**Problemas NP-Hard**

¿Qué significa que un problema sea **NP-Hard**? A raíz del [paper](http://arxiv.org/abs/1403.1911" \o "Candy Crush is NP-Hard" \t "_blank) que demostraba que el juego Candy Crush es un problema de este tipo, el término se ha puesto muy de moda, pero el concepto es algo complejo. Vamos a ver qué significa.

En las Ciencias de la Computación existe una rama que trata de clasificar los problemas de decisión en varias familias en función de la dificultad que conlleva su resolución. Esta rama estudia la **Teoría de la Complejidad Computacional**. El objetivo final es determinar qué clases de problemas se pueden resolver con ciertos recursos y qué cantidad de estos necesita cada uno desde un punto de vista formal.

Esta teoría estudia los **problemas computacionales**, entendidos estos como una pregunta, cuya respuesta depende de unas determinadas variables libres con valores sin especificar y cuya solución tiene unas características que ha de cumplir para ser válida que también están unívocamente especificadas. En resumen: un problema queda definido por “a qué respondemos” y “cómo tiene que ser la respuesta”.

Dentro de estos problemas nos encontramos con varias tipologías. Grosso modo, existen los problemas de **decisión**, los problemas de **búsqueda** y los problemas de **optimización**.

Para poder definir la cantidad de recursos que necesita cada problema para ser resuelto, se define lo que se llaman **clases de complejidad**, que no son más que conjuntos de problemas que requieren una cantidad de recursos parecida para ser resueltos. Por tanto, podemos decir que un problema pertenece a una clase de complejidad, dado que ésta puede ser considerada, matemáticamente, como un conjunto.

Aunque se podría aplicar a cualquiera de los tipos de problemas, se suelen utilizar los **problemas de decisión** cuando hablamos de esta clasificación, por ser a los que más directamente se puede aplicar la teoría. Un problema de decisión es un problema cuya solución únicamente puede ser “sí” o “no”. Ejemplos de problemas de este tipo serían, por ejemplo, “¿es x un número par?”, “¿es x un número primo?” o “¿es coloreable con n colores el grafo completo de x vértices?”.

Otra clasificación que se ha de hacer trata acerca la **decibilidad** de los problemas de decisión. Si existe un **algoritmo** que para cada posible entrada nos de la solución, en forma de sí o no como ya se ha dicho, el problema será **decidible**, en caso contrario no lo será y se denominará **indecidible**. La teoría, en este caso, estudia los problemas decidibles, puesto que si el problema no lo es quiere decir que existe al menos un conjunto de valores de las variables libres para el que no se puede resolver, en cuyo caso no tiene sentido plantearse su complejidad que, por decir algo, pasaría a ser “infinita”.

Un **algoritmo** no es más que una sucesión finita y ordenada de pasos unívocos que nos proveen la solución al problema. Se debe garantizar que, si se siguen esos pasos, se llega a la solución y también que el número de pasos a realizar es finito. Si no cumple alguna de estas dos características, el algoritmo es inválido.

En base a esto, existen varias clases de complejidad para problemas decidibles. Una de ellas es la clase **NP (Nondeterministic Polynomial)**, que corresponde a los problemas que se pueden resolver en**tiempo polinómico** por una máquina de Turing **no determinista**. En otro lugar se encontraría la clase**P (Polynomial)**, que representa los problemas que pueden ser resueltos en tiempo polinómico por una máquina de Turing**determinista y secuencial**. Uno de los problemas más apasionantes de las Ciencias de la Computación actuales se refiere a la pregunta **¿es P = NP?** Aunque la mayoría de la academia parece pensar que esta afirmación no se cumple, dicha proposición no está demostrada ni refutada formalmente. El instituto Clay ha fijado un premio de un millón de dólares para el que resuelva el problema.

Los problemas **NP-Hard**, también llamados **NP-Complejos** serían el conjunto de problemas que son al menos tan complejos como los NP, aunque no tienen por qué pertenecer a la clase NP. Formalmente, es la clase de todos los problemas H para los cuales cualquier problema de NP se puede transformar en H mediante una transformación polinómica. Básicamente son computacionalmente intratables con los medios actuales.