

P vs NP,
o qué tienen en común jugar al Sudoku y curar el cáncer

Iván Arcuschin

Semana de la Computación

Miércoles 14 de Junio, 2017



- Existencia de Yang-Mills y del salto de masa
- La hipótesis de Riemann
- **P versus NP**
- Las ecuaciones de Navier-Stokes
- La conjetura de Hodge
- ~~La conjetura de Poincaré~~
- La conjetura de Birch y Swinnerton-Dyer

- Existencia de Yang-Mills y del salto de masa
- La hipótesis de Riemann
- **P versus NP**
- Las ecuaciones de Navier-Stokes
- La conjetura de Hodge
- ~~La conjetura de Poincaré~~
- La conjetura de Birch y Swinnerton-Dyer





Clementina: la primer computadora científica en Argentina (1961).

Problemas para los cuales hay programas rápidos

Problemas para los cuales se desconoce si hay una forma rápida de resolverlos

Problemas para los cuales se sabe que no hay programas rápidos

Problemas para los cuales hay programas rápidos

- Multiplicar dos números.
- Ordenar una lista de nombres alfabéticamente.
- Buscar una palabra en un diccionario.

Problemas para los cuales se desconoce si hay una forma rápida de resolverlos

Problemas para los cuales se sabe que no hay programas rápidos

Problemas para los cuales hay programas rápidos

- Multiplicar dos números.
- Ordenar una lista de nombres alfabéticamente.
- Buscar una palabra en un diccionario.

Problemas para los cuales se desconoce si hay una forma rápida de resolverlos

Problemas para los cuales se sabe que no hay programas rápidos

- “Jugar al ajedrez de manera perfecta”.

Problemas para los cuales hay programas rápidos

- Multiplicar dos números.
- Ordenar una lista de nombres alfabéticamente.
- Buscar una palabra en un diccionario.

Problemas para los cuales se desconoce si hay una forma rápida de resolverlos

- Encontrar si un predicado lógico tiene solución o no.
- Elegir la ruta más corta para ir de un lugar a otro que pase por varias ciudades.

Problemas para los cuales se sabe que no hay programas rápidos

- “Jugar al ajedrez de manera perfecta”.

Problemas para los cuales hay programas rápidos

- Multiplicar dos números.
- Ordenar una lista de nombres alfabéticamente.
- Buscar una palabra en un diccionario.

Problemas para los cuales se desconoce si hay una forma rápida de resolverlos

- Encontrar si un predicado lógico tiene solución o no.
- Elegir la ruta más corta para ir de un lugar a otro que pase por varias ciudades.
- Ver si un número es primo o no.

Problemas para los cuales se sabe que no hay programas rápidos

- “Jugar al ajedrez de manera perfecta”.

Problemas para los cuales hay programas rápidos

- Multiplicar dos números.
- Ordenar una lista de nombres alfabéticamente.
- Buscar una palabra en un diccionario.
- Ver si un número es primo o no.

Problemas para los cuales se desconoce si hay una forma rápida de resolverlos

- Encontrar si un predicado lógico tiene solución o no.
- Elegir la ruta más corta para ir de un lugar a otro que pase por varias ciudades.

Problemas para los cuales se sabe que no hay programas rápidos

- “Jugar al ajedrez de manera perfecta”.

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos:

Cantidad de pasos para resolver:

Resolver un Sudoku

Como lo resolvemos:

Cantidad de pasos para resolver:

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos: multiplicamos cada dígito de B por cada dígito de A, y sumamos.

Cantidad de pasos para resolver:

Resolver un Sudoku

Como lo resolvemos:

Cantidad de pasos para resolver:

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos: multiplicamos cada dígito de B por cada dígito de A, y sumamos.
Cantidad de pasos para resolver: cantidad de dígitos al cuadrado.

Resolver un Sudoku

Como lo resolvemos:
Cantidad de pasos para resolver:

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos: multiplicamos cada dígito de B por cada dígito de A, y sumamos.
Cantidad de pasos para resolver: cantidad de dígitos al cuadrado.

