

¿P es igual a NP? Respuesta del Milenio. Método para Resolver Rompecabezas con el Algoritmo de Dios

Nilson Rafael Bolívar Barrios¹

nilson_bolivar@hotmail.com

<https://orcid.org/0000-0002-3933-0969>

Investigador Independiente

RESUMEN

La complejidad de cualquier tipo de problemas puede ser desde lo más fácil hasta infinitamente complejo, hay problemas donde existen atajos con los algoritmos de Dios y se pueden resolver en tiempo polinómico en este caso NP está en P, y hay problemas que con los algoritmos de Dios no se resuelve en tiempo polinómico en este caso NP no está en P, cuando tengamos un problema lo resolveremos con el algoritmo más óptimo o con el algoritmo de Dios porque obviamente no existe un algoritmo mejor que el algoritmo de Dios, pueden existir 1, 2 o más algoritmo de Dios que resolverán el mismo problema con la misma cantidad de pasos, es imposible encontrar un algoritmo que resuelva el problema en menos pasos que el algoritmo de Dios. Obviamente si un algoritmo resuelve el problema en más pasos que el algoritmo de Dios entonces no es el algoritmo de Dios. Como resultado utilizamos el mejor algoritmo que es el algoritmo de Dios y observamos en cuantos pasos se resuelve este problema, si son muchos pasos o pocos pasos y tomamos la decisión si vale la pena dedicarle tiempo en resolver ese problema. Es imposible resolver el problema en menos pasos que la cantidad de pasos del algoritmo de Dios.

Palabras clave: Algoritmo de Dios; complejidad; solución; problemas P; problemas NP.

¹ Autor principal

Correspondencia: nilson_bolivar@hotmail.com

Is P equal to NP? Millennium Response. Method to Solve Puzzles with God's Algorithm

ABSTRACT

The complexity of any type of problem can be from the easiest to infinitely complex, there are problems where there are shortcuts with God's algorithms and they can be solved in polynomial time, in this case NP is in P, and there are problems that with the algorithms of God is not solved in polynomial time, in this case NP is not in P, when we have a problem we will solve it with the most optimal algorithm or with God's algorithm because obviously there is no better algorithm than God's algorithm, there can be 1, 2 or more God algorithms that will solve the same problem with the same number of steps, it is impossible to find an algorithm that solves the problem in less steps than God's algorithm. Obviously if an algorithm solves the problem in more steps than God's algorithm then it is not God's algorithm. As a result, we use the best algorithm, which is God's algorithm, and we observe in how many steps this problem is solved, if there are many steps or few steps, and we make the decision if it is worth spending time solving that problem. It is impossible to solve the problem in fewer steps than the number of steps in God's algorithm.

Keywords: *Algorithm of God; complexity; solution; P problems; NP problems.*

Artículo recibido 15 julio 2023

Aceptado para publicación: 15 agosto 2023

Introducción

La pregunta "¿P es igual a NP?" ha desafiado y fascinado a la comunidad matemática y de ciencias de la computación durante décadas. Uno de los siete "Problemas del Milenio", esta conjetura se refiere a las clases de problemas que son inherentemente difíciles y aquellos que, aunque difíciles de resolver, tienen soluciones que pueden ser verificadas rápidamente. Si se demuestra que $P=NP$, implicaría que cada problema cuya solución puede ser verificada rápidamente también puede ser resuelto rápidamente, revolucionando nuestra comprensión de la teoría de la complejidad computacional.

Más allá de la teoría pura, la conjetura tiene ramificaciones prácticas en áreas que van desde la criptografía hasta la optimización de operaciones. En el contexto de los rompecabezas, un dominio que ha sido históricamente fructífero para visualizar y experimentar con problemas NP, se ha planteado la cuestión de si existe una metodología universal que pueda resolver eficientemente cualquier rompecabezas. Este "Algoritmo de Dios", como se ha venido a llamar, sería una suerte de procedimiento óptimo que podría, en teoría, descifrar cualquier rompecabezas con el menor número de movimientos posibles.

En este artículo, exploraremos las profundidades del problema $P=NP$, su significado y su relevancia, y nos adentraremos en una exploración del "Algoritmo de Dios", considerando su viabilidad y aplicabilidad en el contexto de rompecabezas complejos. A través de análisis rigurosos, experimentación práctica y revisión de la literatura existente, buscamos ofrecer una perspectiva fresca y comprensiva sobre uno de los mayores enigmas de nuestra era en el campo de la informática teórica.

OBJETIVOS

Saber que hay problemas NP que no están en P, y que hay problemas NP que si están en P. Utilizar el algoritmo de Dios para resolver los problemas. Recordar que la complejidad de los problemas puede llegar hasta el infinito.

DESARROLLO

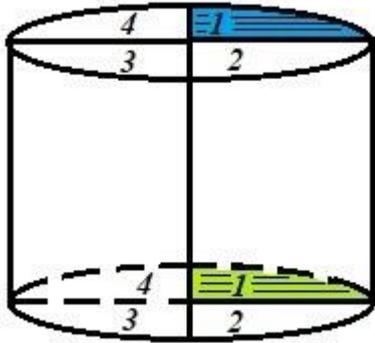
Según Nilson Bolívar (2020), se explica cómo armar un rompecabezas llamado Pyraminx Duo con el algoritmo de Dios, también se da el método para armar el cubo de Rubik u otros rompecabezas en 3 dimensiones con el algoritmo de Dios.

Según Elvira Mayordomo (2012) P suele identificarse con la clase de problemas factibles, o problemas que pueden ser resuelto en la práctica. Evidentemente, un límite de tiempo polinomial puede ser enorme debido tanto a la constante multiplicativa y el grado, pero hay dos razones prácticas por las que pensamos de P como los problemas de solución eficiente.

Según Fortnow, Lance (2009) ¿Cuál es el problema P versus NP? supongamos que tenemos un gran grupo de estudiantes que necesitamos formar parejas para trabajar en proyectos. Sabemos qué estudiantes son compatibles entre sí y queremos ponerlos en grupos compatibles de dos. Podríamos buscar todo lo posible emparejamientos, pero incluso para 40 estudiantes tendríamos más de trescientos billones de trillones de parejas posibles.

