

Análisis de Algoritmos

Un algoritmo es un método para resolver un problema, “un conjunto de pasos que, ejecutados de la manera correcta, permiten obtener un resultado (en un tiempo acotado)”. Pueden existir varios algoritmos para resolver un mismo problema. Cuando se estudian los algoritmos es importante analizar tanto su diseño como su eficiencia. Los algoritmos que resuelven un determinado problema tienen diferentes ventajas y desventajas, relacionadas principalmente con el tiempo de ejecución y el espacio en memoria.

Complejidad en espacio y ejecución

El tiempo de ejecución concierne a la cantidad de tiempo que tarda un algoritmo en encontrar soluciones y el espacio en memoria se relaciona con la cantidad de almacenamiento en memoria requerido por el algoritmo. Normalmente, el tiempo de ejecución es considerado el criterio más importante para seleccionar un algoritmo. Por lo tanto, cuando se realice el análisis de un algoritmo, el término eficiencia estará relacionado con que tan rápido se ejecute un algoritmo y el que determinara su superioridad. Para un algoritmo, una instancia es cada uno de los ejemplares que pertenecen a su dominio de definición.

El tiempo de ejecución está relacionado con el tiempo que le toma a un algoritmo encontrar la solución de un determinado problema y es el criterio que normalmente decide que algoritmo es el mejor. Para seleccionar un algoritmo se usan dos enfoques: el enfoque empírico o el enfoque teórico.

El enfoque empírico consiste en implementar cada algoritmo en un lenguaje de programación y realizar pruebas con numerosas instancias del problema, el algoritmo que obtenga los mejores resultados será el seleccionado. La desventaja del enfoque empírico es que si deseamos medir la eficiencia de un algoritmo en función del tiempo que se requiere para obtener una solución, por ejemplo, en segundos, se presenta el problema que existe una gran diversidad de computadoras que toman diferentes cantidades de tiempo al realizar las mismas operaciones, lo que hace imposible referirse a una medida de tiempo general. Además, obliga a reportar las configuraciones físicas de la computadora donde se ejecuta el algoritmo.

Por otro lado, el enfoque teórico determinar matemáticamente la cantidad de recursos requeridos considerando el tiempo de ejecución como la cantidad de operaciones elementales que realiza un algoritmo en función del tamaño de la instancias de entrada. Una operación elemental se refiere a una operación que toma una cantidad de tiempo constante como lo son: la suma, la multiplicación, la división, entre otras. El tamaño de la entrada del algoritmo es una propiedad del propio algoritmo independiente de la computadora en la que se ejecuta.

Eficiencia de algoritmos

El análisis de algoritmos pretende descubrir si estos son o no eficaces. Establece además una comparación entre los mismos con el fin de saber cuál es el más eficiente, aunque cada uno de los algoritmos de estudio sirva para resolver el mismo problema. Es responsabilidad del programador utilizar los recursos de la computadora de la manera más eficiente que se pueda. Hay diversos métodos y criterios para estudiar la eficiencia de los algoritmos. Por lo general los aspectos a tomar en cuenta para estudiar la eficiencia de un algoritmo son el tiempo que se emplea en resolver el problema y la cantidad de recursos de memoria que ocupa. Para saber qué tan eficiente es un algoritmo hacemos las preguntas:

- ☐ ¿Cuánto tiempo ocupa?
- ☐ ¿Cuánta memoria ocupa?

El tiempo de ejecución de un algoritmo depende de los datos de entrada, de la implementación del programa, del procesador y finalmente de la complejidad del algoritmo. Sin embargo, decimos que el tiempo que requiere un algoritmo para resolver un problema está en función del tamaño n del conjunto de datos para procesar: $T(n)$.

Comparación entre velocidad de ejecución y eficiencia del algoritmo

Suponga que para un cierto problema se han desarrollado los algoritmos A y B, cuyo tiempo de ejecución está dado por las siguientes funciones:

$$T_A(n) = 100 \cdot n$$

$$T_B(n) = 2 \cdot n^2$$

¿Cuál es más eficiente?

Si $n < 100$ B, es más eficiente

Si $n \geq 50$ A es mucho más eficiente.

	n=10	n=50	n=100	n=1000
$T_A(n) = 100 \cdot n$	1000	5000	10000	100 000
$T_B(n) = 2 \cdot n^2$	2000	5000	20000	500 000
Factor B/A	0.2	1	2	5

Costes en tiempo y espacio

La característica básica que debe tener un algoritmo es que sea **correcto**, es decir, que produzca el resultado deseado en tiempo finito. Adicionalmente puede interesarnos que sea claro, que este bien estructurado, que sea fácil de usar, que sea fácil de implementar y que sea **eficiente**. Entendemos por eficiencia de un algoritmo la cantidad de recursos de cómputo que requiere; es decir, cuál es su tiempo de ejecución y que cantidad de memoria utiliza.

A la cantidad de tiempo que requiere la ejecución de un cierto algoritmo se le suele llamar **coste en tiempo** mientras que a la cantidad de memoria que requiere se le suele llamar **coste en espacio**. Es evidente que conviene buscar algoritmos correctos que mantengan tan bajo como sea posible el consumo de recursos que hacen del sistema, es decir, que sean lo más eficientes posible. Cabe hacer notar que el concepto de **eficiencia** de un algoritmo es un **concepto relativo**, esto quiere decir que ante dos algoritmos correctos que resuelven el mismo problema, uno es más **eficiente** que otro si consume **menos recursos**. Por tanto, podemos observar que el concepto de eficiencia y en consecuencia el **concepto de coste** nos permitirá comparar distintos algoritmos entre ellos.

Bibliografía

Amalia Duch. (2007). Análisis de Algoritmos,. Universidad Politécnica de Cataluña. Sitio web: <https://www.cs.upc.edu/~duch/home/duch/analisis.pdf>

Jorge Luis Zapotecatl Lopez., (2014). Análisis de Algoritmos. Instituto Nacional de Astrofísica, Óptica y Electrónica. Sitio web: <http://www.pensamientocomputacional.org/Files/analisis.pdf>

Alma Leticia Palacios Guerrero, (2012). Algoritmos y Estructuras de Datos, Universidad Autónoma de Baja California. Sitio web: <http://fcqi.tij.uabc.mx/usuarios/palacios/UNIDAD%201%20%20Introduccion%20a%20los%20algoritmos%20y%20Estructuras%20de%20Datos.pdf>