Resolver un Sudoku

Como lo resolvemos: para cada celda vacía, pruebo un número del 1 al 9.
Cantidad de pasos para resolver:

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos: multiplicamos cada dígito de B por cada dígito de A, y sumamos.
Cantidad de pasos para resolver: cantidad de dígitos al cuadrado.

Resolver un Sudoku

Como lo resolvemos: para cada celda vacía, pruebo un número del 1 al 9.
Cantidad de pasos para resolver: 9 elevado a la cantidad de celdas vacías.

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos: multiplicamos cada dígito de B por cada dígito de A, y sumamos.
Cantidad de pasos para resolver: cantidad de dígitos al cuadrado.

- Caso 1: ambos números tienen 2 dígitos

Resolver un Sudoku

Como lo resolvemos: para cada celda vacía, pruebo un número del 1 al 9.
Cantidad de pasos para resolver: 9 elevado a la cantidad de celdas vacías.

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos: multiplicamos cada dígito de B por cada dígito de A, y sumamos.
Cantidad de pasos para resolver: cantidad de dígitos al cuadrado.

- Caso 1: ambos números tienen 2 dígitos $\implies 2^2 = 4$ pasos

Resolver un Sudoku

Como lo resolvemos: para cada celda vacía, pruebo un número del 1 al 9.
Cantidad de pasos para resolver: 9 elevado a la cantidad de celdas vacías.

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos: multiplicamos cada dígito de B por cada dígito de A, y sumamos.
Cantidad de pasos para resolver: cantidad de dígitos al cuadrado.

- Caso 1: ambos números tienen 2 dígitos $\implies 2^2 = 4$ pasos

Resolver un Sudoku

Como lo resolvemos: para cada celda vacía, pruebo un número del 1 al 9.
Cantidad de pasos para resolver: 9 elevado a la cantidad de celdas vacías.

- Caso 1: un Sudoku con 2 celdas vacías

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos: multiplicamos cada dígito de B por cada dígito de A, y sumamos.
Cantidad de pasos para resolver: cantidad de dígitos al cuadrado.

- Caso 1: ambos números tienen 2 dígitos $\implies 2^2 = 4$ pasos

Resolver un Sudoku

Como lo resolvemos: para cada celda vacía, pruebo un número del 1 al 9.
Cantidad de pasos para resolver: 9 elevado a la cantidad de celdas vacías.

- Caso 1: un Sudoku con 2 celdas vacías $\implies 9^2 = 81$ pasos

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos: multiplicamos cada dígito de B por cada dígito de A, y sumamos.
Cantidad de pasos para resolver: cantidad de dígitos al cuadrado.

- Caso 1: ambos números tienen 2 dígitos $\implies 2^2 = 4$ pasos
- Caso 2: ambos números tienen 4 dígitos

Resolver un Sudoku

Como lo resolvemos: para cada celda vacía, pruebo un número del 1 al 9.
Cantidad de pasos para resolver: 9 elevado a la cantidad de celdas vacías.

- Caso 1: un Sudoku con 2 celdas vacías $\implies 9^2 = 81$ pasos

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos: multiplicamos cada dígito de B por cada dígito de A, y sumamos.
Cantidad de pasos para resolver: cantidad de dígitos al cuadrado.

- Caso 1: ambos números tienen 2 dígitos $\implies 2^2 = 4$ pasos
- Caso 2: ambos números tienen 4 dígitos $\implies 4^2 = 16$ pasos

Resolver un Sudoku

Como lo resolvemos: para cada celda vacía, pruebo un número del 1 al 9.
Cantidad de pasos para resolver: 9 elevado a la cantidad de celdas vacías.

- Caso 1: un Sudoku con 2 celdas vacías $\implies 9^2 = 81$ pasos

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos: multiplicamos cada dígito de B por cada dígito de A, y sumamos.
Cantidad de pasos para resolver: cantidad de dígitos al cuadrado.

