

# De las Ciencias de la Computación a la Bioinformática en Pocas Escalas!

Gabriel Tolosa

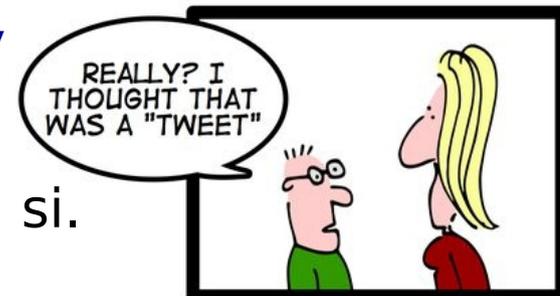
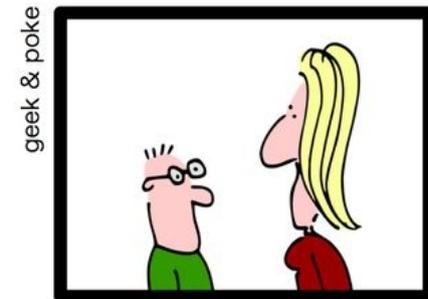
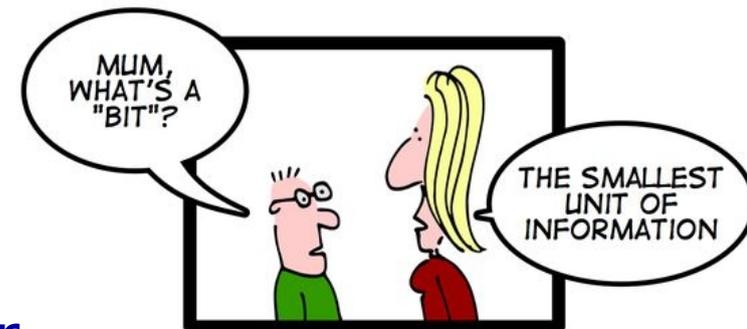


# Ciencias de la Computación???

- Es el “Estudio de las Computadoras”?
  - No realmente...
  - No se trata de las herramientas sino de cómo las usamos y “descubrimos” qué pueden hacer...
- Es el “Estudio de cómo escribir programas”?
  - La programación es una parte importante, pero no lo es todo...
- Es el “Estudio de los usos y aplicaciones de computadoras y software”?
  - Nop! Pero diseñar y construir software, si.

*GEEK & POKE'S ENCYCLOPEDIA OF  
COMPUTER SCIENCE*

*PART 1: THE "BIT"*



# Entonces?

- Modelos formales de computación
- Diseño de algoritmos
- Programación/Lenguajes
- Diseño y análisis de software
- ...
- ...
- ...

Cómo **resolver problemas** usando computadoras

:-)



**KEEP  
CALM**  
AND LOVE

**Computer  
Science**

# Qué ciencia?

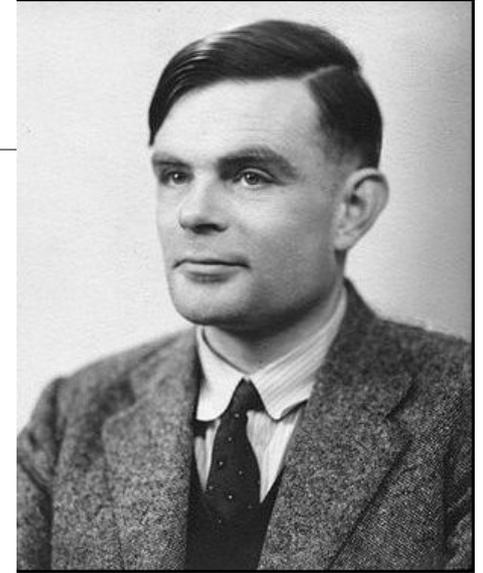
- Comprender a la naturaleza a través de la observación
- En general, sobre “cosas” reales como partículas, océanos, seres vivos, cometas, etc.
- La matemática y las ciencias de la computación “operan” sobre “cosas” abstractas como números, grafos, funciones, listas, etc.

- Que luego se mapean al mundo real!
- Programación es considerado un “Arte”

```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodename()
    label=symbol.sym_name.get(int(ast[0]), ast[0])
    print '  %s [label="%s' % (nodename, label),
    if isinstance(ast[1], str):
        if ast[1].strip():
            print '= %s';' % ast[1]
        else:
            print ''
    else:
        print ''
        children = []
        for n, childenumerate(ast[1:]):
            children.append(dotwrite(child))
        print ', ' % ast -> (' % nodename
        for n, namechildren
            print '%s' % name,
```

# De dónde vienen las CC?



## Raíces en dos campos:

### Matemática

- **Alan Turing y la máquina de Turing (1930s)**
  - Desarrolló teorías (en papel!) sobre cómo desarrollar “computo”.
- **George Boole (1850s)**
  - Fundamentos de la aritmética computacional moderna.

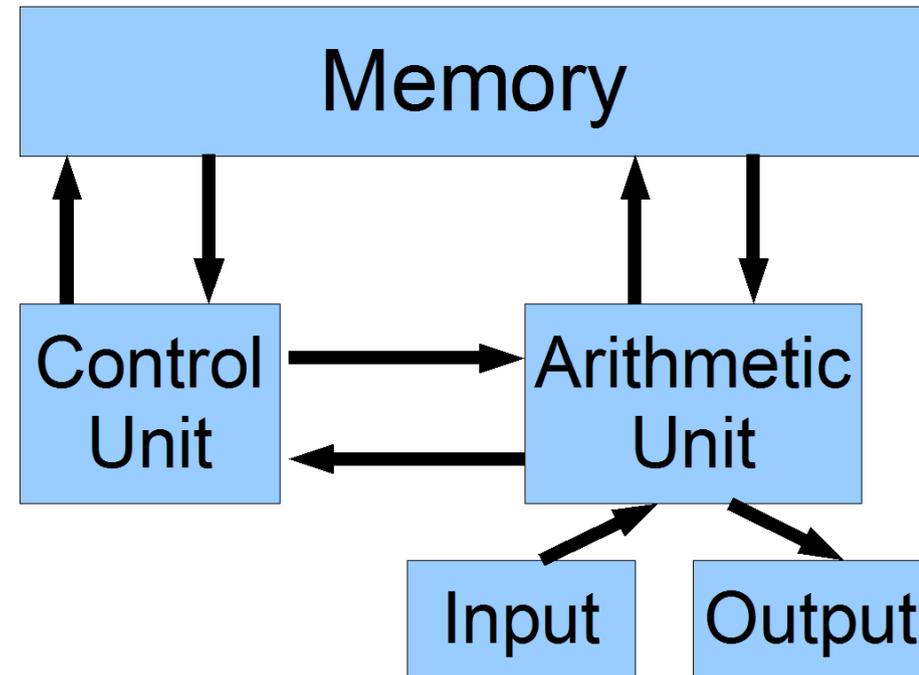
"A computer would deserve to be called intelligent if it could deceive a human into believing that it was human."