Continuando con el artículo, si tenemos un rompecabezas de un cilindro que se divide en la mitad en partes iguales con las siguientes reglas: 1) que la mitad del número de divisiones del cilindro sea un número par, 2) que cada movimiento del cilindro sea un giro de 180 grados es decir media vuelta, 3) que una de las divisiones del cilindro nunca se mueva, que se va a diferenciar porque va a estar de color azul en la parte de arriba y de color verde en la parte de abajo, tendrá el número 1, y unas líneas negras. Esta parte del cilindro nunca se va a mover porque es lo mismo mover la mitad del cilindro con los números 1 y 2 que mover la mitad del cilindro con los números 3 y 4, entonces se decide que nunca se va a mover la mitad del cilindro que tenga el color azul con el número uno y las líneas negras, no importa cuántas divisiones tenga el cilindro la parte azul con el número 1 y las líneas negras nunca se va a mover.

4) La cuarta regla es que por ejemplo si el cilindro tiene 4 divisiones el cilindro va a estar armado si en el sentido horario después de la división del cilindro que tiene el color azul, el número 1 y las líneas negras sigue la división del cilindro con el número 2, después la división del cilindro con el número 3, y por último la división del cilindro con el número 4, como se ve en la siguiente imagen.



5) Cada división que tenga el cilindro va a tener un número ejemplo 1, 2, 3, 4, etcétera, el cilindro va a estar armado cuando después de la parte del cilindro que tiene el color azul, el número 1 y las líneas negras los números de cada división del cilindro van a aumentar de 1 en 1 al sentido horario como se ve en la imagen anterior.

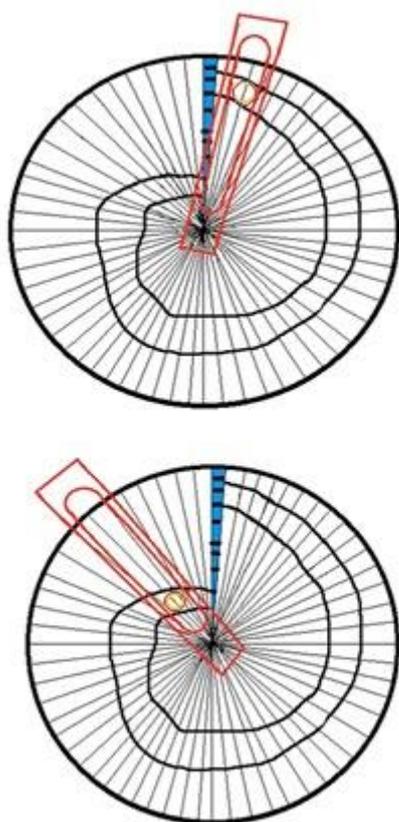
El número de combinaciones totales para este cilindro de la imagen anterior es de 3 factorial $=3 \times 2 \times 1 = 6$ combinaciones. Recordemos que la mitad de sus números de divisiones es un número par, el cilindro tiene 4 divisiones y la mitad de 4 es 2 que es un número par.

Entonces: el cilindro con la mitad de sus números de divisiones par dividido en 8 partes sus números de combinaciones es 7 factorial $=5.040$.

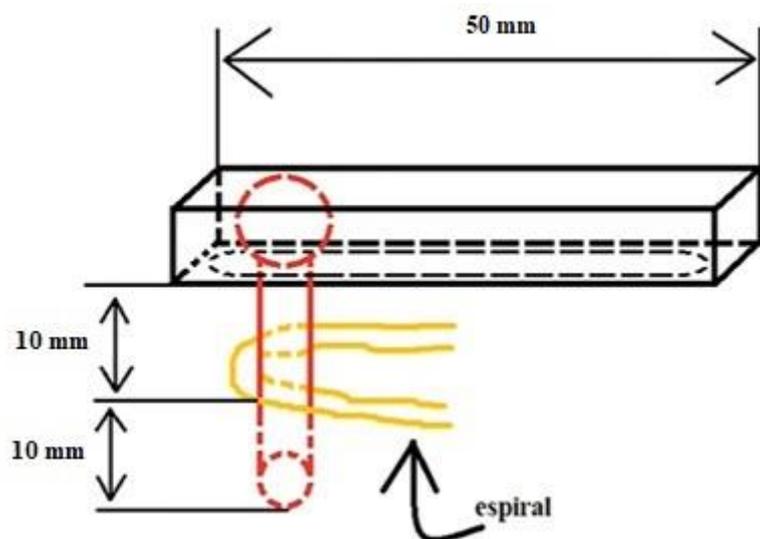
El cilindro con la mitad de sus números de divisiones par dividido en 16 sus números de combinaciones es 15 factorial, y así sucesivamente hasta el infinito.

Entre más divisiones tenga el cilindro el problema se va a volver más complejo, porque el cilindro va a tener más números de combinaciones, y porque va a tener más opciones de movimientos. Y se puede dividir el cilindro a la mitad hasta el infinito.

Ahora tenemos que hacer que este rompecabezas se pueda comprobar si está bien armado o no en un tiempo corto por ejemplo en más o menos un minuto, para eso colocamos algo parecido a la aguja de un reloj que le dé la vuelta al círculo del cilindro en más o menos un minuto, obviamente esta aguja va a estar clavada con el centro del círculo para que pueda dar casi la vuelta, que empiece en el sentido horario en la parte derecha de la parte del cilindro que esta de color azul y que termine en la parte izquierda de la parte del cilindro que tiene el color azul como se observa en la siguiente imagen.

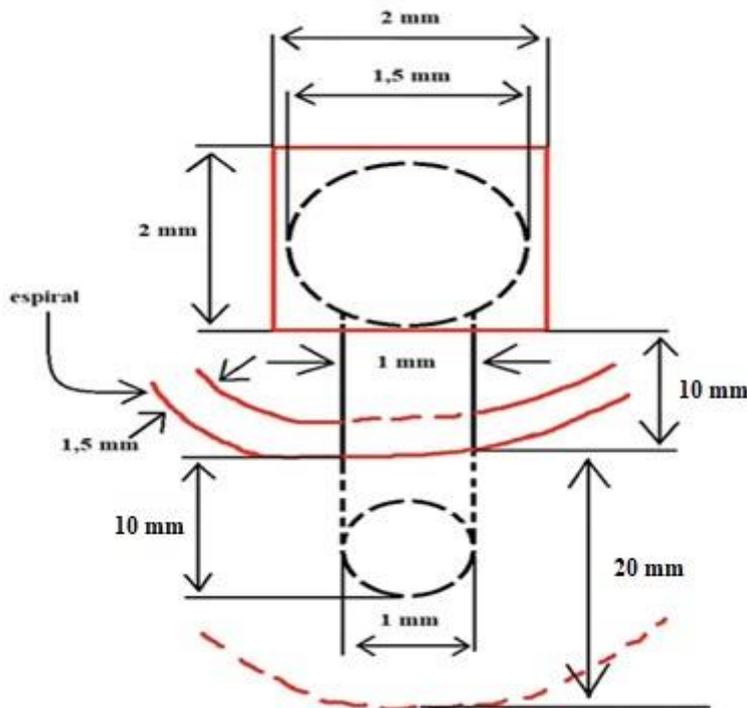


En la imagen anterior se ve también una espiral. En la siguiente imagen vemos un ejemplo del largo de lo que es parecido la aguja de un reloj, y vemos que dentro de la aguja hay una esfera que tiene una varilla soldada que entra en la espiral y a medida que la espiral se va acercando más al centro, la esfera con la varilla soldada se va acercando también cada vez más al centro.