- Caso 1: ambos números tienen 2 dígitos $\implies 2^2 = 4$ pasos
- Caso 2: ambos números tienen 4 dígitos $\implies 4^2 = 16$ pasos

Resolver un Sudoku

Como lo resolvemos: para cada celda vacía, pruebo un número del 1 al 9.
Cantidad de pasos para resolver: 9 elevado a la cantidad de celdas vacías.

- Caso 1: un Sudoku con 2 celdas vacías $\implies 9^2 = 81$ pasos
- Caso 2: un Sudoku con 4 celdas vacías

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos: multiplicamos cada dígito de B por cada dígito de A, y sumamos.
Cantidad de pasos para resolver: cantidad de dígitos al cuadrado.

- Caso 1: ambos números tienen 2 dígitos $\implies 2^2 = 4$ pasos
- Caso 2: ambos números tienen 4 dígitos $\implies 4^2 = 16$ pasos

Resolver un Sudoku

Como lo resolvemos: para cada celda vacía, pruebo un número del 1 al 9.
Cantidad de pasos para resolver: 9 elevado a la cantidad de celdas vacías.

- Caso 1: un Sudoku con 2 celdas vacías $\implies 9^2 = 81$ pasos
- Caso 2: un Sudoku con 4 celdas vacías $\implies 9^4 = 6.561$ pasos

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos: multiplicamos cada dígito de B por cada dígito de A, y sumamos.
Cantidad de pasos para resolver: cantidad de dígitos al cuadrado.

- Caso 1: ambos números tienen 2 dígitos $\implies 2^2 = 4$ pasos
- Caso 2: ambos números tienen 4 dígitos $\implies 4^2 = 16$ pasos
- Caso 3: ambos números tienen 20 dígitos

Resolver un Sudoku

Como lo resolvemos: para cada celda vacía, pruebo un número del 1 al 9.
Cantidad de pasos para resolver: 9 elevado a la cantidad de celdas vacías.

- Caso 1: un Sudoku con 2 celdas vacías $\implies 9^2 = 81$ pasos
- Caso 2: un Sudoku con 4 celdas vacías $\implies 9^4 = 6.561$ pasos

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos: multiplicamos cada dígito de B por cada dígito de A, y sumamos.
Cantidad de pasos para resolver: cantidad de dígitos al cuadrado.

- Caso 1: ambos números tienen 2 dígitos $\implies 2^2 = 4$ pasos
- Caso 2: ambos números tienen 4 dígitos $\implies 4^2 = 16$ pasos
- Caso 3: ambos números tienen 20 dígitos $\implies 20^2 = 400$ pasos

Resolver un Sudoku

Como lo resolvemos: para cada celda vacía, pruebo un número del 1 al 9.
Cantidad de pasos para resolver: 9 elevado a la cantidad de celdas vacías.

- Caso 1: un Sudoku con 2 celdas vacías $\implies 9^2 = 81$ pasos
- Caso 2: un Sudoku con 4 celdas vacías $\implies 9^4 = 6.561$ pasos

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos: multiplicamos cada dígito de B por cada dígito de A, y sumamos.
Cantidad de pasos para resolver: cantidad de dígitos al cuadrado.

- Caso 1: ambos números tienen 2 dígitos $\implies 2^2 = 4$ pasos
- Caso 2: ambos números tienen 4 dígitos $\implies 4^2 = 16$ pasos
- Caso 3: ambos números tienen 20 dígitos $\implies 20^2 = 400$ pasos

Resolver un Sudoku

Como lo resolvemos: para cada celda vacía, pruebo un número del 1 al 9.
Cantidad de pasos para resolver: 9 elevado a la cantidad de celdas vacías.

- Caso 1: un Sudoku con 2 celdas vacías $\implies 9^2 = 81$ pasos
- Caso 2: un Sudoku con 4 celdas vacías $\implies 9^4 = 6.561$ pasos
- Caso 3: un Sudoku con 20 celdas vacías

Pero... ¿Qué tan rápido?