# De dónde vienen las CC?

## Raíces en dos campos:

### Ingeniería

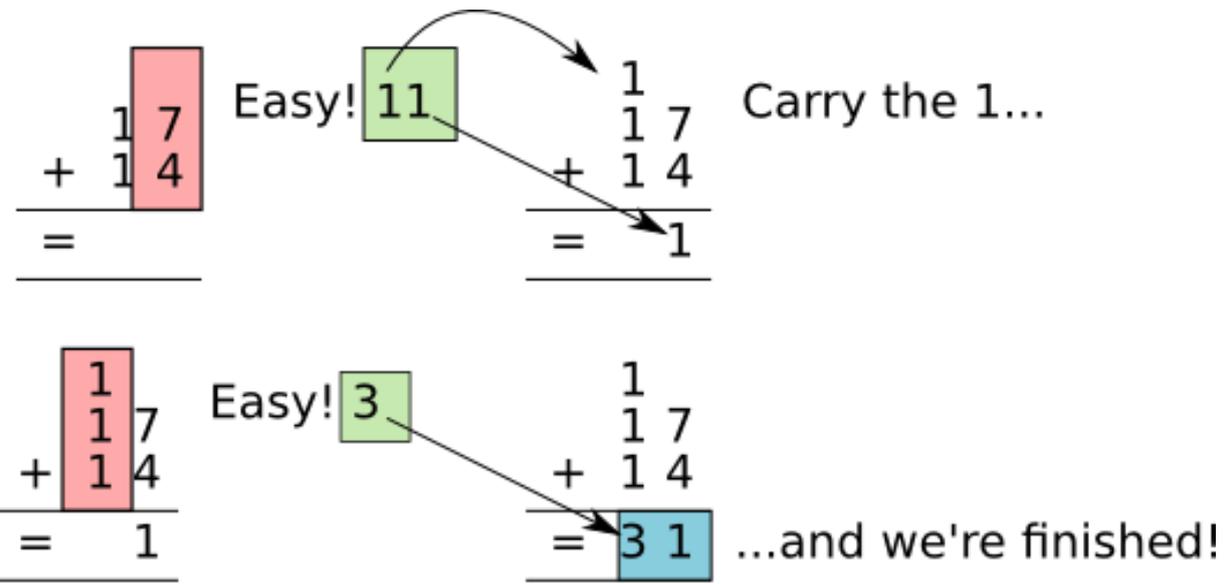
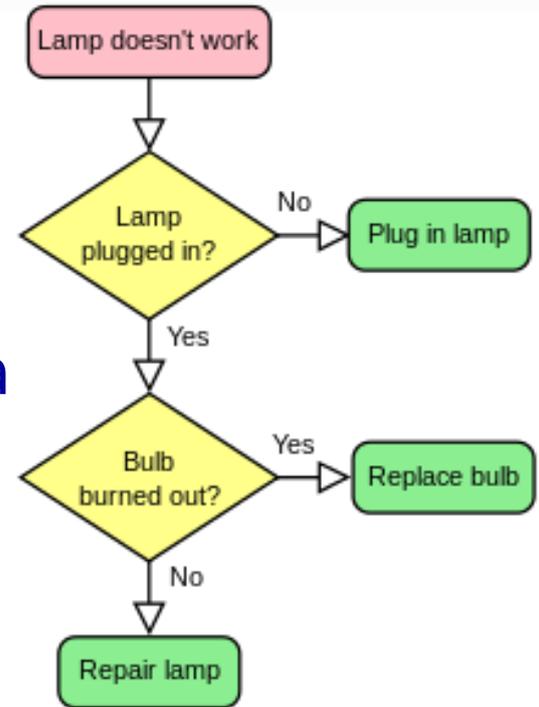
- **John von Neumann y la máquina de von Neumann (1940s)**
  - Mostró cómo construir computadoras físicas con circuitos electrónicos.



**There are only 10 types of people in the world:  
Those who understand binary and those who don't.**

# Algoritmos

- Secuencia clara de pasos para llevar a cabo una tarea
- Ordenada
- Bien definida
- Operaciones posibles
- Lleva un tiempo finito
- Termina



# Ejemplo

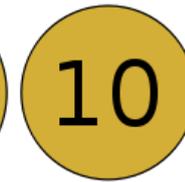
- El problema del “**cambio**”
  - *Convertir una suma de dinero  $W$  en cambio usando la menor cantidad de monedas posibles.*

$$\sum_{j=1}^n w_j x_j = W.$$

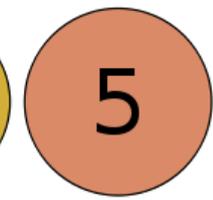
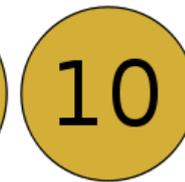
$$36-20=16$$