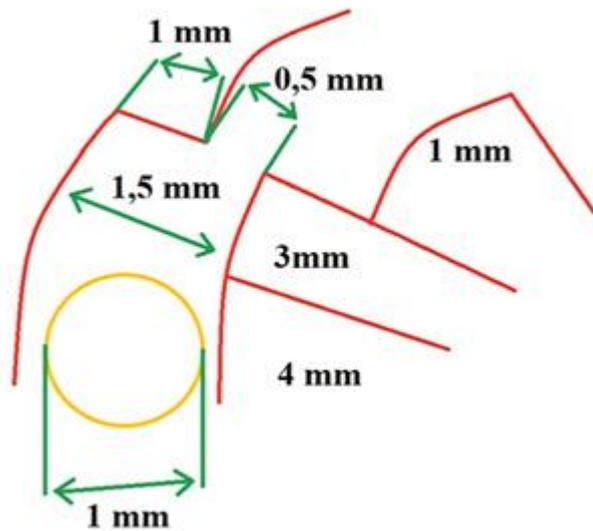


En la siguiente imagen se ve que dentro de lo que es parecido a la aguja del reloj hay una esfera y la varilla soldada en la esfera es más pequeña por ejemplo mide 1 mm y la espiral mide 1.5

milímetros de ancho, también observamos que la varilla soldada no va a rozar con el piso de la espiral porque el piso de la espiral es más profundo.



En la siguiente imagen observamos como ya sabemos que la varilla soldada tiene 1 milímetro de diámetro y la espiral 1,5 milímetros de ancho, si el cilindro está bien armado la espiral se va a ir acercando por ejemplo 1 milímetro al centro del círculo del cilindro en cada parte que este dividida el cilindro por ejemplo si el cilindro de 8 divisiones está bien armado en el sentido horario después del cilindro que esta de color azul las 7 partes del cilindro se van acercando la espiral 1 milímetro en cada división del cilindro por ejemplo si al lado de la parte del cilindro azul en el sentido horario la espiral que está en la parte del cilindro al lado en el sentido horario estaría a 7 milímetros del centro, la parte del cilindro que le sigue estaría a 6 milímetros del centro del círculo, la parte que le sigue al sentido horario estaría a 5 milímetros, la parte que le sigue a 4, la que le sigue a 3, la que le sigue a 2 y la última parte a 1 milímetro del centro, esto ocurre si el cilindro está bien armado y la varilla soldada llegaría hasta el final de la espiral, pero si el cilindro está mal armado por ejemplo si la parte del cilindro que está a 3 milímetros del centro sigue la parte que está a 1 milímetro del centro entonces la varilla soldada no puede pasar porque queda 0,5 milímetros de ancho como se ve en la siguiente imagen a continuación.

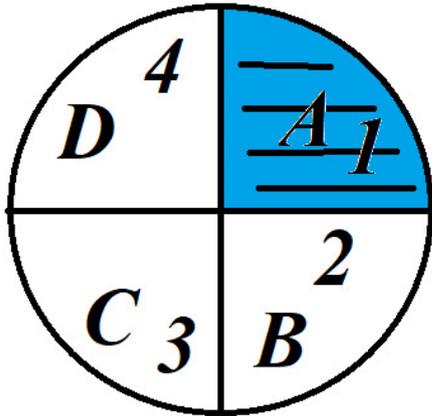


En la imagen anterior comprobamos en más o menos 1 minuto que el cilindro está mal armado así el cilindro tenga muchísimas divisiones nosotros comprobaríamos si está bien o mal armado dependiendo si la aguja llega o no llega al final de la espiral, entonces sería un problema NP porque se puede comprobar fácilmente en un tiempo polinómico.

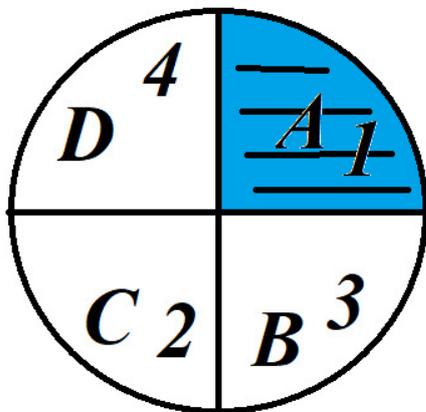
Todos los rompecabezas tienen un algoritmo de Dios y los cilindros de este documento también tienen algoritmo de Dios.

Armemos el cilindro con el algoritmo de Dios

Parece imposible para un ser humano armar un rompecabezas con el algoritmo de Dios, pero es posible usando este método, el cilindro está dividido en 4 partes y a cada parte del cilindro vamos a colocarle una letra A, B, C, D, y también a cada parte del cilindro vamos a colocarle un número, que son 1, 2, 3, 4 como se observa en la siguiente imagen a continuación.

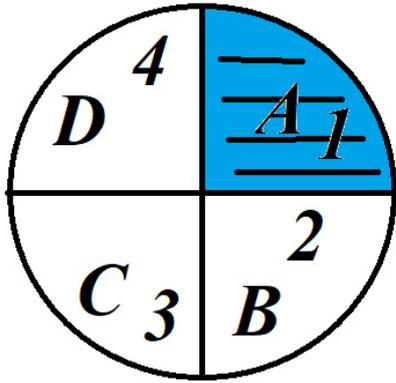


Las letras A, B, C, D son como sombras es decir que no se van a mover si el cilindro se mueve las letras no se van a mover sólo se van a mover los números, la parte del cilindro que tiene el número 2 también tiene la letra B y la parte del cilindro que tiene el número 3 tiene la letra C, recordemos que los movimientos del cilindro son giros de media vuelta, si tenemos el cilindro armado y movemos los números 2 y 3 el cilindro queda como se observa en la siguiente imagen que tenemos a continuación.