Comparemos dos programas:

Multiplicar dos números A y B

Como lo resolvemos: multiplicamos cada dígito de B por cada dígito de A, y sumamos.
Cantidad de pasos para resolver: cantidad de dígitos al cuadrado.

- Caso 1: ambos números tienen 2 dígitos $\implies 2^2 = 4$ pasos
- Caso 2: ambos números tienen 4 dígitos $\implies 4^2 = 16$ pasos
- Caso 3: ambos números tienen 20 dígitos $\implies 20^2 = 400$ pasos

Resolver un Sudoku

Como lo resolvemos: para cada celda vacía, pruebo un número del 1 al 9.
Cantidad de pasos para resolver: 9 elevado a la cantidad de celdas vacías.

- Caso 1: un Sudoku con 2 celdas vacías $\implies 9^2 = 81$ pasos
- Caso 2: un Sudoku con 4 celdas vacías $\implies 9^4 = 6.561$ pasos
- Caso 3: un Sudoku con 20 celdas vacías $\implies 9^{20} = 1,215 * 10^{19}$ pasos

Pero yo tengo el último Intel i7! Seguro tarda menos. . .

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

Sudoku con 30 celdas ocupadas (51 celdas libres)

Usando la misma cuenta que antes, un Sudoku con 51 celdas vacías llevaría $9^{51} = 4,63 * 10^{48}$ pasos.

Pero yo tengo el último Intel i7! Seguro tarda menos. . .

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

Sudoku con 30 celdas ocupadas (51 celdas libres)

Usando la misma cuenta que antes, un Sudoku con 51 celdas vacías llevaría $9^{51} = 4,63 * 10^{48}$ pasos.

Resolverlo en un Intel Core i7

El Intel Core i7 más nuevo tiene 3.10 GHz. Eso significa que hace aproximadamente $3,1 * 10^9$ ciclos por segundo. Si somos generosos y asumimos que en cada ciclo el CPU revisa una combinación. . . Entonces tardaría aproximadamente $1,49 * 10^{39}$ segundos. . . Algo así como $4,72 * 10^{31}$ años.

La edad actual del Universo se calcula en 14 billones de años: $1,4 * 10^{10}$.

Pero yo tengo el último Intel i7! Seguro tarda menos. . .

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

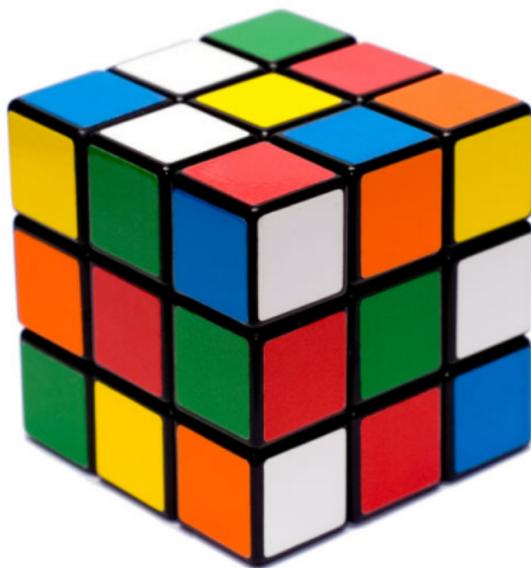
Sudoku con 30 celdas ocupadas (51 celdas libres)

Usando la misma cuenta que antes, un Sudoku con 51 celdas vacías llevaría $9^{51} = 4,63 * 10^{48}$ pasos.

Resolverlo en la supercomputadora Titan

Titan es una supercomputadora con un pico máximo de velocidad de 27 petaFLOPS por ciclo. Si lo comparamos con el Intel Core i7 que puede hacer 8 FLOPS por ciclo, diríamos que es $3.4 * 10^{26}$ veces más potente. O sea, que hace $1,054 * 10^{36}$ ciclos por segundo. En definitiva, Titan tardaría en resolverlo. . . Aproximadamente $4,39 * 10^{12}$ segundos. . . Algo así como 139.113 años.