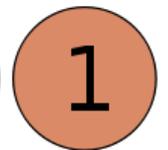
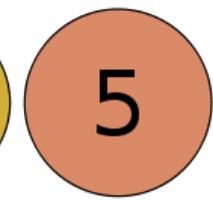
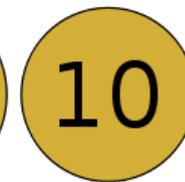
$$16-10=6$$



$$6-5=1$$



$$1-1=0$$



# Ejemplo

- El problema del “**cambio**”
  - *Ojo! Con algunas aproximaciones*
  - *Considerar el ejemplo*
  - **Monedas = {0.30, 0.20, 0.05, 0.01}**
    - *Cómo es el cambio de 0.40?*
  - *Con la estrategia del “siempre el mas grande”*
    - $0.30 + 0.05 + 0.05 = 0.40$  **(3 monedas)**
  - *Pero..*
    - $0.20 + 0.20 = 0.40$  **(2 monedas)**

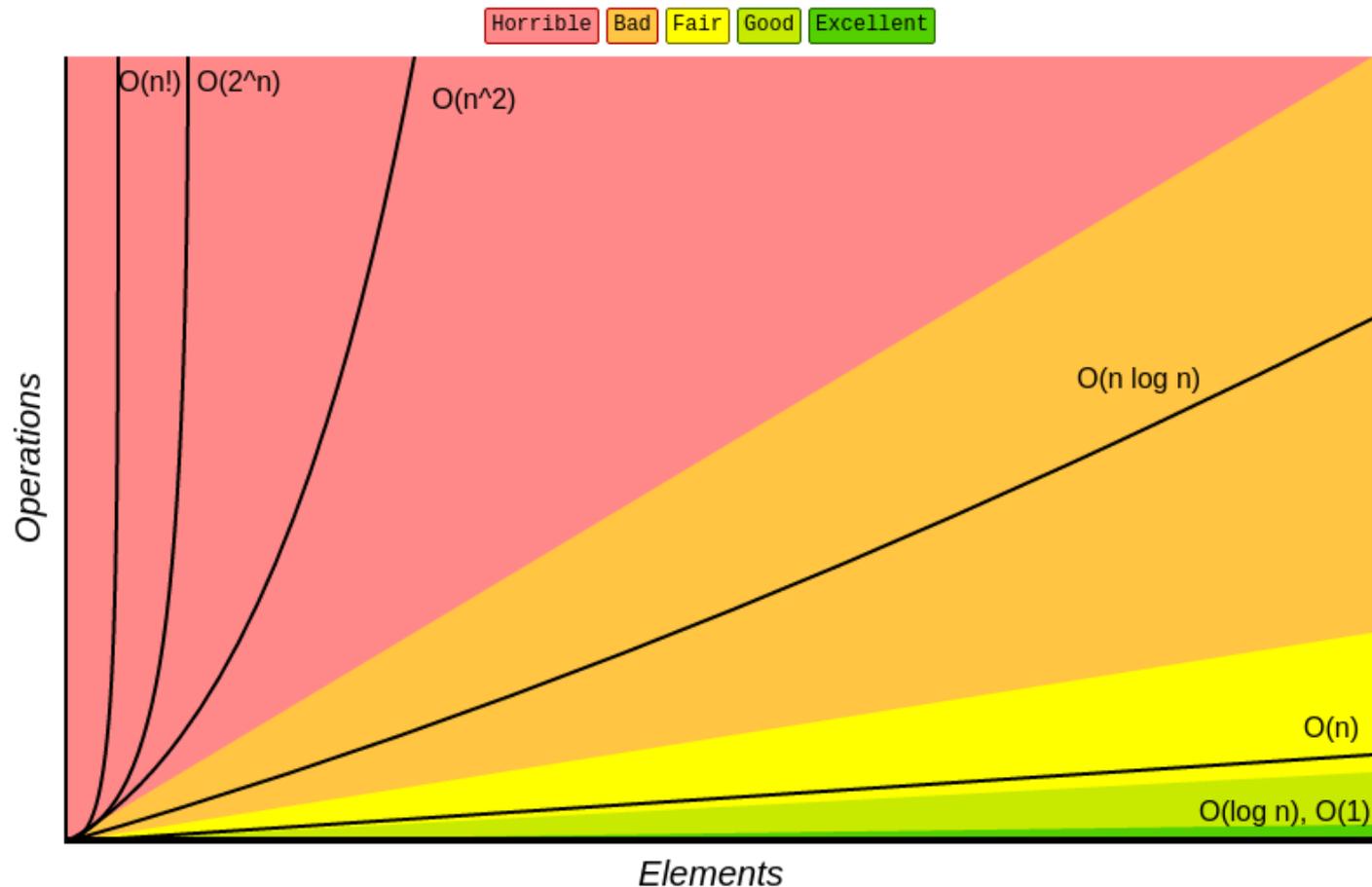
# Qué nos interesa?

- El primer (y más difícil) paso para resolver un problema computacional es identificarlo precisamente (y generalizarlo)
  - Identificar la “mejor” estrategia para resolverlo
  - Dos preguntas importantes:
    - Funciona correctamente?
    - Qué recursos requiere?
      - Tiempo
      - Espacio

**Teoría de la complejidad!**

# Qué nos interesa?

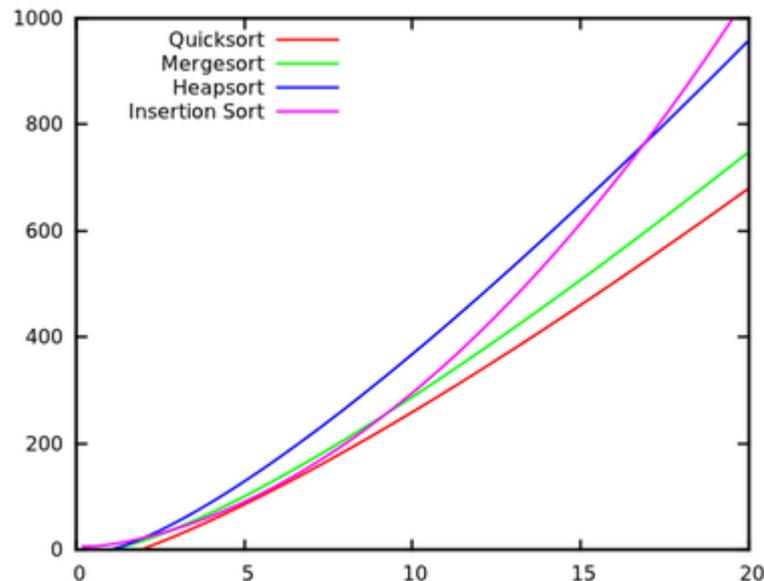
- Cómo crece el uso de recursos de un algoritmo en función del tamaño de la entrada ( $n$ )



# Dos problemas principales

- **Sorting**

- Dada una lista de elementos (y un procedimiento de comparación) retornarla en orden.
- Traten de pensar (y sistematizar) cómo ordenan a mano las cartas de truco (4 palos, 10 cartas por palo).
- Traten de pensar cuánto cuesta el procedimiento.