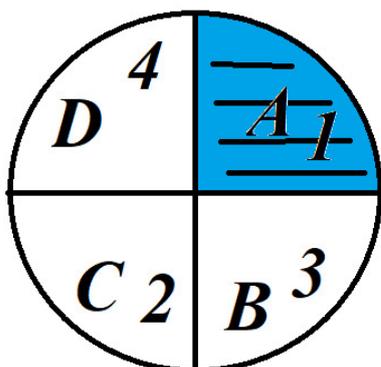
En la imagen anterior nos damos cuenta que los números 2 y 3 se mueven cuando se mueve el cilindro es como si escribiéramos los números y las letras B y C no se movieron a pesar de que el cilindro se movió es como si las letras fueran sombras, entonces las letras A, B, C, y D, no se van a mover nunca si el cilindro se mueve las letras nunca se van a mover, y los números se van a mover cuando se mueva el cilindro como se ve en la imagen anterior.

Lo que vamos a hacer es tener el cilindro armado como se observa en la siguiente imagen a continuación.



Entonces hacemos una tabla y colocamos en la tabla movimiento cero, también colocamos, parte del cilindro 1 y como la parte del cilindro 1 está en la letra A colocamos en la tabla la letra A, en la tabla donde dice parte del cilindro 2 colocamos B porque el 2 está en la letra B, en la parte del cilindro 3 colocamos la letra C porque el 3 está en la letra C, y en la parte del cilindro 4 colocamos la letra D porque el 4 está en la letra D, después colocamos en la tabla combinación como el 1 está en A, el 2 en B, el 3 en C, el 4 en D, la combinación sería ABCD, después en la tabla se pone la palabra me devuelvo y como hay cero movimientos se pone armado porque el cilindro está armado.

Ahora vamos a desarmar el cilindro con un solo movimiento, y nos damos cuenta que el cilindro tiene 2 opciones de movimientos que son mover las dos partes del cilindro que esta abajo y las dos partes del cilindro que está en la parte izquierda recordemos que la parte del cilindro que está de color azul no se mueve, entonces escogemos un rango de movimientos hagamos que el movimiento de la parte de abajo del cilindro se mueva primero que la parte izquierda del cilindro entonces desarmamos el cilindro con un solo movimiento la parte de abajo del cilindro, y el cilindro queda como se observa en la siguiente imagen a continuación.



Y en la tabla en movimiento 1 colocamos media vuelta abajo como la parte del cilindro 1 está en la letra A colocamos en la tabla la letra A, en la tabla donde dice parte del cilindro 2 colocamos C porque el 2 está en la letra C, en la parte del cilindro 3 colocamos la letra B porque el 3 está en la letra B, y en la parte del cilindro 4 colocamos la letra D porque el 4 está en la letra D, después colocamos en la tabla combinación como el 1 está en A, el 2 en C, el 3 en B, el 4 en D, la combinación sería ACBD, después en la tabla se pone la palabra me devuelvo y como hay un movimiento y el cilindro se desarmo con media vuelta abajo entonces es como si fuéramos grabado un video desarmando el cilindro y después le damos reversa al video y en el video en reversa el cilindro esta desarmado y se arma con media vuelta abajo por eso colocamos media vuelta abajo, y hacemos lo mismo con el movimiento uno media vuelta a la izquierda, también hacemos lo mismo con los movimientos dos y los movimientos tres, entonces recordemos que el número de combinaciones totales del cilindro es 3 factorial que es igual a 6, entonces como desarmamos el cilindro con 3 movimientos y los 3 movimientos nos dio 6 combinaciones totales distintas del cilindro y ya no es necesario desarmar el cilindro con 4 movimientos porque las combinaciones de los 4 movimientos se van a repetir con las combinaciones con menos movimientos lo mismo ocurrirá con las combinaciones de 5 movimientos, 6 movimientos, y así sucesivamente, nosotros vamos a desarmar el cilindro y los movimientos innecesarios se eliminan así que sólo se desarmará el cilindro hasta 3 movimientos. Porque hasta 3 movimientos hay 6 opciones diferentes de combinaciones que son ABCD, ABDC, ACBD, ACDB, ADBC y ADCB. En las siguientes tablas las letras (MVA) quieren decir media vuelta abajo, y las letras (MVI) quiere decir media vuelta izquierda. Veamos la tabla de la que estamos hablando a continuación que está dividida en 4 partes.

Tabla parte 1.

movimiento 1	movimiento 2	movimiento 3	parte del cilindro 1	parte del cilindro 2	parte del cilindro 3	parte del cilindro 4	combinación	me devuelvo
cero			A	B	C	D	ABCD	armado
movimientos								
MVA			A	C	B	D	ACBD	MVA
MVI			A	B	D	C	ABDC	MVI

Continuación de la tabla parte 2.

MVA	MVA	A	B	C	D	ABC	MVA	MVA
						D		MVA
MVA	MVI	A	D	B	C	ADB	MVI	MVA
						C		
MVI	MVA	A	C	D	B	ACD	MVA	MVI
						B		
MVI	MVI	A	B	C	D	ABC	MVI	MVI
						D		

Continuación de la tabla parte 3.

MVA	MVA	MVA	A	C	B	D	AC	MVA	MVA	MVA
							BD			
MVA	MVA	MVI	A	B	D	C	AB	MVI	MVA	MVA
							DC			
MVA	MVI	MVA	A	D	C	B	AD	MVA	MVI	MVA
							CB			
MVA	MVI	MVI	A	C	B	D	AC	MVI	MVI	MVA
							BD			

Continuación de la tabla parte 4.