P

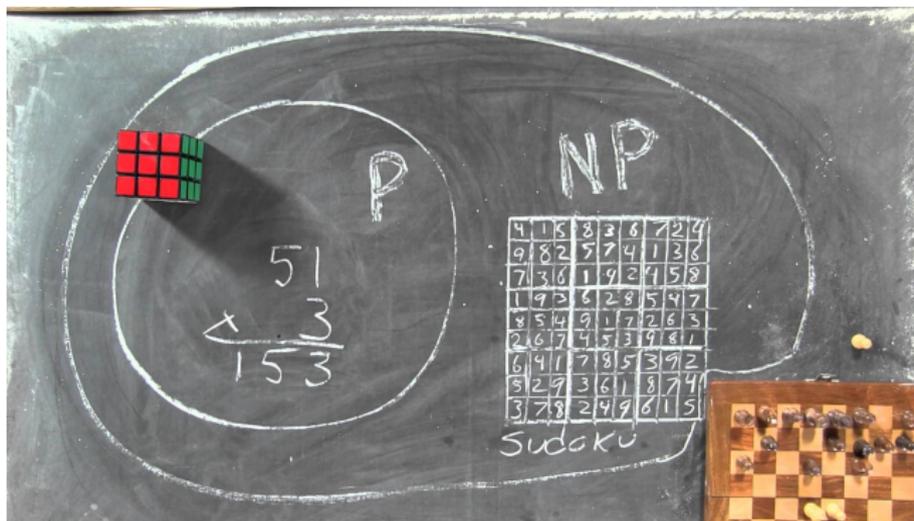
$$\begin{array}{r} 51 \\ \leftarrow 3 \\ \hline 153 \end{array}$$

NP

4	15	8	3	8	7	2	4	
9	8	2	5	7	4	1	3	6
7	3	6	1	9	2	4	5	8
1	9	3	6	2	8	5	4	7
8	5	4	9	1	7	2	6	3
2	6	7	4	5	3	9	8	1
6	4	1	7	8	5	3	9	2
5	2	9	3	6	1	8	7	4
3	7	8	2	4	9	6	1	5

Sudoku





Idea intuitiva

- En la clase de problemas **P** están los que son fáciles de resolver.
- En la clase de problemas **NP** están los que es fácil verificar si una solución es correcta o no.

La pregunta del millón de dolares

¿ $P = NP$ o $P \neq NP$?

¿ $P = NP$ o $P \neq NP$?

Es decir

- ¿Encontraremos en algún momento un programa rápido para cada problema que existe? ($P = NP$)

¿ $P = NP$ o $P \neq NP$?

