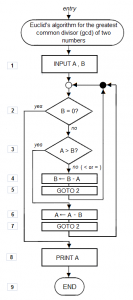
<https://victoryepes.blogs.upv.es/2012/05/05/algoritmo/>

Qué es un algoritmo?

[](http://victoryepes.blogs.upv.es/files/2012/05/Euclid_flowchart_1.png)Algoritmo de Euclides

Un [*algoritmo*](http://es.wikipedia.org/wiki/Algoritmo) es un conjunto prescrito de reglas o instrucciones bien definidas para la resolución de un problema. En general, se trata de encontrar el método más “eficiente”, no siendo baladí el modo de medir dicha eficiencia. Para resolver esta circunstancia, en la década de los 70 numerosos científicos se interesaron por la complejidad computacional de los problemas y los algoritmos. En muchos casos se asimila el [*rendimiento algorítmico*](http://es.wikipedia.org/wiki/Ordenamiento_de_burbuja) a la medida del tiempo medio de ejecución empleado por un procedimiento para completar su operación con un conjunto de datos. Además, es posible relacionar el esfuerzo de cálculo con la dimensión del problema a resolver.

Un algoritmo muestra una [*complejidad polinómica*](http://www.lab.dit.upm.es/~lprg/material/apuntes/o/index.html) si necesita un tiempo *O(nk)*, donde *n* muestra la dimensión de entrada y *k* es una constante independiente de *n*. Si la función que denota la complejidad no está acotada por un polinomio, el algoritmo presenta una *complejidad en tiempo exponencial*.

 Un [*problema de decisión*](http://es.wikipedia.org/wiki/Problema_de_decisi%C3%B3n) es aquel que puede ser contestado con una afirmación o una negación. Llamemos P a la clase de problemas de decisión que pueden ser resueltos en tiempo cálculo que crece de forma polinomial ante incrementos lineales del número de elementos que intervienen, y NP aquellos solubles en un tiempo polinomial indeterminado, es decir, que se puede resolver en tiempo polinomial con una [máquina Turing](http://es.wikipedia.org/wiki/M%C3%A1quina_de_Turing) no determinística (ordenador). Un ordenador no determinístico puede ser contemplado como un [autómata](http://es.wikipedia.org/wiki/Aut%C3%B3mata_programable) capaz de ejecutar un número ilimitado (pero finito) de ejecuciones en paralelo. Sólo los problemas en P son resolubles eficientemente mediante algoritmos, no conociéndose un procedimiento polinomial de resolución para los NP, siendo obvio que P pertenezca NP. Si lo contrario también ocurriera, P pertenecería a NP, querría decir que para la mayoría de los problemas de interés existen algoritmos eficientes que los resolvieran. Sin embargo, no se conoce la forma de demostrar que la igualdad P=NP sea cierta, ni tampoco que haya problemas en NP que no estén en P, es decir, la existencia de algún problema en NP que no se pueda resolver en tiempo polinómico (ver Díaz *et al*., 1996).

Un problema *X* se dice que es NP-*completo* (NPC) si cualquier problema en NP se puede transformar en *X* en tiempo polinomial. En este sentido, los NPC son una clase de problemas en NP muy difíciles. Si un solo problema en NPC se resolviera en tiempo polinomial, entonces todos los problemas NP también lo harían, lo cual no está demostrado a fecha de hoy. Sin embargo, no es necesario demostrar que un problema pertenece a NP para ofrecer evidencias de que es imposible resolverlo eficientemente. Sea *Y* un problema de decisión que no se conoce si es NP. Si un problema en NP-completo puede transformarse en *Y*, entonces *Y* no puede resolverse en tiempo polinomial (salvo que se demuestre que P=NP). Este problema *Y* sería como mínimo tan difícil como los NPC, llamándose NP-*hard* (NPH). Es decir, pueden existir problemas NPH que no sean NPC. A efectos prácticos únicamente nos interesa confirmar la NP-dificultad de un problema.

En la vida real existen numerosos [problemas prácticos](http://victoryepes.blogs.upv.es/2012/04/27/complicados-problemas-distribucion-fisica/) para los cuales se desconocen algoritmos eficientes (Yepes, 2002), pero cuya dificultad intrínseca no ha conseguido demostrar nadie. Es posible que existan realmente algoritmos eficientes, aunque también puede ocurrir que estos problemas sean intrínsecamente difíciles; no obstante se carecen de las técnicas necesarias para demostrarlo. La importancia práctica de estos problemas ha asegurado que cada uno de ellos por separado haya sido objeto de esfuerzos sostenidos para hallar un método de solución eficiente. Por este motivo, se cree que no existen tales algoritmos. Como nadie, de momento, ha encontrado algoritmos eficientes para los problemas NP-completos, en cuanto se demuestra que un problema pertenece a esta clase, muchos investigadores tienden a pensar que no merece la pena buscar algoritmos eficientes para ellos. Lamentablemente, muchos de los problemas importantes que aparecen en Investigación Operativa son NP-completos. En Garey y Johnson (1979) se encuentra una visión más completa de la complejidad computacional.

**REFERENCIAS**

DÍAZ, A.; GLOVER, F.; GHAZIRI, H.M.; GONZÁLEZ, J.L.; LAGUNA, M.; MOSCATO, P.; TSENG, F.T. (1996). *Optimización Heurística y Redes Neuronales en Dirección de Operaciones e Ingeniería.* Paraninfo, Madrid. 235 pp.

GAREY, M.R.; JOHNSON, D.S. (1979). *Computers and Intractability – A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company.

YEPES, V. (2002). *Optimización heurística económica aplicada a las redes de transporte del tipo VRPTW.*Tesis Doctoral. Escuela Técnica Superior de Ingenieros de Caminos, Canales y Puertos. Universitat Politècnica de València. 352 pp. ISBN: 0-493-91360-2. [(pdf)](http://personales.upv.es/vyepesp/ResumenTesis.PDF)