MVI	MVA	MVA	A	B	D	C	ABD	MVA	MVA	MVI
							C			
MVI	MVA	MVI	A	D	C	B	ADC	MVI	MVA	MVI
							B			

MVI	MVI	MVA	A	C	B	D	ACB	MVA	MVI	MVI
							D			
MVI	MVI	MVI	A	B	D	C	ABD	MVI	MVI	MVI
							C			

Ahora ordenamos alfabéticamente las combinaciones de la tabla, y solo copiamos las partes de la tabla donde están las combinaciones y los movimientos que se devuelven, entonces donde dice me devuelvo lo reemplazamos por la frase se arma con, y colocamos una columna nueva en la tabla que se va a llamar combinación número y allí se va contando el número de combinaciones del cilindro. La combinación ABDC que se arma con 1 movimiento que es media vuelta izquierda, también se arma con 3 movimientos que son media vuelta izquierda, media vuelta abajo, media vuelta abajo, también se arma con otros 3 movimientos que son media vuelta abajo, media vuelta abajo, media vuelta izquierda, entonces el algoritmo de Dios busca la menor cantidad de movimientos posibles, la cantidad perfecta u óptima de movimientos para armar el cilindro, entonces la combinación se puede armar con un movimiento que es media vuelta izquierda o con 3 movimientos y el algoritmo de Dios escoge (uno) 1 movimiento, entonces a los 3 movimientos le colocamos eliminado y así hacemos con todas las combinaciones de la siguiente tabla, en la combinación ADCB tiene 2 opciones de armarse con 3 movimientos que son media vuelta abajo, media vuelta izquierda, media vuelta abajo, también tiene otra opción de armarse con 3 movimientos que es media vuelta izquierda, media vuelta abajo, media vuelta izquierda, entonces como las dos opciones de armarse son 3 movimientos es la misma cantidad de movimientos que tiene esa combinación esto quiere decir que el algoritmo de Dios se va a armar de 2 formas posibles con 3 movimientos, el algoritmo de Dios no es único se busca armar la combinación con la menor cantidad de movimientos posibles y si hay varias opciones con la misma cantidad de movimientos para armarse entonces se puede usar cualquiera de esas opciones. En las siguientes tablas las letras (MVA) quieren decir media vuelta abajo, y las letras (MVI) quiere decir media vuelta izquierda. Vamos enumerando las combinaciones que no son eliminadas y nos tiene que dar 6 combinaciones que es igual al número de combinaciones totales que tiene el cilindro que es

3 factorial igual a 6, todo lo que se ha explicado se observa en la siguiente tabla que está dividida en 2 partes a continuación la parte 1 de la tabla.

Combinación	se arma con			combinación número
ABCD	armado			1
ABCD	MVA	MVA		eliminado
ABCD	MVI	MVI		eliminado
ABDC	MVI			2
ABDC	MVI	MVI	MVI	eliminado
ABDC	MVI	MVA	MVA	eliminado
ABDC	MVA	MVA	MVI	eliminado

Continuación de la tabla parte 2 a continuación.

ACBD	MVA			3
ACBD	MVA	MVA	MVA	eliminado
ACBD	MVA	MVI	MVI	eliminado
ACBD	MVI	MVI	MVA	eliminado
ACDB	MVA	MVI		4
ADBC	MVI	MVA		5
ADCB	MVA	MVI	MVA	6
ADCB	MVI	MVA	MVI	6

Ahora eliminamos las combinaciones eliminadas en la tabla anterior y la tabla se va a llamar Algoritmo de Dios, reemplazamos se arma con por se arma con el algoritmo de Dios y nos queda como se observa a continuación.

Algoritmo de Dios				
combinaciones	se arma con el algoritmo de Dios			combinación número
ABCD	armado			1
ABDC	MVI			2
ACBD	MVA			3
ACDB	MVA	MVI		4
ADBC	MVI	MVA		5
ADCB	MVA	MVI	MVA	6
ADCB	MVI	MVA	MVI	6

La tabla anterior se coloca en orden alfabético para buscar las combinaciones con más facilidad. Entre más inteligentes seamos y más nos esforcemos vamos a ser mejores resolviendo problemas más complejos en otras palabras la inteligencia y el esfuerzo es directamente proporcional a ser mejor resolviendo problemas más complejos, también se puede decir que entre más inteligencia y más esfuerzo vamos a crear mejores algoritmo con la cantidad de menos pasos posibles para resolver ese problema pero recordemos que el algoritmo de Dios es el límite y no existe un algoritmo que resuelva el problema en menos pasos que el algoritmo de Dios por eso un problema de tipo P puede tener un atajo para resolver el problema en un tiempo polinómico pero si es muy complejo puede que no tengamos la inteligencia suficiente para resolverlo, o lo vamos a resolver con un algoritmo que necesite muchísimos pasos o con un algoritmo en un tiempo no polinómico.

Podemos hacer el algoritmo de Dios para el cilindro dividido en 4 partes de otra forma con inteligencia

Podemos hacer el algoritmo de Dios para el cilindro dividido en 4 partes de otra forma con inteligencia por ejemplo si el cilindro está armado es obvio que, si repetimos 2 veces el mismo movimiento por ejemplo media vuelta abajo, media vuelta abajo el cilindro se va a devolver a cero movimientos entonces quitamos las combinaciones que tengan los movimientos que se repiten de seguido 2 veces o más y así hacemos la tabla del algoritmo de Dios más rápido, pero en un cilindro con más divisiones o más complejo lo obvio se puede hacer cada vez más complejo hasta el infinito que no tendremos la inteligencia suficiente para darnos cuenta por eso se hace la tabla del algoritmo de Dios paso a paso en esta investigación por si no se dan cuenta de lo obvio. En la siguiente tabla a continuación observamos el número de combinaciones con respecto al número de movimientos del cilindro con 4 divisiones que tiene que dar 6 combinaciones totales.

Número de movimientos para armarse	0	1	2	3	combinaciones totales
con el algoritmo de Dios					
Número de combinaciones con respecto al número de movimientos para armarse con el algoritmo de Dios del cilindro dividido en 4 partes	1	2	2	1	6

La tabla anterior es una tabla incompleta del algoritmo de Dios del cilindro dividido en 4 partes porque dice cuántas combinaciones tiene con respecto al número de movimientos para armarse con el algoritmo de Dios. Lo mismo ocurre con otros rompecabezas como por ejemplo el Cubo de Rubik que está en la tabla que está dividida en 2, veamos la primera parte de la tabla a continuación.

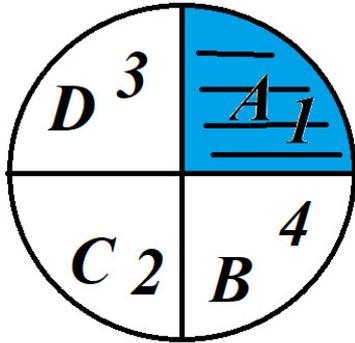
Numero de movimientos para armarse con el algoritmo de Dios	Número de combinaciones con respecto al número de movimientos para armarse con el algoritmo de Dios
0	1
1	18
2	243
3	3,24
4	43,239
5	574,908
6	7,618,438
7	100,803,036
8	1,332,343,288
9	17,596,479,795
10	232,248,063,316

Ahora observemos la segunda parte de la tabla a continuación.

