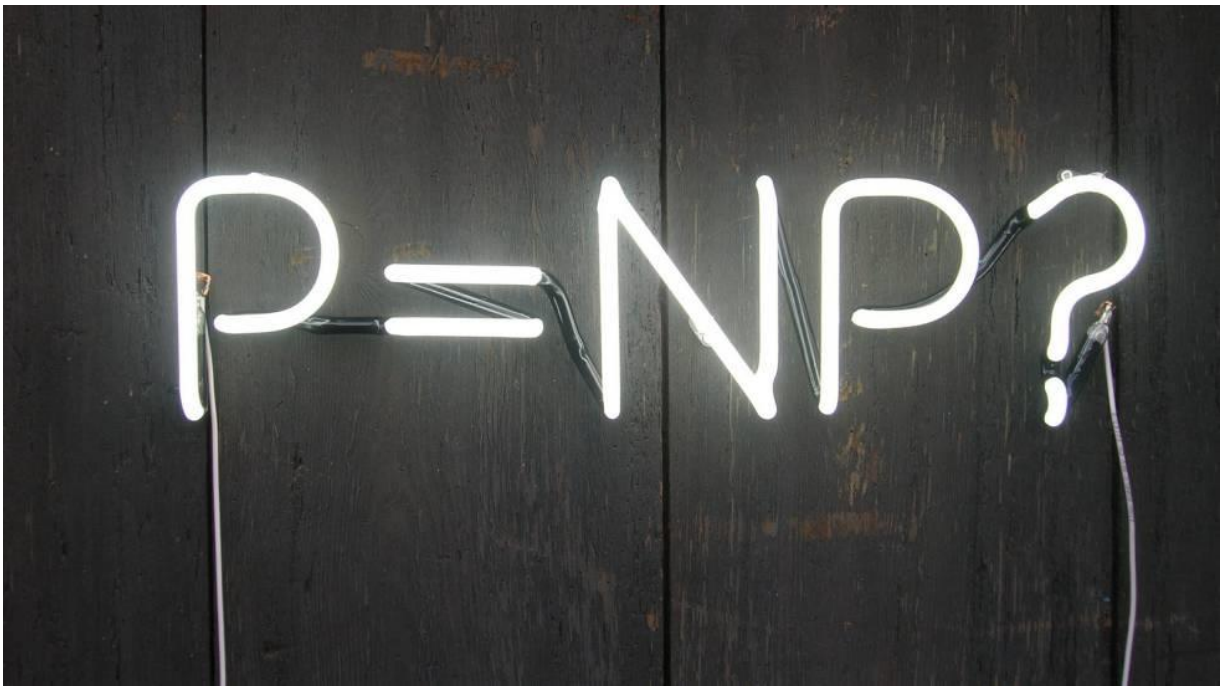
El problema que los informáticos no han podido resolver en 45 años

La pregunta "¿P=NP?" trae de cabeza a los programadores desde 1971



Muchas personas se preguntan qué diantres se esconde tras la pregunta ¿P=NP?, y por qué parece ser tan importante.

¿Qué diantres se esconde tras la pregunta '¿P=NP?', y por qué parece ser tan importante para los informáticos? Se trata de una pregunta, todavía sin respuesta desde 1971, año en que fue planteada, que trae de cabeza a los informáticos. Si fuera P≠NP, las cosas seguirían más o menos igual, pero si fuera P=NP, entonces muchas cosas cambiarían y no necesariamente para mejor. Veamos por qué.

Gran parte de la ciudadanía tiende a pensar que los computadores pueden resolver todos los problemas, y que los que no pueden resolver hoy, podrán hacerlo mañana, porque su potencia de cálculo crece continuamente. Los informáticos sabemos en cambio, que una infinidad de problemas de cómputo no tendrán solución nunca (los llamamos problemas *indecidibles*), y que para otros problemas existen algoritmos

que los resuelven, pero empleando para ello tanto tiempo de cálculo, que a efectos prácticos es como si fueran irresolubles (los llamamos problemas *intratables*). Los problemas que resuelven los computadores en un tiempo razonable los llamamos *polinomiales*, y todos ellos se agrupan en la llamada clase P. Se dicen así porque su tiempo de cómputo está descrito por un polinomio en el tamaño de los datos. Por ejemplo, el problema de multiplicar dos matrices de *n* filas y *n* columnas se puede resolver utilizando menos de n^3 multiplicaciones. Ninguno de los problemas intratables está en la clase P.

MÁS INFORMACIÓN

-  El joven que se enfrenta al problema matemático del millón de dólares
-  Lo imposible

Hay otra clase de problemas, a la que llamamos NP, cuya definición está hecha de tal manera que incluye todos los problemas de la clase P, pero también otros muchos que se comportan de un modo intrigante. Uno de esos problemas es el llamado *problema del viajante de comercio:* dado un mapa de carreteras, consiste en encontrar el camino más corto para visitar *n* ciudades una sola vez y volver al punto de origen. Para estos nuevos problemas de la clase NP, los mejores algoritmos que se conocen tienen un coste similar al de los problemas intratables, pero nadie ha podido demostrar que no existan algoritmos polinomiales para ellos. Tampoco nadie ha demostrado que sean intratables. Están, por decirlo así, en una especie de limbo informático: no se sabe si son polinomiales, o si son intratables. La teoría desarrollada en estos años ha llegado sin embargo a alguna conclusión útil: ha definido una subclase de la clase NP, la subclase de los problemas NP-completos, en la cual se agrupan los problemas más costosos de la clase NP, de tal forma que, si

para uno cualquiera de dichos problemas se encontrara un algoritmo polinomial, entonces todos ellos se resolverían en tiempo polinomial y además la clase NP colapsaría a P, es decir tendríamos la igualdad P=NP. Más aún, si se demostrara que uno solo de los problemas NP-completos es intratable, entonces todos ellos lo serían y tendríamos la desigualdad P≠NP.

La criptografía actual depende de un problema de la clase NP, el de la descomposición en factores de un número, para el que no tenemos algoritmos eficientes

Las consecuencias de esto último no serían muchas: simplemente dejaríamos de buscar algoritmos polinomiales para una serie de problemas interesantes, porque sabríamos con seguridad que tales algoritmos no existen. En cambio, si fuera P=NP, habríamos encontrado algoritmos polinomiales para todos esos problemas. La parte buena de ello es que podríamos resolver, en tiempos muy cortos, *problemas del viajante* con miles de ciudades y otros cientos de problemas útiles para los que hoy tenemos algoritmos muy costosos, y eso sería beneficioso para la industria, las comunicaciones y el desarrollo en general. La parte mala es que las claves criptográficas se descifrarían con gran facilidad, y muchas cuentas bancarias y comunicaciones cifradas quedarían expuestas a los amigos de lo ajeno.

En efecto, la criptografía actual depende de un problema de la clase NP, el de la descomposición en factores de un número, para el que no tenemos algoritmos eficientes. El más eficiente de todos tardó 18 meses en descomponer en factores un número de 200 cifras decimales, que son los que se usan habitualmente en criptografía. La seguridad de las claves descansa precisamente en esta dificultad, hoy por hoy insalvable. Si fuera P=NP, entonces la descomposición en factores pasaría a ser un problema polinomial y se podría resolver eficientemente. La criptografía tendría que ingeniárselas para basar la seguridad de sus claves en la resolución de algún problema realmente intratable, porque los de la clase NP habrían pasado todos ellos a la categoría de eficientes.

Ricardo Peña Marí es catedrático de la Universidad Complutense de Madrid.