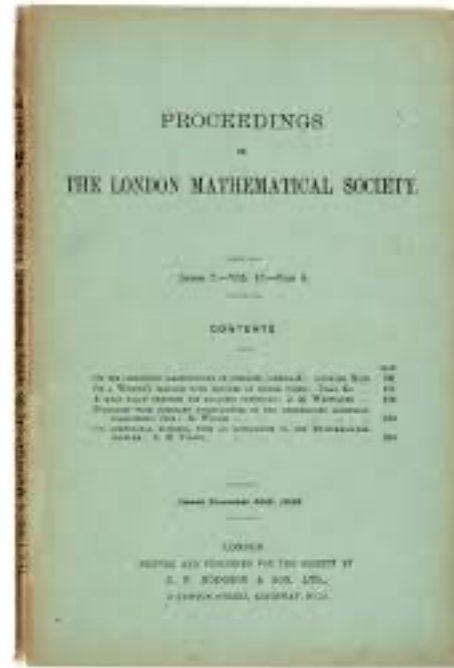
¿Todos los problemas tienen algoritmos eficientes?

- **No:**

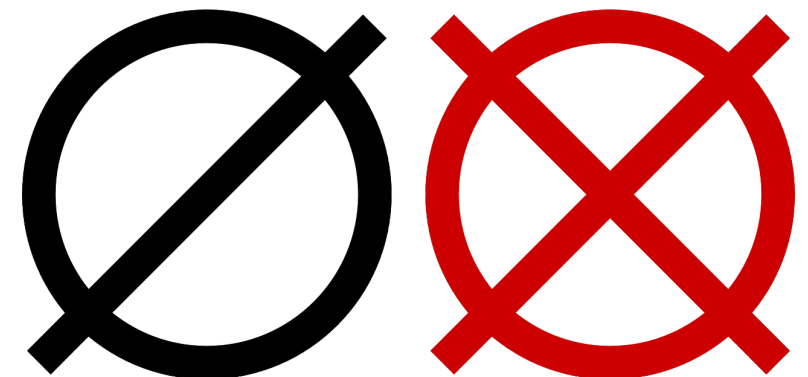


¿Todos los problemas tienen algoritmos eficientes?

- **No:**



- Imposible de crear un algoritmo que detecte que un algoritmo está "atascado"



■ ¿Todos los problemas tienen algoritmos eficientes?

- **Restrinjamos los problemas:**

¿Todos los problemas tienen algoritmos eficientes?

- **Restringamos los problemas:**
- Problemas donde podemos chequear la solución eficientemente.

¿Todos los problemas tienen algoritmos eficientes?

- **Restringamos los problemas:**
- Problemas donde podemos chequear la solución eficientemente.
- **Entrada:** Un puzzle de sudoku.

| | | | |
|---|--|---|---|
| 1 | | | |
| | | | 3 |
| 2 | | | |
| | | 4 | |

Los números del 1 al 4 deben aparecer en cada fila, columna y subcuadrícula.

¿Todos los problemas tienen algoritmos eficientes?

- **Restringamos los problemas:**
- Problemas donde podemos chequear la solución eficientemente.
- **Entrada:** Un puzzle de sudoku.

| | | | |
|---|---|---|---|
| 1 | 3 | 2 | 4 |
| 4 | 2 | 1 | 3 |
| 2 | 4 | 3 | 1 |
| 3 | 1 | 4 | 2 |

Los números del 1 al 4 deben aparecer en cada fila, columna y subcuadrícula.

- **Salida:** La solución.

Sudoku generalizado

- Sudokus de $n \times n$.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | | 9 | | | | 4 | | |
| 7 | | 8 | 3 | | 4 | 9 | | |
| 6 | | 1 | | | | 7 | 3 | |
| 4 | 6 | 2 | 5 | 3 | 9 | 8 | 7 | 1 |
| 3 | 8 | 5 | 7 | 2 | 1 | 6 | 4 | 9 |
| 1 | 9 | 7 | 4 | 6 | 8 | 2 | 5 | 3 |
| 2 | 5 | 6 | 1 | | | 3 | 9 | 4 |
| 9 | 1 | 3 | | 4 | | 5 | 8 | 7 |
| 8 | 7 | 4 | | 5 | 3 | 1 | 2 | 6 |