# Dos problemas principales

- **Searching**

- Casi todos los problemas se pueden reducir a uno de búsqueda.
- El problema aparece cuando el espacio de búsqueda es suficientemente grande!!!
- Estrategias
  - Fuerza bruta (secuencial)
  - Binaria
  - Indexada



**Cuánto  
cuesta  
c/u?**

# Por qué es importante?

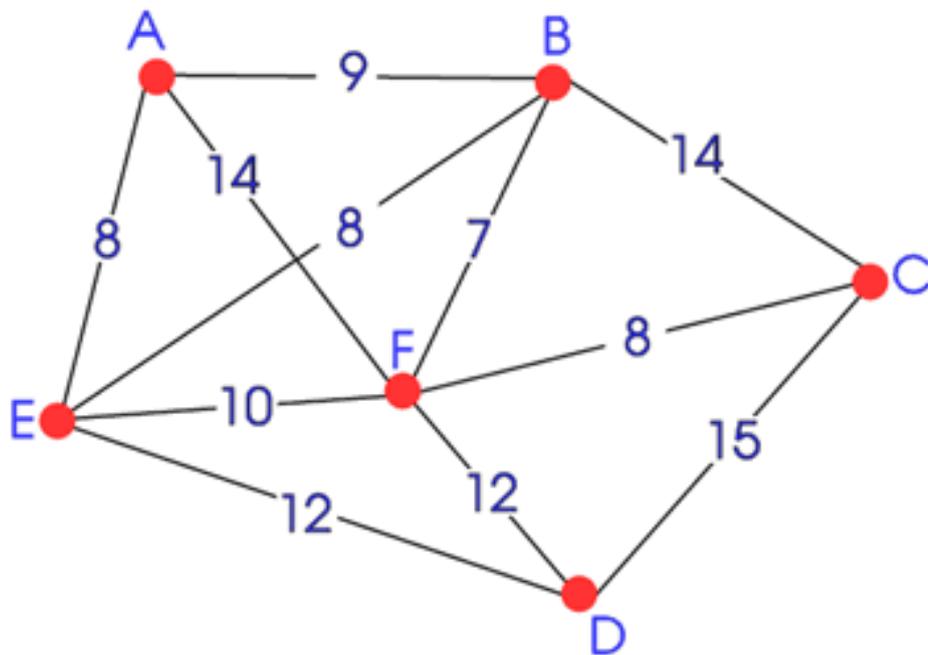
- **Ejemplo:**

- Dado el siguiente conjunto de números **N**, verificar si algún subconjunto suma **x**
- $N = \{1, 3, 12, 29, 34, 45, 77, 81\}$
- $x = 89$
- Un algoritmo *dummy* arma todos los subconjuntos y compara.
- Fácil para  $|N| = 8$ , pero...
- Suponiendo  $|N| = 100$  y cada evaluación toma 0,00001 s
- Cantidad de subconjuntos =  $2^{100}$
- Luego:
  - $T = 2^{100} \times 0,00001 \text{ s} = 1,46 \times 10^{17} \text{ siglos!}$

# Un clásico...

- **El problema del viajante de comercio**

- Dada una lista de ciudades y las distancias entre cada par de ellas, ¿cuál es la ruta más corta posible que visita cada ciudad exactamente una vez y regresa a la ciudad origen?

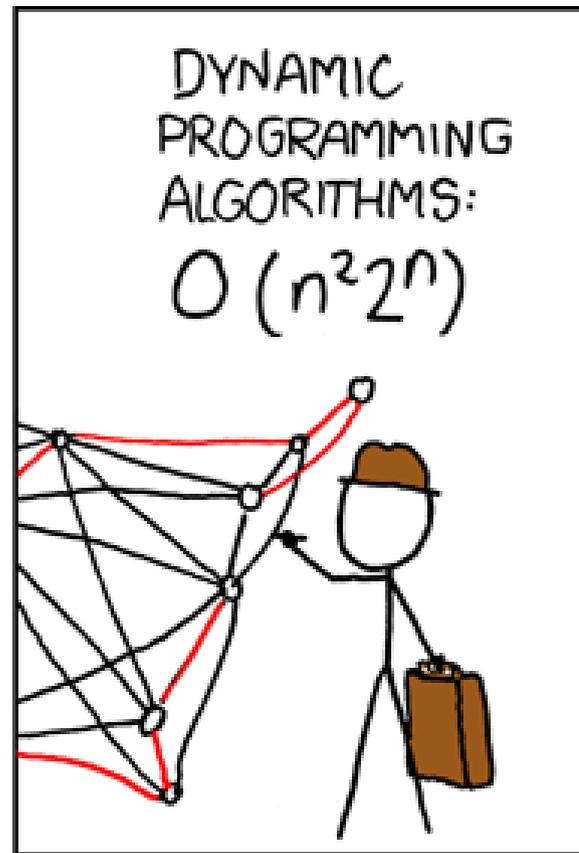
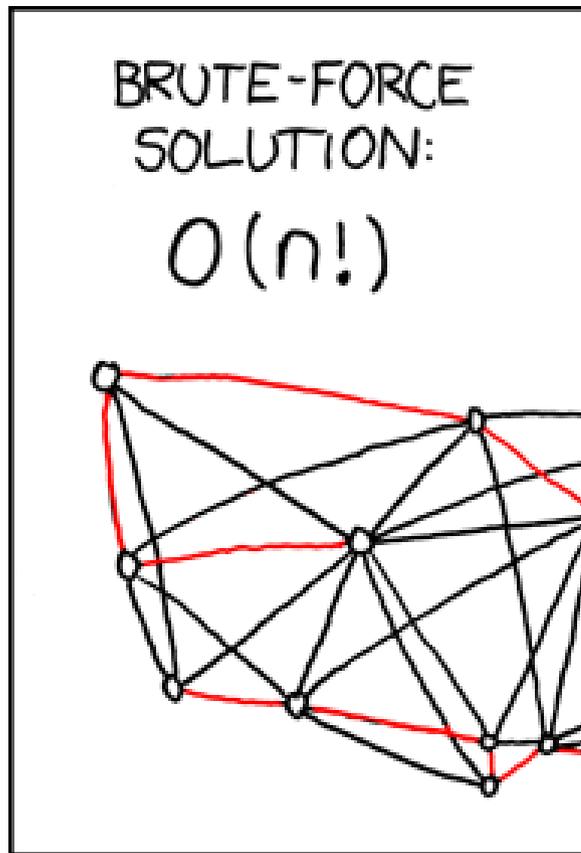