11	3,063,288,809,012
12	40,374,425,656,248
13	531,653,418,284,628
14	6,989,320,578,825,358
15	91,365,146,187,124,313
16	acerca de 1,100,000,000,000,000,000
17	acerca de 12,000,000,000,000,000,000
18	acerca de 29,000,000,000,000,000,000
19	acerca de 1,500,000,000,000,000,000
20	acerca de 490,000,000

Ejercicio número 1

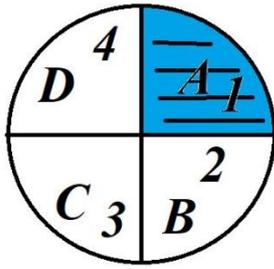
Todo lo que hicimos anteriormente es para armar el cilindro con el algoritmo de Dios, supongamos que tenemos el cilindro desarmado como se muestra en la siguiente imagen a continuación.



En la imagen anterior observamos que la combinación del cilindro es ACDB, En la siguiente tabla las letras (MVA) quieren decir media vuelta abajo, y las letras (MVI) quiere decir media vuelta izquierda. entonces buscamos en la tabla algoritmo de Dios la combinación y será fácil de buscar porque la tabla tiene las combinaciones en orden alfabético.

Algoritmo de Dios				
combinaciones	se arma con el algoritmo de Dios			combinación número
ABCD	armado			1
ABDC	MVI			2
ACBD	MVA			3
ACDB	MVA	MVI		4
ADBC	MVI	MVA		5
ADCB	MVA	MVI	MVA	6
ADCB	MVI	MVA	MVI	6

Observamos que en la tabla la combinación ACDB se arma con el algoritmo de Dios con media vuelta abajo y media vuelta izquierda y armamos el cilindro con el algoritmo de Dios como se observa en la siguiente imagen.



No existe un algoritmo que resuelva el problema en menos pasos que el algoritmo de Dios, ese algoritmo es imposible, por ejemplo, la combinación ADCB se arma con el algoritmo de Dios con 2 opciones de 3 movimientos y es imposible armarlo con menos de 3 movimientos.

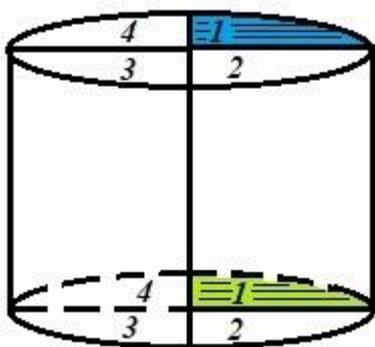
Con este método podemos armar rompecabezas en 3 dimensiones con el algoritmo de Dios como el Cubo de Rubik, el Pyraminx Duo, Cubo de Rubik 4x4, etcétera, por más complejo que parezca el rompecabezas si se puede armar con el algoritmo de Dios usando el mismo método que usamos para armar el cilindro dividido en 4 divisiones, así el cilindro se vaya dividiendo en la mitad hasta el infinito este método sirve para armarlo con el algoritmo de Dios. No existe un algoritmo que resuelva el problema en menos pasos que el algoritmo de Dios, ese algoritmo es imposible.

Continuando con el artículo todos los rompecabezas se pueden armar con el algoritmo de Dios, pero imaginemos que estamos armando un rompecabezas por ejemplo un cubo de Rubik con el algoritmo de Dios y 20 movimientos es la solución óptima para armar ese rompecabezas entonces lo armamos con 20 movimientos y grabamos un video con una cámara armándolo con 20 movimientos, después vemos el video en reversa y obviamente en el video vemos que el rompecabezas esta armado y después lo desarmamos con 20 movimientos entonces nos damos cuenta que el rompecabezas se puede armar y desarmar con el número de movimientos del algoritmo de Dios que en este caso serían 20 movimientos.

Ahora si el cilindro tiene 4 divisiones lo desarmamos con el algoritmo de Dios con 1 movimiento para después armarlo con el algoritmo de Dios con 1 movimiento, si el cilindro tiene 8 divisiones lo desarmamos con el algoritmo de Dios con 2 movimientos para después armarlo con el algoritmo de Dios con 2 movimientos, si el cilindro tiene 12 divisiones lo desarmamos con el algoritmo de Dios con 3 movimiento para después armarlo con el algoritmo de Dios con 3 movimiento y seguimos esta secuencia hasta el infinito. La secuencia es que el número de

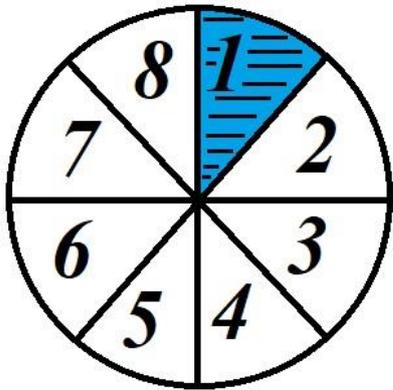
divisiones del cilindro va aumentando de 4 en 4 y el número de movimientos con que se va a armar con el algoritmo de Dios el cilindro va aumentando de 1 en 1 en otras palabras que la cantidad de movimientos con que vamos a armar el cilindro con el algoritmo de Dios es la cuarta parte de la cantidad de divisiones del cilindro.

El cilindro dividido en 4 partes tiene 3 factorial de combinaciones totales y se va a desarmar con el algoritmo de Dios con 1 movimiento entonces se puede mover las partes 2 y 3 del cilindro para después armarlo con el algoritmo de Dios, o se puede desarmar moviendo las partes 3 y 4 del cilindro para después armarlo con el algoritmo de Dios entonces como hay 2 opciones de movimientos para mover el cilindro hacemos una fórmula 2 que es el número de opciones de movimientos, y elevamos a la potencia el número de movimientos con que se va a armar y como se va a armar con 1 movimiento porque 1 es la cuarta parte de 4, entonces sería 2 elevado a la 1 que es igual a 2 que es el número de posibilidades de combinaciones con que se va a armar el cilindro con el algoritmo de Dios.