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 6 | | | | | 8 | 11 | | | 15 | 14 | | | 16 | |
| 15 | 11 | | | | 16 | 14 | | | 12 | | | 6 | | | |
| 13 | | 9 | 12 | | | | | 3 | 16 | 14 | | 15 | 11 | 10 | |
| 2 | 16 | | 11 | | 15 | 10 | 1 | | | | | | | | |
| | 15 | 11 | 10 | | | 16 | 2 | 13 | 8 | 9 | 12 | | | | |
| 12 | 13 | | | 4 | 1 | 5 | 6 | 2 | 3 | | | | 11 | 10 | |
| 5 | | 6 | 1 | 12 | | 9 | | 15 | 11 | 10 | 7 | 16 | | 3 | |
| | 2 | | | 10 | | 11 | 6 | | 5 | | | 13 | | 9 | |
| 10 | 7 | 15 | 11 | 16 | | | | 12 | 13 | | | | | 6 | |
| 9 | | | | | | 1 | | 2 | | 16 | 10 | | | 11 | |
| 1 | | 4 | 6 | 9 | 13 | | | 7 | | 11 | | 3 | 16 | | |
| 16 | 14 | | | 7 | | 10 | 15 | 4 | 6 | 1 | | | | 13 | 8 |
| 11 | 10 | | 15 | | | | 16 | 9 | 12 | 13 | | | 1 | 5 | 4 |
| | | 12 | | 1 | 4 | 6 | | 16 | | | | 11 | 10 | | |
| | | 5 | | 8 | 12 | 13 | | 10 | | | 11 | 2 | | | 14 |
| 3 | 16 | | | 10 | | | 7 | | | 6 | | | | | 12 |

Sudoku generalizado

- Sudokus de $n \times n$.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | | 9 | | | | 4 | | |
| 7 | | 8 | 3 | | 4 | 9 | | |
| 6 | | 1 | | | | 7 | 3 | |
| 4 | 6 | 2 | 5 | 3 | 9 | 8 | 7 | 1 |
| 3 | 8 | 5 | 7 | 2 | 1 | 6 | 4 | 9 |
| 1 | 9 | 7 | 4 | 6 | 8 | 2 | 5 | 3 |
| 2 | 5 | 6 | 1 | | | 3 | 9 | 4 |
| 9 | 1 | 3 | | 4 | | 5 | 8 | 7 |
| 8 | 7 | 4 | | 5 | 3 | 1 | 2 | 6 |

| | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|----|
| | 6 | | | | | 8 | 11 | | | 15 | 14 | | | 16 | | |
| 15 | 11 | | | | 16 | 14 | | | 12 | | | 6 | | | | |
| 13 | | 9 | 12 | | | | | 3 | 16 | 14 | | 15 | 11 | 10 | | |
| 2 | 16 | | 11 | | 15 | 10 | 1 | | | | | | | | | |
| | 15 | 11 | 10 | | | 16 | 2 | 13 | 8 | 9 | 12 | | | | | |
| 12 | 13 | | | 4 | 1 | 5 | 6 | 2 | 3 | | | | 11 | 10 | | |
| 5 | | 6 | 1 | 12 | | 9 | | 15 | 11 | 10 | 7 | 16 | | 3 | | |
| | 2 | | | 10 | | 11 | 6 | | 5 | | | 13 | | 9 | | |
| 10 | 7 | 15 | 11 | 16 | | | | 12 | 13 | | | | | 6 | | |
| 9 | | | | | | 1 | | 2 | | 16 | 10 | | | 11 | | |
| 1 | | 4 | 6 | 9 | 13 | | | 7 | | 11 | | 3 | 16 | | | |
| 16 | 14 | | | 7 | | 10 | 15 | 4 | 6 | 1 | | | | 13 | 8 | |
| 11 | 10 | | 15 | | | | | 16 | 9 | 12 | 13 | | | 1 | 5 | 4 |
| | | 12 | | 1 | 4 | 6 | | 16 | | | | 11 | 10 | | | |
| | | 5 | | 8 | 12 | 13 | | 10 | | | 11 | 2 | | | | 14 |
| 3 | 16 | | | 10 | | | 7 | | | 6 | | | | | | 12 |

- Podemos chequear si la solución está correcta en $3n^2$ operaciones.

Sudoku generalizado

- Sudokus de $n \times n$.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | | 9 | | | | 4 | | |
| 7 | | 8 | 3 | | 4 | 9 | | |
| 6 | | 1 | | | | 7 | 3 | |
| 4 | 6 | 2 | 5 | 3 | 9 | 8 | 7 | 1 |
| 3 | 8 | 5 | 7 | 2 | 1 | 6 | 4 | 9 |
| 1 | 9 | 7 | 4 | 6 | 8 | 2 | 5 | 3 |
| 2 | 5 | 6 | 1 | | | 3 | 9 | 4 |
| 9 | 1 | 3 | | 4 | | 5 | 8 | 7 |
| 8 | 7 | 4 | | 5 | 3 | 1 | 2 | 6 |

| | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|----|
| | 6 | | | | | 8 | 11 | | | 15 | 14 | | | 16 | | |
| 15 | 11 | | | | 16 | 14 | | | 12 | | | 6 | | | | |
| 13 | | 9 | 12 | | | | | 3 | 16 | 14 | | 15 | 11 | 10 | | |
| 2 | 16 | | 11 | | 15 | 10 | 1 | | | | | | | | | |
| | 15 | 11 | 10 | | | 16 | 2 | 13 | 8 | 9 | 12 | | | | | |
| 12 | 13 | | | 4 | 1 | 5 | 6 | 2 | 3 | | | | 11 | 10 | | |
| 5 | | 6 | 1 | 12 | | 9 | | 15 | 11 | 10 | 7 | 16 | | 3 | | |
| | 2 | | | 10 | | 11 | 6 | | 5 | | | 13 | | 9 | | |
| 10 | 7 | 15 | 11 | 16 | | | | 12 | 13 | | | | | 6 | | |
| 9 | | | | | | 1 | | 2 | | 16 | 10 | | | 11 | | |
| 1 | | 4 | 6 | 9 | 13 | | | 7 | | 11 | | 3 | 16 | | | |
| 16 | 14 | | | 7 | | 10 | 15 | 4 | 6 | 1 | | | | 13 | 8 | |
| 11 | 10 | | 15 | | | | | 16 | 9 | 12 | 13 | | | 1 | 5 | 4 |
| | | 12 | | 1 | 4 | 6 | | 16 | | | | 11 | 10 | | | |
| | | 5 | | 8 | 12 | 13 | | 10 | | | 11 | 2 | | | | 14 |
| 3 | 16 | | | 10 | | | 7 | | | 6 | | | | | | 12 |

- Podemos chequear si la solución está correcta en $3n^2$ operaciones.

Resolviendo Sudoku

- ¿Podemos resolver sudoku en $10n^2$?

Resolviendo Sudoku

- ¿Podemos resolver sudoku en $10n^2$?
 - ¿ $10 \cdot n^{10}$?

Resolviendo Sudoku

- ¿Podemos resolver sudoku en $10n^2$?
 - ¿ $10 \cdot n^{10}$?
 - ¿ $10 \cdot n^{200}$?

Resolviendo Sudoku

- ¿Podemos resolver sudoku en $10n^2$?
 - ¿ $10 \cdot n^{10}$?
 - ¿ $10 \cdot n^{200}$?
 - ¿ $10 \cdot n^{99999999999999}$?

Resolviendo Sudoku

- ¿Podemos resolver sudoku en $10n^2$?
 - ¿ $10 \cdot n^{10}$?
 - ¿ $10 \cdot n^{200}$?
 - ¿ $10 \cdot n^{999999999999}$?
- Premio de un millón de dolares por responder la pregunta anterior.



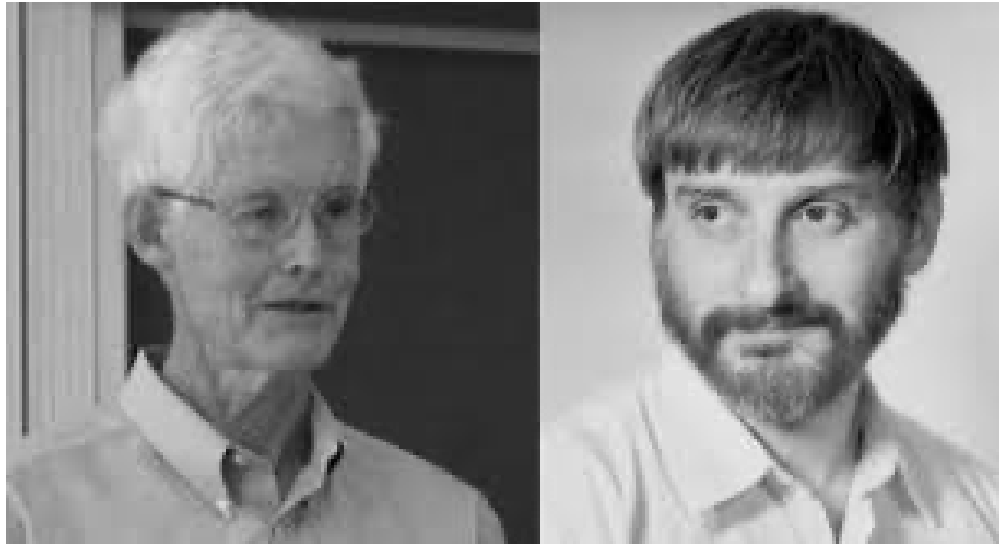
¿Por qué?

- Sudoku es un problema especial (NP-completo).

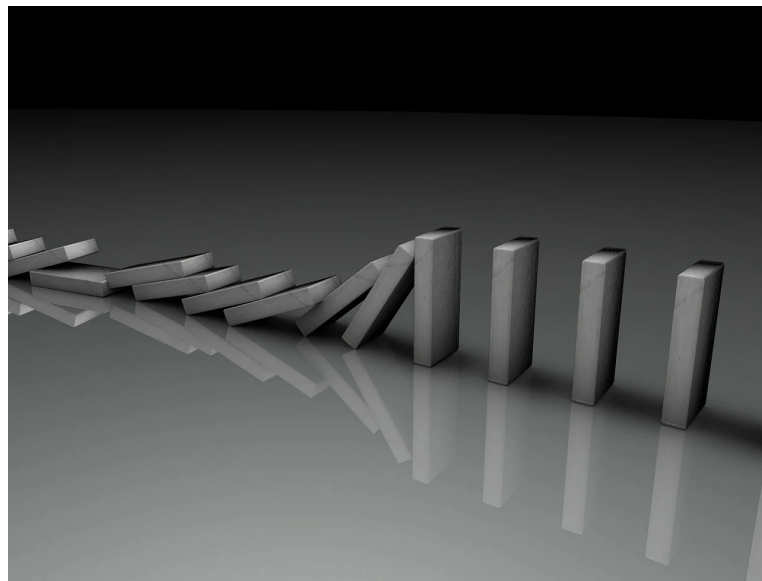


¿Por qué?