- Algoritmo exacto  $O(n!)$
- Con 6 ciudades  $\rightarrow$  720 combinaciones
- Para las 23 provincias argentinas  $\rightarrow$

25,852,016,738,885,000,000,000

# Un clásico...

- **La solución óptima!**

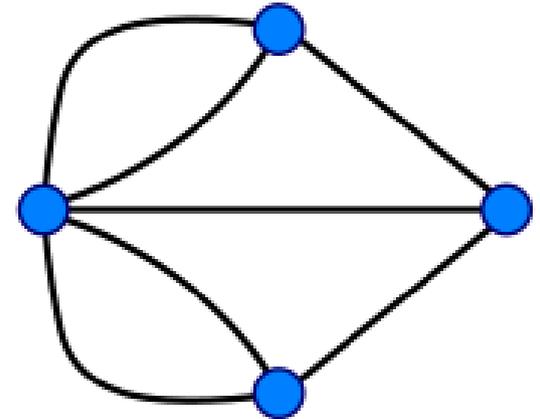
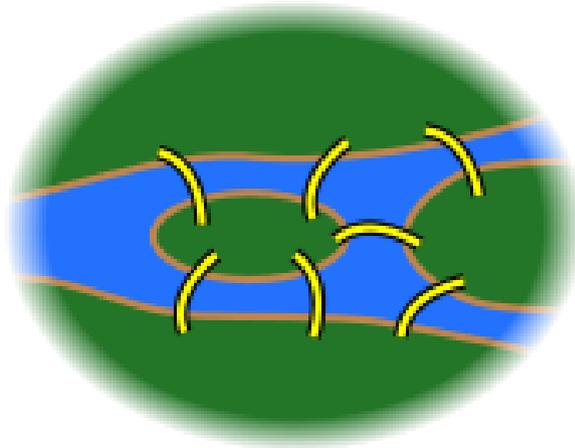
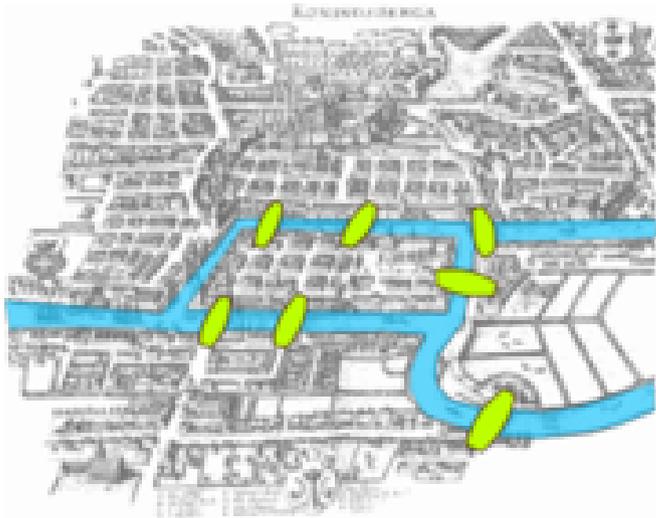


- Para las 23 provincias argentinas → 4.437.573.632

## Otro clásico...

- **Los puentes de Königsberg (Euler, 1736)**

¿Es posible dar un paseo comenzando desde cualquier región, pasando por todos los puentes sólo una, y regresando al punto de partida?



# Un problema más “moderno”

[Todos](#) [Imágenes](#) [Noticias](#) [Videos](#) [Maps](#) [Más ▾](#) [Herramientas de búsqueda](#)

Cerca de 1,760,000 resultados (0.34 segundos)

Quizás quisiste decir: **decisión**

## Decisión o Desición - Como se escribe

[www.como-se-escribe.com/decision-o-desicion/](http://www.como-se-escribe.com/decision-o-desicion/) ▾

Aprende cuál es la palabra correcta entre Decisión o Desición y su significado | [www.como-se-escribe.com](http://www.como-se-escribe.com).

## Cómo se escribe decisión o desición - unComo

[educacion.uncomo.com](http://educacion.uncomo.com) > [Educación](#) > [Formación](#) > [Aprende a escribir y a leer](#) ▾

Cómo se escribe decisión o desición. Escribir bien sin cometer errores ortográficos es algo fundamental a la hora de presentar textos o redacciones importantes ...

## Cómo se escribe: Decisión o Desición - ¡Solución FÁCIL y definición!

<https://www.spoots.com> > [Educación](#) > [Aprendizaje y estudio](#) > [Cómo se escribe](#) ▾

Hoy veremos cómo se escribe: decisión o desición, 2 palabras hómfonas (o sea, que se escriben diferente pero suenan igual) que son muy parecidas y que ...