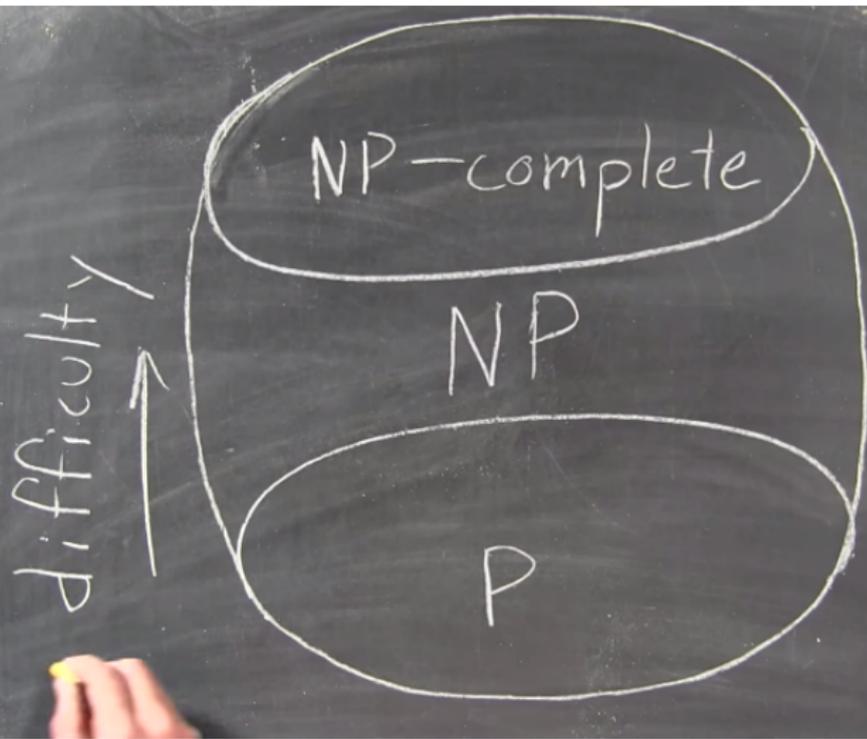
Es decir

- ¿Encontraremos en algún momento un programa rápido para cada problema que existe? ($P = NP$)
- ¿O de verdad hay algunos problemas más difíciles de resolver que otros? ($P \neq NP$)

¿Ser capaz de reconocer rápidamente una solución correcta significa que también hay una forma rápida de encontrarla?

¿Ser capaz de reconocer rápidamente una solución correcta significa que también hay una forma rápida de encontrarla?

¿Ser capaces de apreciar una melodía significa que también somos capaces de componerla?



Ejemplos de problemas NP-Complejos

- El problema del cartero: empezando desde el correo, pasar por todas las direcciones de los destinatarios, minimizando la cantidad de cuadras a recorrer.

Ejemplos de problemas NP-Complejos

- El problema del cartero: empezando desde el correo, pasar por todas las direcciones de los destinatarios, minimizando la cantidad de cuadras a recorrer.
- Asignar horarios adecuados para cada materia, con tal de minimizar los conflictos de horarios.

Ejemplos de problemas NP-Complejos

- El problema del cartero: empezando desde el correo, pasar por todas las direcciones de los destinatarios, minimizando la cantidad de cuadras a recorrer.
- Asignar horarios adecuados para cada materia, con tal de minimizar los conflictos de horarios.
- Asignar asientos para los invitados de una boda, de forma que los amigos se sienten juntos, y los enemigos estén en mesas separadas.

Ejemplos de problemas NP-Complejos

- El problema del cartero: empezando desde el correo, pasar por todas las direcciones de los destinatarios, minimizando la cantidad de cuadras a recorrer.
- Asignar horarios adecuados para cada materia, con tal de minimizar los conflictos de horarios.
- Asignar asientos para los invitados de una boda, de forma que los amigos se sienten juntos, y los enemigos estén en mesas separadas.
- Dada una serie de axiomas y un teorema, encontrar una demostración para ese teorema (o mostrar que no existe dicha demostración)

Ejemplos de problemas NP-Completo

- El problema del cartero: empezando desde el correo, pasar por todas las direcciones de los destinatarios, minimizando la cantidad de cuadras a recorrer.
- Asignar horarios adecuados para cada materia, con tal de minimizar los conflictos de horarios.
- Asignar asientos para los invitados de una boda, de forma que los amigos se sienten juntos, y los enemigos estén en mesas separadas.
- Dada una serie de axiomas y un teorema, encontrar una demostración para ese teorema (o mostrar que no existe dicha demostración)
- Encontrar la disposición óptima de los componentes dentro de un circuito electrónico.

Ejemplos de problemas NP-Completo

- El problema del cartero: empezando desde el correo, pasar por todas las direcciones de los destinatarios, minimizando la cantidad de cuadras a recorrer.
- Asignar horarios adecuados para cada materia, con tal de minimizar los conflictos de horarios.
- Asignar asientos para los invitados de una boda, de forma que los amigos se sienten juntos, y los enemigos estén en mesas separadas.
- Dada una serie de axiomas y un teorema, encontrar una demostración para ese teorema (o mostrar que no existe dicha demostración)
- Encontrar la disposición óptima de los componentes dentro de un circuito electrónico.
- Crucigramas