- Sudoku es un problema especial (NP-completo).



- Todos esos problemas están interrelacionados.

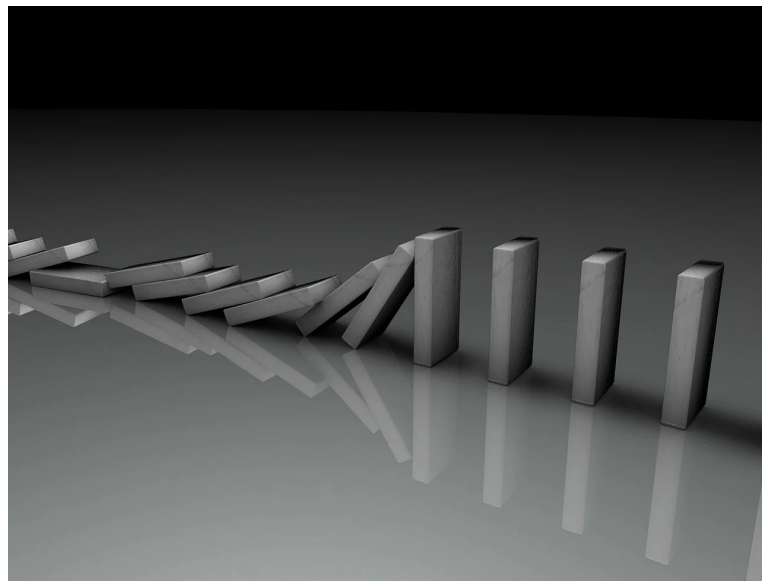


¿Por qué?

- Sudoku es un problema especial (NP-completo).



- Todos esos problemas están interrelacionados.



- Un algoritmo eficiente para cualquiera, resuelve el resto.

¿Que problemas son NP-completos?

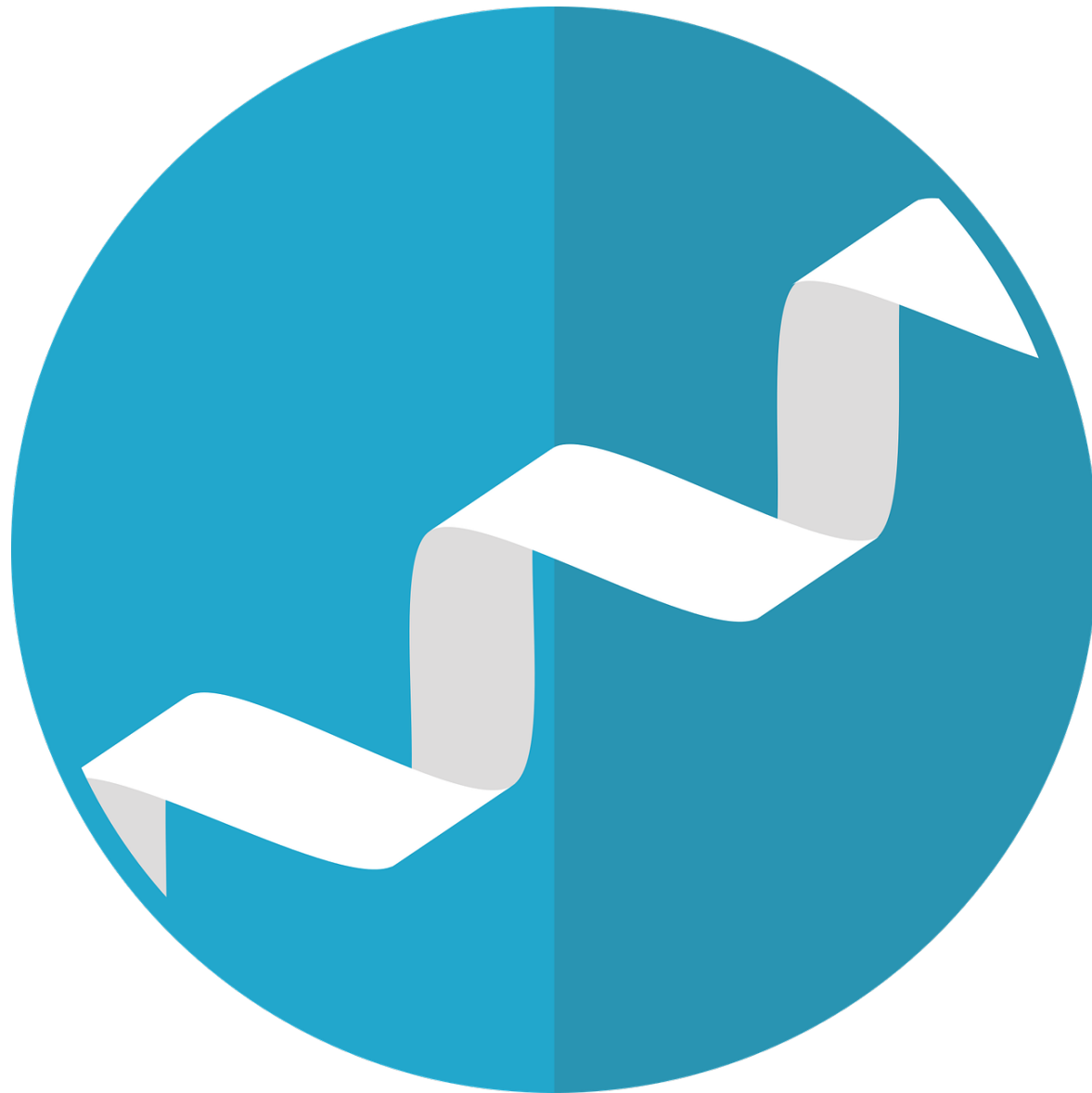
- Sudoku.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | | 9 | | | | 4 | | |
| 7 | | 8 | 3 | | 4 | 9 | | |
| 6 | | 1 | | | | 7 | 3 | |
| 4 | 6 | 2 | 5 | 3 | 9 | 8 | 7 | 1 |
| 3 | 8 | 5 | 7 | 2 | 1 | 6 | 4 | 9 |
| 1 | 9 | 7 | 4 | 6 | 8 | 2 | 5 | 3 |
| 2 | 5 | 6 | 1 | | | 3 | 9 | 4 |
| 9 | 1 | 3 | | 4 | | 5 | 8 | 7 |
| 8 | 7 | 4 | | 5 | 3 | 1 | 2 | 6 |