## Decisión o Desición - Diccionario de Dudas

[www.diccionariodedudas.com/decision-o-desicion/](http://www.diccionariodedudas.com/decision-o-desicion/) ▾

Decisión es la forma correcta de escribir esta palabra. La grafía desición es una incorrección que consiste en confundir la posición de las letras c y s en...

**Cómo lo resolverían?**

# Veamos...

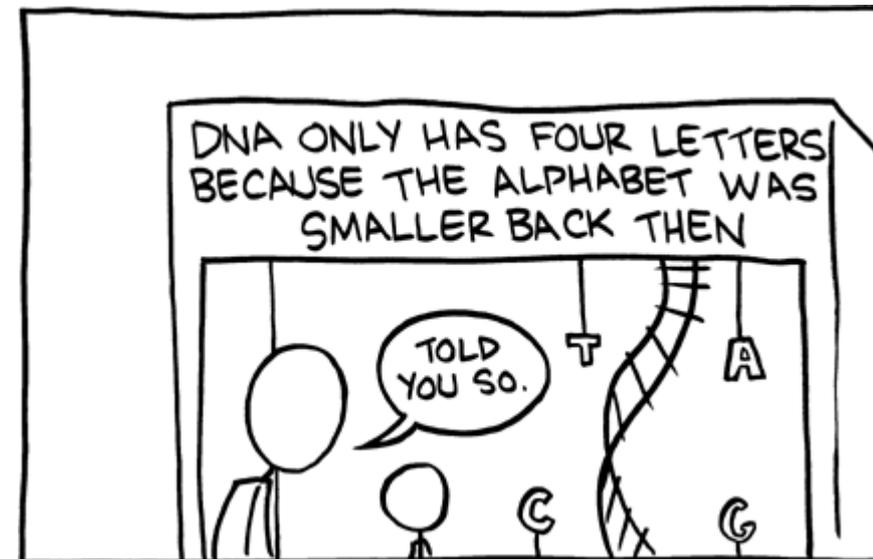
- **Este problema se llama “Distancia de Levenshtein”**  
Dadas 2 cadenas, retornar la distancia entre ellas, medida como el número mínimo de operaciones de edición requeridas para convertir una cadena en la otra.

		P	A	R	K
	delete ↓				
S					
P					
A			insert →		
K					substitute ↘
E					

		P	A	R	K
	0	1	2	3	4
S	1	1	2	3	4
P	2	1	2	3	4
A	3	2	1	2	3
K	4	3	2	2	2
E	5	4	3	3	3

# Y la bioinformática???

- **En las 2 primeras charlas Federico y Sabrina presentaron (entre otras cosas) el ADN...**
  - Hablaron de nucleóticos, exones, etc...
  - Hablaron de secuenciar y de alinear...
  - Hablaron del TAMAÑO del genoma (por ej, humano)
  - Dijeron que la informática viene al “rescate” para ayudar a “encontrar” patrones que pueden ser de interés...
- **Y?**

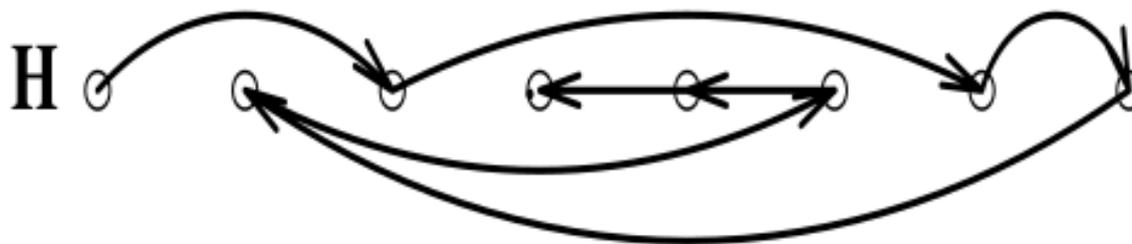


# Y la bioinformática???

- **Grafos**

## Sequence reconstruction (Hamiltonian path approach)

$S = \{ \text{ATG} \quad \text{AGG} \quad \text{TGC} \quad \text{TCC} \quad \text{GTC} \quad \text{GGT} \quad \text{GCA} \quad \text{CAG} \}$



*Vertices: l-tuples from the spectrum S. Edges: overlapping l-tuples.*

*Path visiting ALL VERTICES corresponds to sequence reconstruction*      ATGCAGGTCC