Ejemplos de problemas NP-Completos

- El problema del cartero: empezando desde el correo, pasar por todas las direcciones de los destinatarios, minimizando la cantidad de cuadras a recorrer.
- Asignar horarios adecuados para cada materia, con tal de minimizar los conflictos de horarios.
- Asignar asientos para los invitados de una boda, de forma que los amigos se sienten juntos, y los enemigos estén en mesas separadas.
- Dados una serie de axiomas y un teorema, encontrar una demostración para ese teorema (o mostrar que no existe dicha demostración)
- Encontrar la disposición óptima de los componentes dentro de un circuito electrónico.
- Crucigramas
- Buscaminas

Ejemplos de problemas NP-Completo

- El problema del cartero: empezando desde el correo, pasar por todas las direcciones de los destinatarios, minimizando la cantidad de cuadras a recorrer.
- Asignar horarios adecuados para cada materia, con tal de minimizar los conflictos de horarios.
- Asignar asientos para los invitados de una boda, de forma que los amigos se sienten juntos, y los enemigos estén en mesas separadas.
- Dada una serie de axiomas y un teorema, encontrar una demostración para ese teorema (o mostrar que no existe dicha demostración)
- Encontrar la disposición óptima de los componentes dentro de un circuito electrónico.
- Crucigramas
- Buscaminas
- Tetris

Ejemplos de problemas NP-Completos

- El problema del cartero: empezando desde el correo, pasar por todas las direcciones de los destinatarios, minimizando la cantidad de cuadras a recorrer.
- Asignar horarios adecuados para cada materia, con tal de minimizar los conflictos de horarios.
- Asignar asientos para los invitados de una boda, de forma que los amigos se sienten juntos, y los enemigos estén en mesas separadas.
- Dados una serie de axiomas y un teorema, encontrar una demostración para ese teorema (o mostrar que no existe dicha demostración)
- Encontrar la disposición optima de los componentes dentro de un circuito electrónico.
- Crucigramas
- Buscaminas
- Tetris
- Super Mario Bros

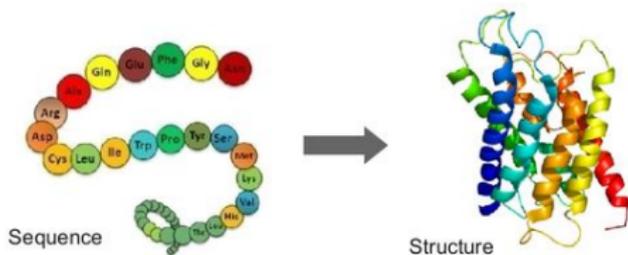
Ejemplos de problemas NP-Completo

- El problema del cartero: empezando desde el correo, pasar por todas las direcciones de los destinatarios, minimizando la cantidad de cuadras a recorrer.
- Asignar horarios adecuados para cada materia, con tal de minimizar los conflictos de horarios.
- Asignar asientos para los invitados de una boda, de forma que los amigos se sienten juntos, y los enemigos estén en mesas separadas.
- Dada una serie de axiomas y un teorema, encontrar una demostración para ese teorema (o mostrar que no existe dicha demostración)
- Encontrar la disposición óptima de los componentes dentro de un circuito electrónico.
- Crucigramas
- Buscaminas
- Tetris
- Super Mario Bros
- Sudoku

Descubriendo la estructura 3D de las proteínas

La **predicción de la estructura de las proteínas** es la predicción o cálculo de la estructura tridimensional de una proteína desde su secuencia de aminoácidos, es decir, la predicción de sus estructuras secundaria y terciaria desde su estructura primaria.