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | 6 | | | | 8 | 11 | | 15 | 14 | | 16 | | | | |
| 15 | 11 | | | 16 | 14 | | | 12 | | 6 | | | | | |
| 13 | | 9 | 12 | | | | 3 | 16 | 14 | 15 | 11 | 10 | | | |
| 2 | 16 | | 11 | | 15 | 10 | 1 | | | | | | | | |
| | 15 | 11 | 10 | | | 16 | 2 | 13 | 8 | 9 | 12 | | | | |
| 12 | 13 | | | 4 | 1 | 5 | 6 | 2 | 3 | | | | 11 | 10 | |
| 5 | | 6 | 1 | 12 | | 9 | | 15 | 11 | 10 | 7 | 16 | | 3 | |
| | 2 | | | 10 | | 11 | 6 | | 5 | | | 13 | | 9 | |
| 10 | 7 | 15 | 11 | 16 | | | | 12 | 13 | | | | | 6 | |
| 9 | | | | | | 1 | | 2 | | 16 | 10 | | | 11 | |
| 1 | | 4 | 6 | 9 | 13 | | | 7 | | 11 | | 3 | 16 | | |
| 16 | 14 | | | 7 | | 10 | 15 | 4 | 6 | 1 | | | | 13 | 8 |
| 11 | 10 | | 15 | | | | 16 | 9 | 12 | 13 | | | 1 | 5 | 4 |
| | | 12 | | 1 | 4 | 6 | 16 | | | | | 11 | 10 | | |
| | | 5 | | 8 | 12 | 13 | | 10 | | | | 11 | 2 | | 14 |
| 3 | 16 | | | 10 | | | 7 | | | 6 | | | | | 12 |

¿Que problemas son NP-completos?

- Sudoku.
- Doblado de proteínas.



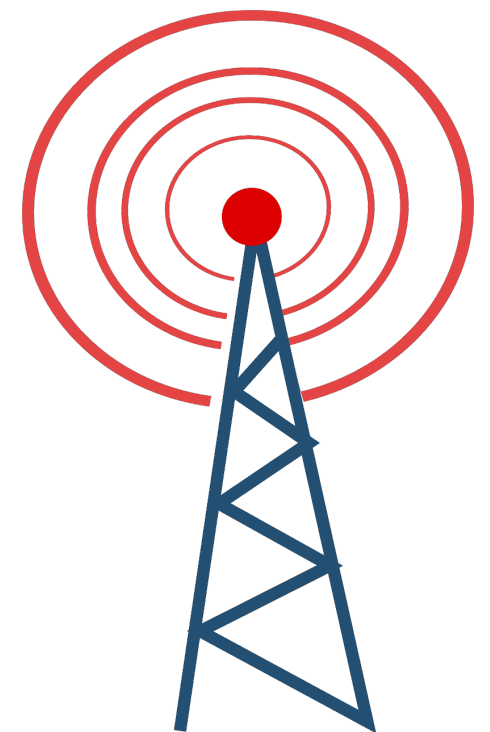
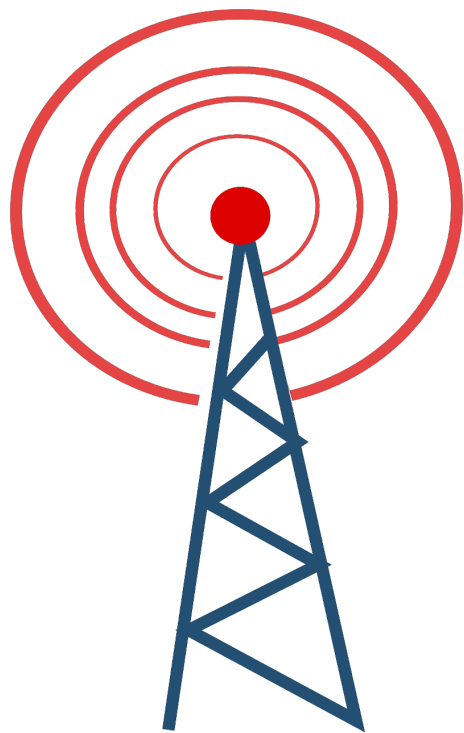
¿Que problemas son NP-completos?