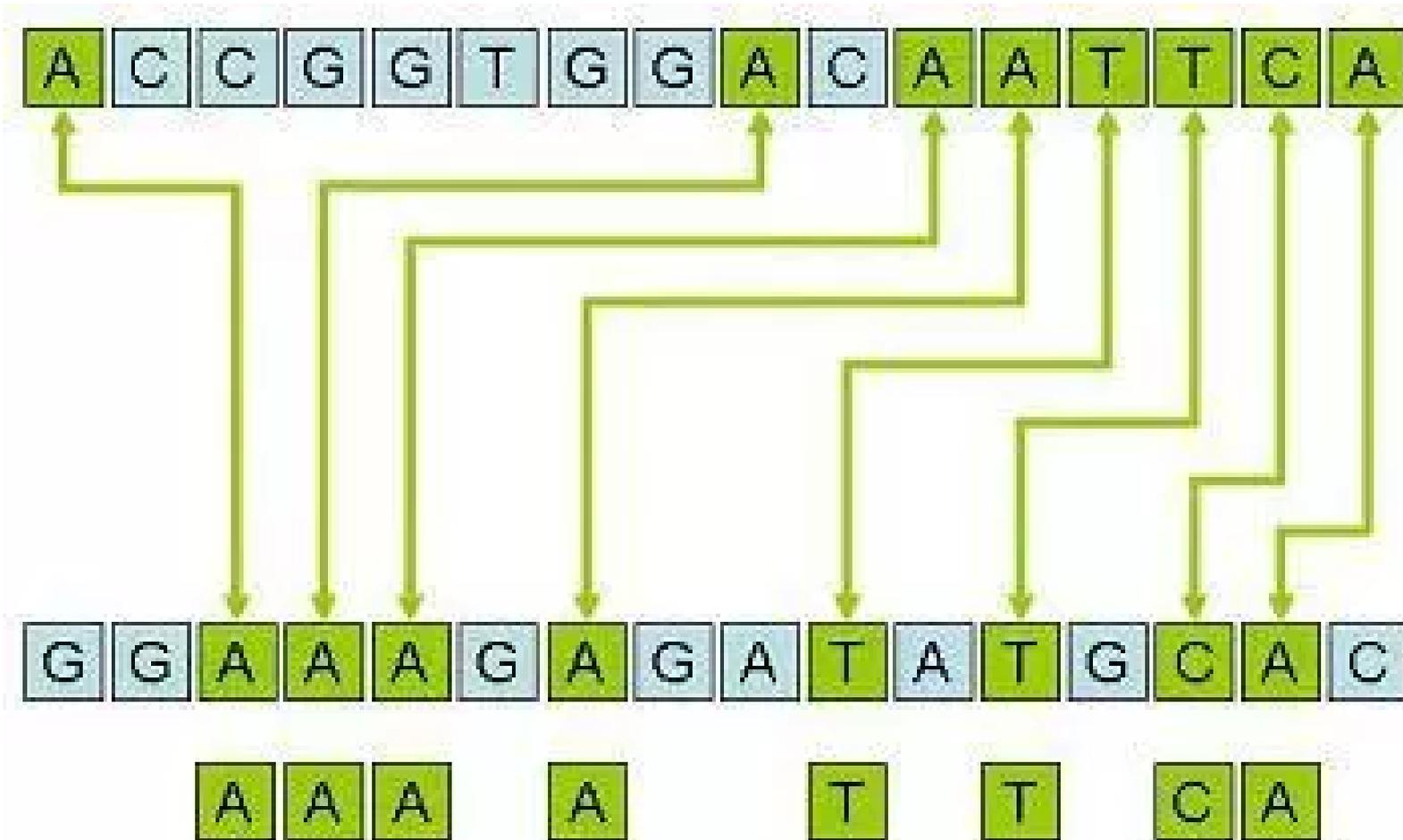
# Y la bioinformática???

- Distancia de Edición

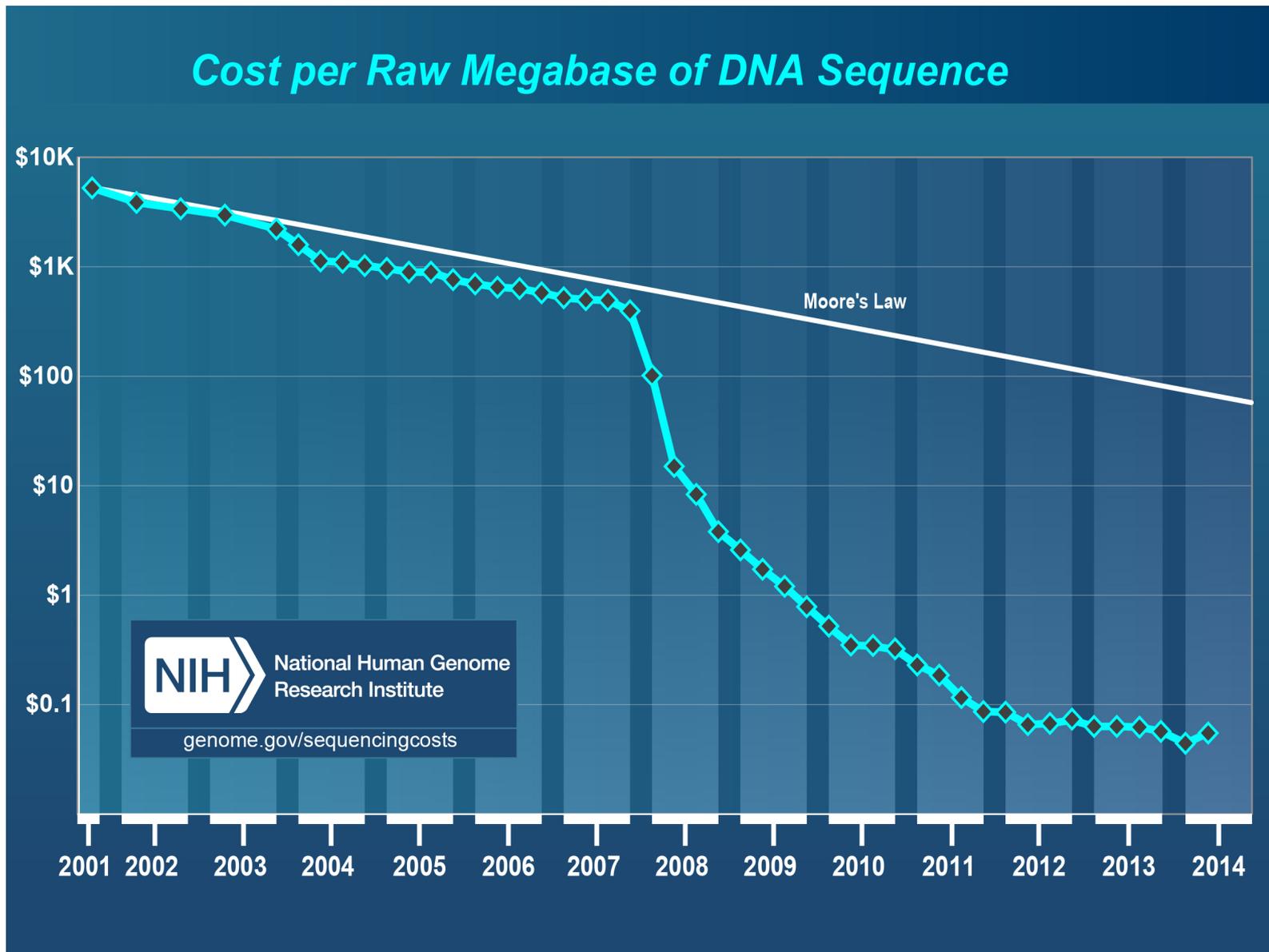
$$H = \begin{matrix} & & \boxed{A} & \boxed{C} & \boxed{A} & \boxed{C} & \boxed{A} & \boxed{C} & \boxed{T} & \boxed{A} \\ \begin{matrix} - \\ \boxed{A} \\ \boxed{G} \\ \boxed{C} \\ \boxed{A} \\ \boxed{C} \\ \boxed{A} \\ \boxed{C} \\ \boxed{A} \end{matrix} & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & \mathbf{2} & 1 & 2 & 1 & 2 & 1 & 0 & 2 \\ & 0 & \mathbf{1} & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ & 0 & 0 & \mathbf{3} & 2 & 3 & 2 & 3 & 2 & 1 \\ & 0 & 2 & 2 & \mathbf{5} & 4 & 5 & 4 & 3 & 4 \\ & 0 & 1 & 4 & 4 & \mathbf{7} & 6 & 7 & 6 & 5 \\ & 0 & 2 & 3 & 6 & 6 & \mathbf{9} & 8 & 7 & 8 \\ & 0 & 1 & 4 & 5 & 8 & 8 & \mathbf{11} & \mathbf{10} & 9 \\ & 0 & 2 & 3 & 6 & 7 & 10 & 10 & 10 & \mathbf{12} \end{matrix}$$

