

Algoritmia/Introducción

Introducción

Este libro versa sobre el análisis y el diseño de algoritmos. Los métodos algorítmicos incluidos son, entre otros, divide y vencerás, vuelta a atrás, programación dinámica, algoritmos voraces y algoritmos probabilísticos.

Cualquier problema soluble puede por lo general resolverse a través de las siguientes formas:

- El camino obvio
- El camino metódico
- El camino inteligente
- El camino milagroso

En el primer nivel, la solución obvia puede consistir en una búsqueda exhaustiva de la respuesta al problema. De forma intuitiva, la solución obvia es aquella que aparece fácilmente con unos conocimientos básicos de programación.

El camino metódico es el que trata de ilustrar este libro. Tras comprender el material presentado aquí, deberían poderse resolver, siguiendo la metodología enseñada, los problemas de una forma más efectiva y eficiente.

El tercer nivel, el camino inteligente, requiere una mayor comprensión de los elementos implicados en el problema y sus propiedades (por ejemplo, ciertos algoritmos numéricos pueden basarse en propiedades matemáticas poco obvias). Un algoritmo de este tipo puede ser complicado de entender por no resultar obvio que es correcto, o puede resultar difícil comprender que realmente es más eficiente de lo que parece.

Finalmente, el cuarto nivel, el camino milagroso, está reservado para casos verdaderamente extraños en los que la solución es realmente poco intuitiva.

Obviamente, la diferenciación entre estos niveles es relativa, y puede ser que algún algoritmo algo más complejo de lo habitual se presente en el actual libro además de las metodologías básicas.

Prerrequisitos

Para comprender el material expuesto en este libro se necesitan firmes conocimientos de programación para poder entender el pseudocódigo y traducirlo en soluciones funcionales. Asimismo son necesarios conocimientos de estructuras de datos, incluyendo arrays, pilas, colas, listas enlazadas, árboles, montículos y colas de prioridad, conjuntos y grafos.

De forma adicional, son recomendables otros conocimientos como algoritmos de búsqueda binaria, de ordenación, de búsqueda primero en profundidad y primero en anchura. En el caso de no estar familiarizado con estos prerrequisitos, es aconsejable revisar previamente las secciones dedicadas a los mismos en el libro Fundamentos de programación.

Introducción a los métodos algorítmicos

Una muestra de las técnicas algorítmicas vistas en este libro es la siguiente:

- **Divide y vencerás:** muchos problemas, particularmente aquellos cuya entrada consiste en un array, pueden solucionarse a través de sucesivas reducciones del problema en subproblemas menores y la posterior combinación de dichas soluciones en un único resultados. Los algoritmos de ordenación quicksort y mergesort son algunos ejemplos en los cuales esta técnica se aplica de forma sencilla.
- **Vuelta atrás:** casi cualquier problema puede representarse a través de un algoritmo de vuelta atrás. En esta técnica, se consideran todas las posibles elecciones para solucionar un problema y, recursivamente, se resuelven los subproblemas suponiendo una determinada elección. El conjunto de sucesivas llamadas genera un árbol donde

cada rama representa una determinada secuencia de elecciones. Así, si existe solución al problema, esta se encontrará finalmente. En general, la vuelta atrás es un método de fuerza bruta muy ineficiente, pero es posible aplicar optimizaciones que reduzcan el tamaño del árbol y, consecuentemente, mejoren la eficiencia del algoritmo.

- **Programación dinámica:** se trata de una optimización para algoritmos de vuelta atrás. Cuando un determinado subproblema necesita tratarse varias veces (por ejemplo, ramas repetidas en el árbol generado), es posible ahorrar tiempo solucionando una sola vez el subproblema y almacenando la solución en una tabla.
 - **Algoritmos voraces:** este tipo de algoritmos resultan de utilidad cuando se conoce suficiente información acerca de las posibles elecciones, de tal forma que "el caso mejor" pueda determinarse sin necesidad de considerar todas las opciones. Típicamente, los algoritmos voraces no son difíciles de escribir, pero sí de demostrar su corrección.
 - **Algoritmos de escalada:** la idea básica consiste en comenzar con una mala solución a un determinado problema y, repetidamente, aplicar optimizaciones a la misma hasta que esta sea óptima o satisfaga algún otro requisito.
 - **Algoritmos probabilísticos:** hacen uso del azar para alcanzar una solución, basándose en un análisis estadístico que indique que, en un número lo suficientemente alto de ocasiones, se alcanzará una solución óptima al problema.
-

Fuentes y contribuyentes del artículo

Algoritmia/Introducción *Fuente:* <http://es.wikibooks.org/w/index.php?oldid=157510> *Contribuyentes:* Gothmog, Kuartas, 4 ediciones anónimas

Licencia

Creative Commons Attribution-Share Alike 3.0 Unported
[//creativecommons.org/licenses/by-sa/3.0/](https://creativecommons.org/licenses/by-sa/3.0/)