- Sudoku.
- Doblado de proteínas.
- Problemas de ruteo de vehículos.



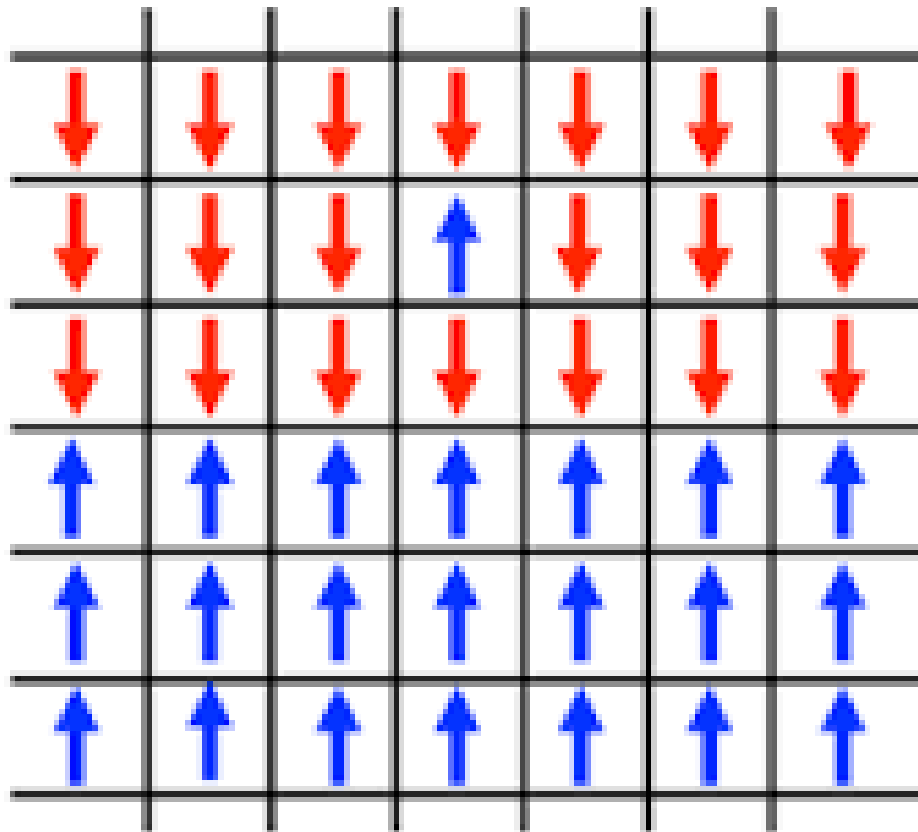
¿Que problemas son NP-completos?

- Sudoku.
- Doblado de proteínas.
- Problemas de ruteo de vehículos.
- Problemas en telecomunicaciones.



¿Que problemas son NP-completos?

- Sudoku.
- Doblado de proteínas.
- Problemas de ruteo de vehículos.
- Problemas en telecomunicaciones.
- Modelos termodinámicos (Ising)



