

## Analisis

A menudo surge una pregunta interesante. Cuando dos programas resuelven el mismo problema pero se ven diferentes, ¿es un programa mejor que el otro?

Con el fin de responder esta pregunta, tenemos que recordar que hay una diferencia importante entre un programa y el algoritmo subyacente que el programa está representando. Como dijimos en el Capítulo 1, un algoritmo es una lista genérica, paso a paso, de instrucciones para resolver un problema. Es un método para resolver cualquier caso del problema de tal manera que dada una entrada particular, el algoritmo produzca el resultado deseado. Un programa, por otro lado, es un algoritmo que ha sido codificado en algún lenguaje de programación. Pueden existir muchos programas para el mismo algoritmo, dependiendo del programador y del lenguaje de programación que se esté utilizando.

## Complejidad de espacio y ejecución

En un sentido amplio, dado un problema y un dispositivo donde resolverlo, es necesario proporcionar un método preciso que lo resuelva, adecuado al dispositivo. A tal método lo denominamos algoritmo. En el presente texto nos vamos a centrar en dos aspectos muy importantes de los algoritmos, como son su diseño y el estudio de su eficiencia. El primero se refiere a la búsqueda de métodos o procedimientos, secuencias finitas de instrucciones adecuadas al dispositivo que disponemos, que permitan resolver el problema. Por otra parte, el segundo nos permite medir de alguna forma el coste (en tiempo y recursos) que consume un algoritmo para encontrar la solución y nos ofrece la posibilidad de comparar distintos algoritmos que resuelven un mismo problema.

Una vez dispongamos de un algoritmo que funciona correctamente, es necesario definir criterios para medir su rendimiento o comportamiento. Estos criterios se centran principalmente en su simplicidad y en el uso eficiente de los recursos. A menudo se piensa que un algoritmo sencillo no es muy eficiente.

Sin embargo, la sencillez es una característica muy interesante a la hora de diseñar un algoritmo, pues facilita su verificación, el estudio de su eficiencia y su mantenimiento. De ahí que muchas veces prime la simplicidad y legibilidad del código frente a alternativas más crípticas y eficientes del algoritmo.

Respecto al uso eficiente de los recursos, éste suele medirse en función de dos parámetros: el espacio, es decir, memoria que utiliza, y el tiempo, lo que tarda en ejecutarse. Ambos representan los costes que supone encontrar la solución al problema planteado mediante un algoritmo. Dichos parámetros van a servir además para comparar algoritmos entre sí, permitiendo determinar el más adecuado de entre varios que solucionan un mismo problema. En este capítulo nos centraremos solamente en la eficiencia temporal. El tiempo de ejecución de un algoritmo va a depender de diversos factores como son: los datos de entrada que le suministremos, la calidad del código generado por el compilador para crear el programa objeto, la naturaleza y rapidez de las instrucciones máquina del procesador concreto que ejecute el programa, y la complejidad intrínseca del algoritmo.

<http://www.lcc.uma.es/~av/Libro/CAP1.pdf>