Siguiendo esta secuencia el cilindro dividido en 8 partes tiene 7 factorial de combinaciones totales y tiene 4 opciones de movimientos porque en la primera opción de movimiento se puede mover la parte del cilindro con los números 2, 3, 4, y 5, en la segunda opción se mueven los números 3, 4, 5, y 6, en la tercera opción se mueven los números 4, 5, 6, y 7, y en la cuarta opción se mueven los números 5, 6, 7, y 8, en la formula se coloca 4 porque tiene 4 opciones de movimientos y se va a armar con 2 movimientos porque 2 es la cuarta parte de 8, la formula quedaría 4 elevado a la $2 = 16$ que es el número de posibilidades de combinaciones con que se va a armar el cilindro con

el algoritmo de Dios, en la siguiente imagen a continuación observamos el cilindro dividido en 8 partes.



El cilindro dividido en 12 partes tiene 11 factorial de combinaciones totales tiene 6 opciones de movimientos en la formula se coloca 6 y se va a armar con 3 movimientos porque 3 es la cuarta parte de 12, la formula quedaría 6 elevado a la 3 = 216, que es el número de posibilidades de combinaciones con que se va a armar el cilindro con el algoritmo de Dios. y así sucesivamente hasta el infinito.

Entonces nos damos cuenta que el número de combinaciones totales del cilindro es mayor al número de posibilidades de combinaciones con que se va a armar el cilindro con el algoritmo de Dios.

Entonces sacamos el porcentaje del número de posibilidades de combinaciones para armar el cilindro dividido entre el número de combinaciones totales del cilindro por ejemplo un cilindro dividido en 4 partes su número de combinaciones totales es 3 factorial que es igual a 6 y el número de posibilidades de combinaciones para armar el cilindro con el algoritmo de Dios es 2 entonces 6 que es el número de combinaciones totales del cilindro es el 100 % y queremos saber el 2 que porcentaje es de 6 entonces dividimos 2 entre 6 y nos da el 0.33 por 100 es el 33% y en la siguiente tabla observamos que el porcentaje para el cilindro dividido en 8 es el 0,317% y observamos que el porcentaje va bajando a medida que el cilindro tiene más divisiones y así continuara bajando el porcentaje hasta el infinito, esto quiere decir que podemos armar el cilindro con el algoritmo de Dios con la cuarta parte del número que está dividido el cilindro porque el número de

combinaciones para armar el cilindro con el algoritmo de Dios va a ser menor que el número de combinaciones totales que está dividido el cilindro.

El número de combinaciones con que se va a armar con el algoritmo de Dios el cilindro es una unidad en bruto porque el algoritmo de Dios va a omitir las combinaciones innecesarias de los movimientos innecesarios esto quiere decir que el número de combinaciones con que se va a armar con el algoritmo de Dios el cilindro es un número más bajo que el que está en la siguiente tabla y el porcentaje es más bajo aún pero no hago este trabajo para resumir en lo posible este archivo y ya esto se explicó cómo hacerlo en este mismo artículo cuando hicimos la tabla del algoritmo de Dios, lo importante es saber que el porcentaje va bajando. La tabla está dividida en 2 partes observemos la primera parte a continuación.

Número de fila	Porcentaje del número de posibilidades de combinaciones para armar el cilindro dividido entre el número de combinaciones totales del cilindro mitad par	su número de posibilidades de combinaciones para armar el cilindro mitad par	Número de combinaciones totales del cilindro mitad par	Número de divisiones del cilindro mitad par
1	33,33333333 %	2	6	4

Ahora observemos la segunda parte de la tabla a continuación.

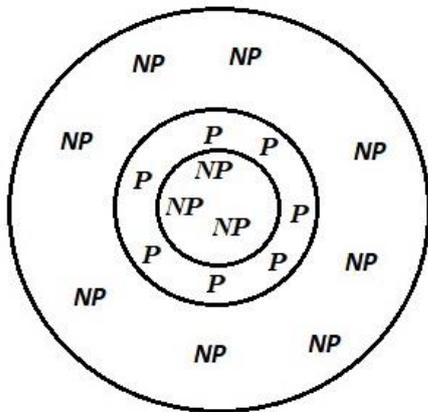
2	0,317460317 %	16	5.040	8
3	0,000541126 %	216	39.916.800	12
4	3,13E-09 %	4.096	1,31E+12	16
5	8,20E-13 %	100.000	1,22E+17	20
6	1,15E-16 %	2.985.984	2,59E+22	24
7	9,67E-21 %	105.413.504	1,09E+28	28

8	5,23E-25 %	4.294.967.296	8,22E+33	32
9	1,93E-29 %	198.359.290.368	1,03E+40	36
10	5,02E-34 %	10.240.000.000.000	2,04E+46	40

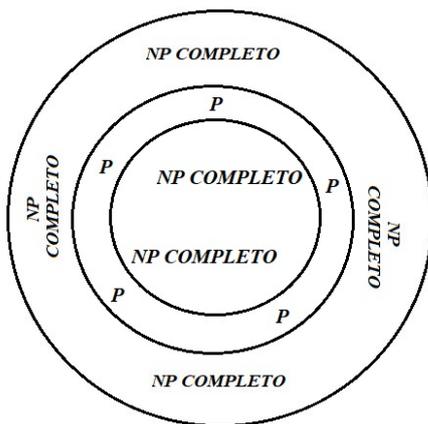
Siguiendo el ejemplo anterior si tardamos por ejemplo 1 segundo en cada movimiento con el algoritmo de Dios para armar el cilindro entonces entre más divisiones tenga el cilindro vamos a necesitar más movimientos para armarlo con el algoritmo de Dios y vamos a tardar más tiempo en armar el cilindro con el algoritmo de Dios en este caso NP no está en P, porque por ejemplo si el cilindro tiene 400 millones de divisiones hay que armarlo con 100 millones de movimientos con el algoritmo de Dios recordemos que es la cuarta parte del número de divisiones totales del cilindro, por eso tardaremos 100 millones de segundos en armarlo porque supongamos que un movimiento es igual a un segundo y entre más divisiones tenga el cilindro más tiempo tardaremos en armarlo con el algoritmo de Dios, recordemos que entre más divisiones tenga el cilindro el problema se vuelve más complejo porque el cilindro tiene más opciones de movimientos, también recordemos que no existe un algoritmo que resuelva el problema en menos pasos que el algoritmo de Dios, ese algoritmo es imposible.