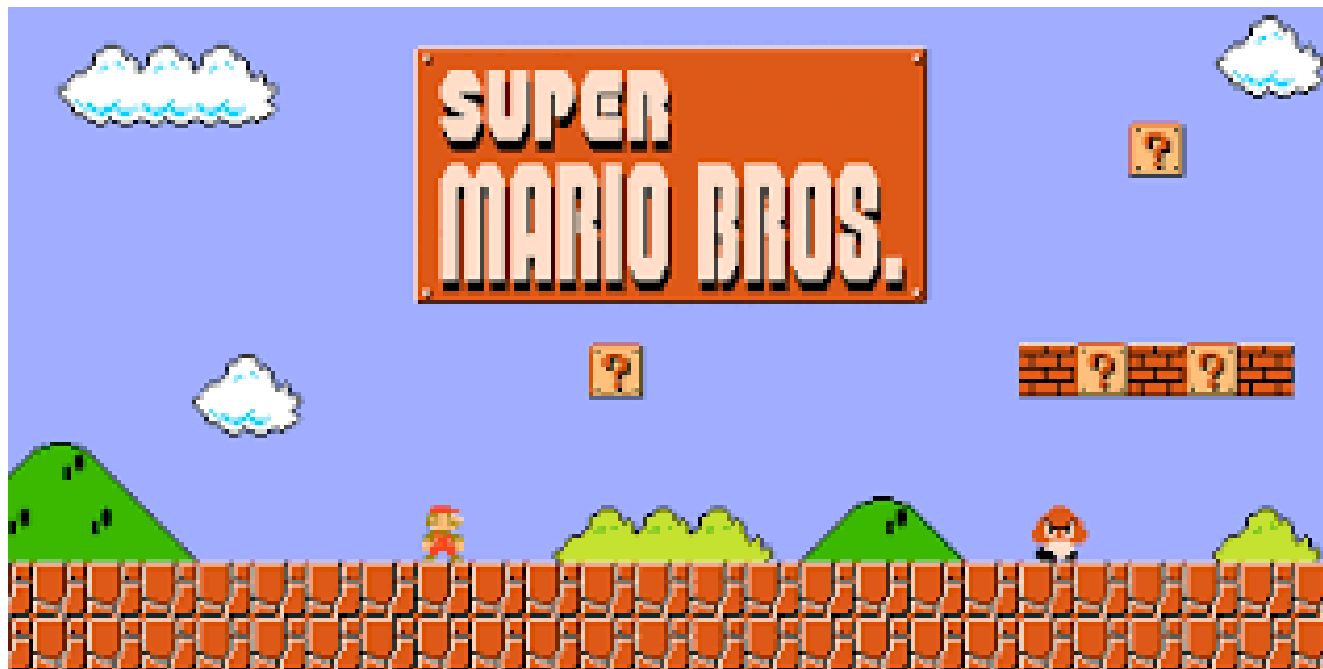
¿Que problemas son NP-completos?

- Sudoku.
- Doblado de proteínas.
- Problemas de ruteo de vehículos.
- Problemas en telecomunicaciones.
- Modelos termodinámicos (Ising)
- Asignación de pacientes a camas de hospital.



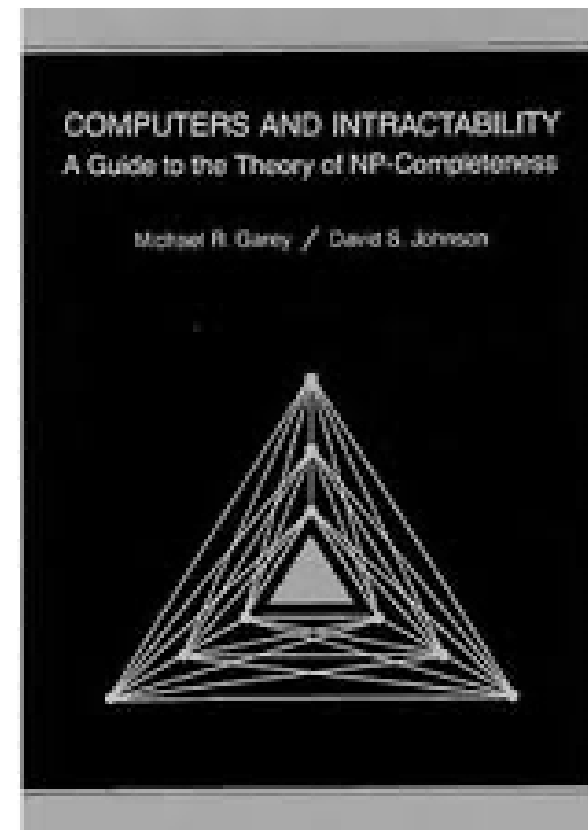
¿Que problemas son NP-completos?

- Sudoku.
- Doblado de proteínas.
- Problemas de ruteo de vehículos.
- Problemas en telecomunicaciones.
- Modelos termodinámicos (Ising)
- Asignación de pacientes a camas de hospital.
- Jugar bien Super Mario.



¿Que problemas son NP-completos?

- Sudoku.
- Doblado de proteínas.
- Problemas de ruteo de vehículos.
- Problemas en telecomunicaciones.
- Modelos termodinámicos (Ising)
- Asignación de pacientes a camas de hospital.
- Jugar bien Super Mario.
- Muchos, muchos, más.



■ ¿Que opinan los científicos de la computación?

- Problema muy, muy, difícil.

¿Que opinan los científicos de la computación?

- Problema muy, muy, difícil.
- La mayoría opina que **no hay** algoritmos eficientes para problemas NP-completos.



P=NP?

¿Que opinan los científicos de la computación?

- Problema muy, muy, difícil.
- La mayoría opina que **no hay** algoritmos eficientes para problemas NP-completos.



P=NP?

- Contras y pros.