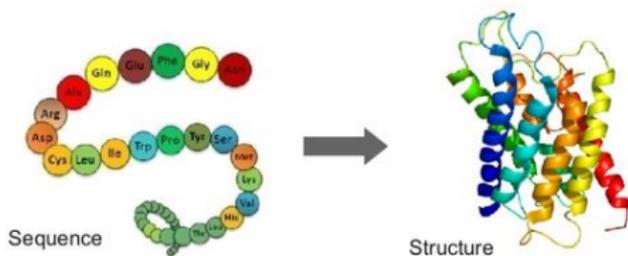
Es uno de los principales objetivos de la bioinformática y de la química teórica, y altamente importante en medicina (en diseño de fármacos, por ejemplo) y biotecnología (en el diseño de nuevas enzimas, por ejemplo).



Descubriendo la estructura 3D de las proteínas

La **predicción de la estructura de las proteínas** es la predicción o cálculo de la estructura tridimensional de una proteína desde su secuencia de aminoácidos, es decir, la predicción de sus estructuras secundaria y terciaria desde su estructura primaria.

Es uno de los principales objetivos de la bioinformática y de la química teórica, y altamente importante en medicina (en diseño de fármacos, por ejemplo) y biotecnología (en el diseño de nuevas enzimas, por ejemplo).



Este problema está **dentro de la clase de problemas NP** que todavía no tienen un programa rápido que los resuelva.

Si consiguiéramos un programa rápido para todos los problemas que existen ($P = NP$)...

Podríamos **predecir la estructura de las proteínas** eficientemente! Lo cual nos ayudaría a crear nuevas drogas de manera más rápida.

Si consiguiéramos un programa rápido para todos los problemas que existen ($P = NP$)...

Podríamos **predecir la estructura de las proteínas** eficientemente! Lo cual nos ayudaría a crear nuevas drogas de manera más rápida.

Los transportes de todo tipo se podrían organizar de manera óptima para mover gente y productos, lo cual disminuiría el tiempo y costo de los viajes.

Si consiguiéramos un programa rápido para todos los problemas que existen ($P = NP$)...

Podríamos **predecir la estructura de las proteínas** eficientemente! Lo cual nos ayudaría a crear nuevas drogas de manera más rápida.

Los transportes de todo tipo se podrían organizar de manera óptima para mover gente y productos, lo cual disminuiría el tiempo y costo de los viajes.

Las fábricas podrían aumentar su velocidad de producción y disminuir los desechos.

Si consiguiéramos un programa rápido para todos los problemas que existen ($P = NP$)...

Podríamos **predecir la estructura de las proteínas** eficientemente! Lo cual nos ayudaría a crear nuevas drogas de manera más rápida.

Los transportes de todo tipo se podrían organizar de manera óptima para mover gente y productos, lo cual disminuiría el tiempo y costo de los viajes.

Las fábricas podrían aumentar su velocidad de producción y disminuir los desechos.

Muchos de los algoritmos de **encriptación** que se utilizan actualmente para navegar internet de manera segura serían fáciles de hackear, porque están basados en problemas de factorización de números en primos (que actualmente está en NP).

Si consiguiéramos un programa rápido para todos los problemas que existen ($P = NP$)...

Podríamos **predecir la estructura de las proteínas** eficientemente! Lo cual nos ayudaría a crear nuevas drogas de manera más rápida.

Los transportes de todo tipo se podrían organizar de manera óptima para mover gente y productos, lo cual disminuiría el tiempo y costo de los viajes.

Las fábricas podrían aumentar su velocidad de producción y disminuir los deshechos.

Muchos de los algoritmos de **encriptación** que se utilizan actualmente para navegar internet de manera segura serían fáciles de hackear, porque están basados en problemas de factorización de números en primos (que actualmente está en NP).

Podríamos probar los 5 Problemas del Milenio restantes :)



En palabras de Scott Aaronson, Investigador del MIT:

Si $P = NP$, entonces el mundo sería un lugar muy diferente de lo que creemos que es. No habría ningún valor especial en los “saltos creativos”, ninguna diferencia fundamental entre resolver un problema y reconocer la solución una vez que se encontró. Cualquier persona que pudiera apreciar una sinfonia sería Mozart; Cualquiera que pudiera seguir un argumento paso a paso sería Gauss.