Ahora cambiemos la situación si limitamos el número de movimientos para armar con el algoritmo de Dios el cilindro por ejemplo 20 movimientos y si el cilindro tiene 80 divisiones, de 80 divisiones hacia delante el cilindro se va a armar con 20 movimientos con el algoritmo de Dios, no importa si el cilindro tiene 1 millón, 10 millones, 1 billón e infinidad de divisiones, el cilindro siempre se va a armar con 20 movimientos con el algoritmo de Dios y solo tardaríamos 20 movimientos o 20 segundos en armar el cilindro sin importar que el cilindro se vuelva infinitamente más complejo siempre se armara con 20 movimientos, en este caso NP si está en P. Según lo anterior en la siguiente imagen observamos que los problemas P están dentro de los problemas NP, porque si podemos encontrar en un tiempo polinómico una solución también podemos comprobar en tiempo polinómico una solución, pero hay problemas NP que no están en P, porque se puede comprobar una solución en un tiempo polinómico, pero no se puede encontrar una solución en un tiempo polinómico. Y hay problemas NP que sí están en P, porque se puede

comprobar una solución en un tiempo polinómico y se puede encontrar una solución en un tiempo polinómico.



En la siguiente imagen es lo mismo que la imagen anterior, pero con NP completo.



Respondiendo la pregunta del milenio ¿P es igual a NP?

Si preguntamos a todo el conjunto de problemas de tipo P y de tipo NP es decir ¿todos los problemas P son iguales a todos los problemas NP? La respuesta es que todos los problemas P no son iguales a todos los problemas NP porque habrá problemas de tipo NP que no estén en P. Y habrá problemas de tipo NP que sí estarán en P. Lo mismo ocurre con los problemas NP completo como se observa en las 2 imágenes anteriores.

Pero si decimos individualmente a cada problema de tipo P y de tipo NP si es posible "verificar" rápidamente las soluciones de un problema (es decir, es un problema de tipo NP), ¿eso implica que también es posible "obtener" las respuestas con la misma rapidez? (es decir, es un problema de tipo P)>>>, donde "rápidamente" significa "en tiempo polinómico". En este caso hay 2

respuestas la primera respuesta es que hay casos donde el problema de tipo NP no estará en P y la segunda respuesta es que hay casos donde el problema de tipo NP si estará en P, lo mismo ocurre con NP completo como se observa en las 2 imágenes anteriores.

CONCLUSIONES

Los resultados de esta investigación apoyan la idea de que las complejidades de los problemas aumentan hasta el infinito porque el cilindro entre más divisiones tenga se volverá más complejo, y en algunos casos el algoritmo de Dios resolverá el problema de tipo NP en tiempo polinómico NP si está en P, en otros casos el algoritmo de Dios no resolverá el problema de tipo NP en tiempo polinómico NP no está en P. Y en la vida real habrá problemas donde NP si está en P y problemas donde NP no está en P. No existe un algoritmo que resuelva el problema en menos pasos que el algoritmo de Dios, ese algoritmo es imposible.

BIBLIOGRAFÍAS

A. S. Fraenkel and D. Lichtenstein, Computing a perfect strategy for $n \times n$ chess requires time exponential in n , Proc. 8th Int. Coll. Automata, Languages, and Programming, Springer LNCS 115 (1981) 278-293 and J. Comb. Th. A 31 (1981) 199-214.

Arvind, Vikraman; Kurur, Piyush P. (2006). "Graph isomorphism is in SPP". Information and Computation. 204 (5): 835–852.

Armar el Pyraminx Duo con el algoritmo de Dios.

Le enseñaremos cómo armar el cubo de Rubik y muchos otros rompecabezas en 3 dimensiones con el algoritmo de Dios. Nilson Bolívar (2020) 1-199 <https://docs.google.com/document/d/10EgTFh2v-dWUFIGkP4GEldRg1ajbNIyg/edit?usp=sharing&oid=115495646788988650320&rtpof=true&sd=true>

Aviezri Fraenkel and D. Lichtenstein (1981). "Computing a perfect strategy for $n \times n$ chess requires time exponential in n ". Journal of Combinatorial Theory. Series A. 31 (2): 199–214.

- Colbourn, Charles J. (1984). "The complexity of completing partial Latin squares". Discrete Applied Mathematics. 8 (1): 25–30
- Cook, Stephen (1971). "The complexity of theorem proving procedures". Proceedings of the Third Annual ACM Symposium on Theory of Computing. pp. 151–158
- Elvira Mayordomo. "P versus NP" Archived 16 February 2012 at the Wayback Machine Monografías de la Real Academia de Ciencias de Zaragoza 26: 57–68 (2004).
- Fischer, Michael J.; Rabin, Michael O. (1974). "Super-Exponential Complexity of Presburger Arithmetic". Proceedings of the SIAM-AMS Symposium in Applied Mathematics. 7: 27–41. Archived from the original on 15 September 2006. Retrieved 15 October 2017.
- Fortnow, Lance (2009). "The status of the P versus NP problem" (PDF). Communications of the ACM. 52 (9): 78–86. CiteSeerX 10.1.1.156.767. doi:10.1145/1562164.1562186. S2CID 5969255. Archived from the original (PDF) on 24 February 2011. Retrieved 26 January 2010.
- Hartmanis, Juris. "Gödel, von Neumann, and the P = NP problem" (PDF). Bulletin of the European Association for Theoretical Computer Science. 38: 101–107.
- I. Holyer (1981). "The NP-completeness of some edge-partition problems". SIAM J. Comput. 10 (4): 713–717.
- Johnson, David S. (1987). "The NP-completeness column: An ongoing guide (edition 19)". Journal of Algorithms. 8 (2): 285–303.
- Jump up to:^a ^b Cook, Stephen (April 2000). "The P versus NP Problem" (PDF). Clay Mathematics Institute. Archived (PDF) from the original on 16 December 2013. Retrieved 18 October 2006.
- Kawarabayashi, Ken-ichi; Kobayashi, Yusuke; Reed, Bruce (2012). "The disjoint paths problem in quadratic time". Journal of Combinatorial Theory. Series B. 102 (2): 424–435.
- Ladner, R.E. (1975). "On the structure of polynomial time reducibility". Journal of the ACM. 22: 151–171 See Corollary 1.1.

P. T. Ledner "On the structure of polynomial time reducibility," Journal ACM, 22, pp. 151–171,

1975, Corollary 1.1

Sipser, Michael: Introduction to the Theory of Computation, Second Edition, International Edition, page 270. Thomson Course Technology, 2006. Definition 7.19 and Theorem

7.20

Valiant, Leslie G. (1979). "The complexity of enumeration and reliability problems". SIAM

Journal on Computing, 8 (3): 410–421.

William I. Gasarch (junio de 2002). «The P=?NP poll.». SIGACT News 33 (2): 34-47