# Y la bioinformática???

- **Longest Common Subsequence**

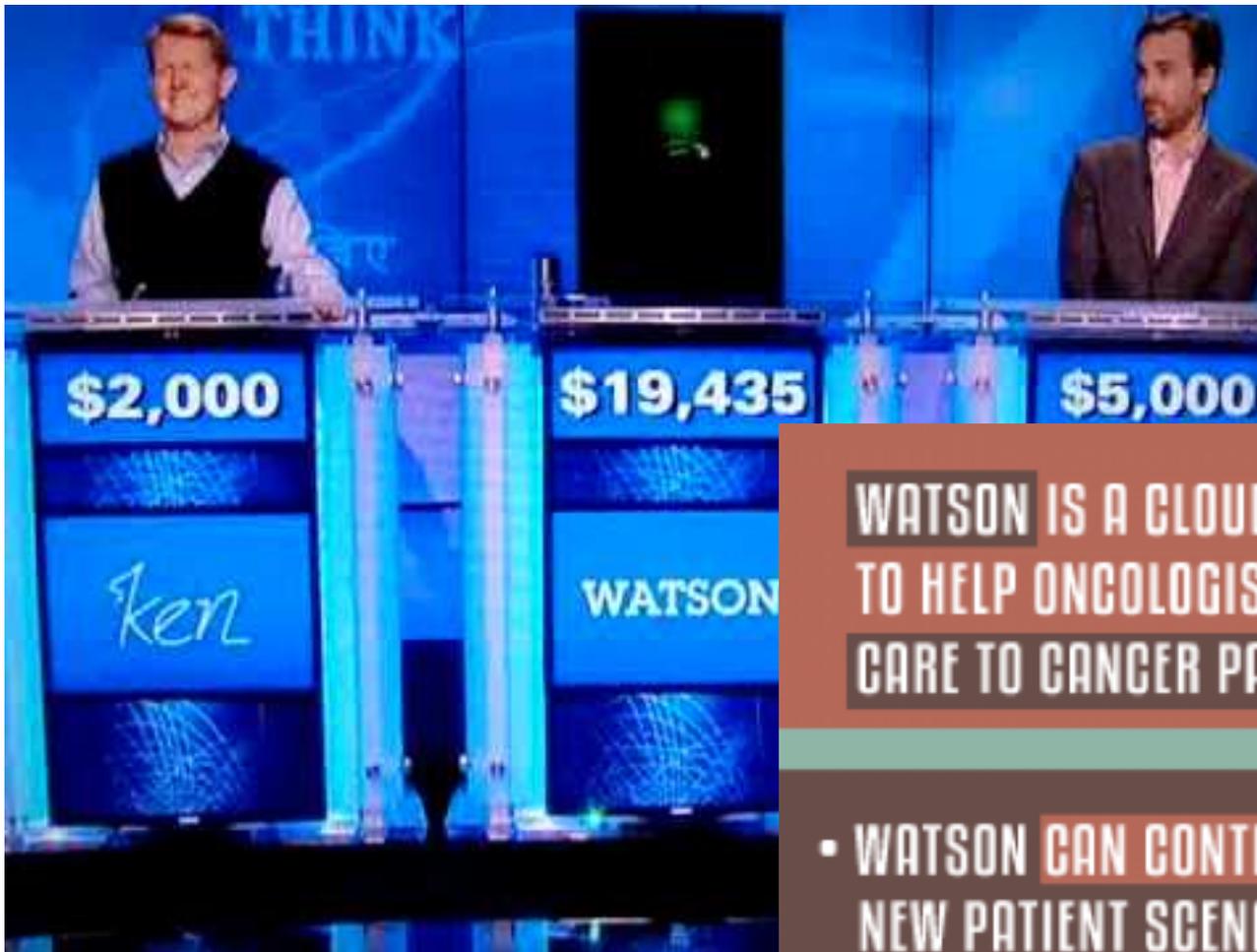


# Genómica vs Ley de Moore



# CS en los “bordes”...

- **Watson (2008)**



**WATSON IS A CLOUD-BASED PROTOTYPE DESIGNED TO HELP ONCOLOGISTS DELIVER MORE PERSONALIZED CARE TO CANCER PATIENTS**

- **WATSON CAN CONTINUALLY LEARN THROUGH NEW PATIENT SCENARIOS AND NEW RESEARCH, JOURNAL ARTICLES AND CLINICAL STUDIES**

# CS en los “bordes”...

- **The Next Rembrandt**



# CS en los “bordes”...

- **The Next Rembrandt**



# CS en los “bordes”...

- **Deep Dream**





# Preguntas



Brought to you by STC's ImmsLink  
© 2015 Scientific Technologies Corporation



“ Today I’m going to download  
the original anti-virus  
software into you. ”



mu**Ch**As **Gr**ACiAs!

**Gabriel T**olosa  
tolosoft@unlu.edu.ar