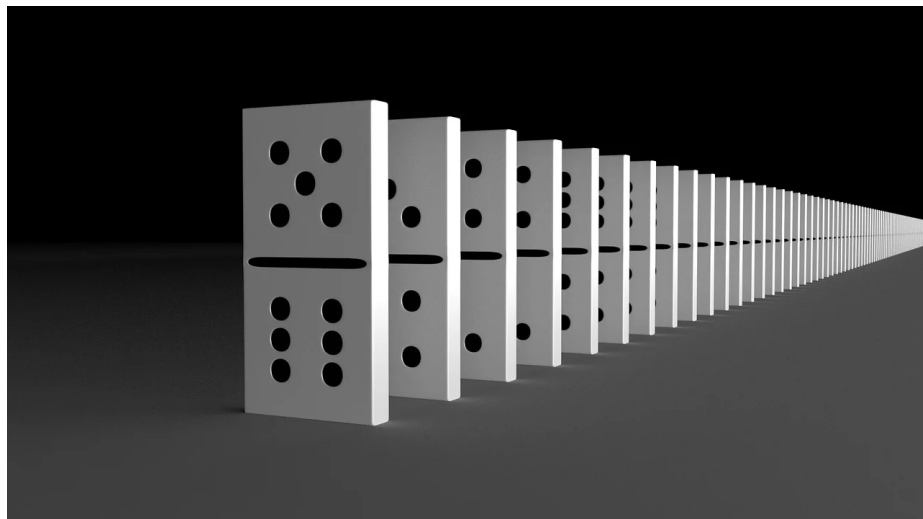
¿Que opinan los científicos de la computación?

- Problema muy, muy, difícil.
- La mayoría opina que **no hay** algoritmos eficientes para problemas NP-completos.



P=NP?

- Contras y pros.



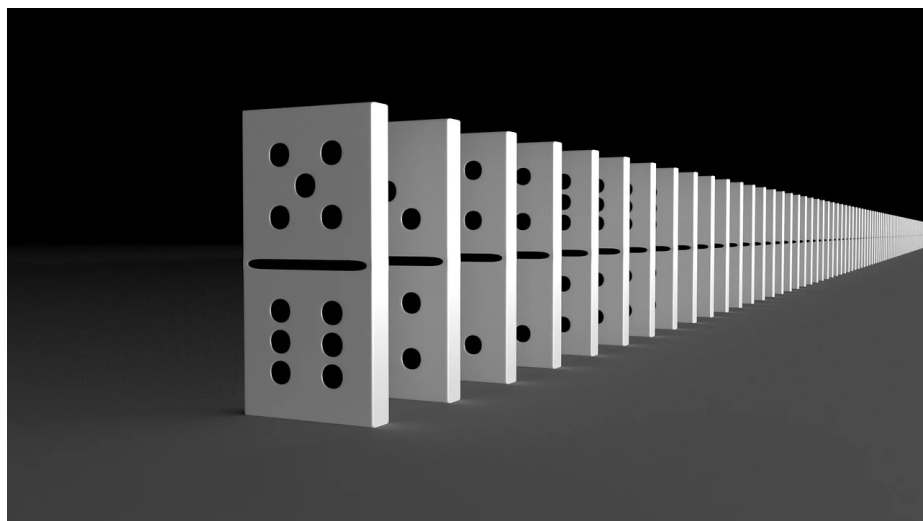
¿Que opinan los científicos de la computación?

- Problema muy, muy, difícil.
- La mayoría opina que **no hay** algoritmos eficientes para problemas NP-completos.



P=NP?

- Contras y pros.



■ Conceptos clave

- **Algoritmo:** Instrucciones para resolver un problema computacional.

Conceptos clave

- **Algoritmo:** Instrucciones para resolver un problema computacional.
- **Eficiencia** de algoritmos: Número de operaciones y su efecto en el tiempo.

Conceptos clave

- **Algoritmo:** Instrucciones para resolver un problema computacional.
- **Eficiencia** de algoritmos: Número de operaciones y su efecto en el tiempo.
- Clave diseñar algoritmos eficientes, sobre todo en la gran cantidad de datos.

Conceptos clave

- **Algoritmo:** Instrucciones para resolver un problema computacional.
- **Eficiencia** de algoritmos: Número de operaciones y su efecto en el tiempo.
- Clave diseñar algoritmos eficientes, sobre todo en la gran cantidad de datos.
- No todos los problemas tienen algoritmos o algoritmos eficientes.

Conceptos clave

- **Algoritmo:** Instrucciones para resolver un problema computacional.
- **Eficiencia** de algoritmos: Número de operaciones y su efecto en el tiempo.
- Clave diseñar algoritmos eficientes, sobre todo en la gran cantidad de datos.
- No todos los problemas tienen algoritmos o algoritmos eficientes.
- **NP-completo:** Problemas entrelazados.

Conceptos clave

- **Algoritmo:** Instrucciones para resolver un problema computacional.
- **Eficiencia** de algoritmos: Número de operaciones y su efecto en el tiempo.
- Clave diseñar algoritmos eficientes, sobre todo en la gran cantidad de datos.
- No todos los problemas tienen algoritmos o algoritmos eficientes.
- **NP-completo:** Problemas entrelazados.

¿Preguntas?