

Resumen Analítico de Eficiencia

Raramente existe un único algoritmo para resolver un problema determinado.

Cuando se comparan dos algoritmos diferentes que resuelven el mismo problema, normalmente se encontrará que un algoritmo es un orden de magnitud más eficiente que el otro. En este sentido, lo importante es que el programador sea capaz de reconocer y elegir el algoritmo más eficiente.

Entonces, ¿qué es eficiencia? La eficiencia de un algoritmo es la propiedad mediante la cual un algoritmo debe alcanzar la solución al problema en el tiempo más corto posible o utilizando la cantidad más pequeña posible de recursos físicos y que sea compatible con su exactitud o corrección (es decir que sus resultados sean correctos).

Un programador debe saber identificar el algoritmo más eficiente dentro del conjunto de aquellos que resuelven con exactitud un problema dado (el más eficiente de todos).

¿Cómo medir la eficiencia de un algoritmo?

Existen diferentes métodos con los que se trata de medir la eficiencia de los algoritmos; entre ellos, los que se basan en el número de operaciones que debe efectuar un algoritmo para realizar una tarea; otros métodos se centran en tratar de medir el tiempo que se emplea en llevar a cabo una determinada tarea, ya que lo importante para el usuario final es que ésta, se efectúe de forma correcta y en el menor tiempo posible.

Sin embargo, estos métodos presentan varias dificultades, ya que cuando se trata de generalizar la medida establecida, ésta depende de factores como la máquina en la que se efectuó, el ambiente del procesamiento y el tamaño de la muestra, entre otros factores.

El estudio de la eficiencia de los algoritmos se centra, fundamentalmente, en el análisis de la ejecución de bucles en función del número de instrucciones que contiene.

La eficiencia la podemos medir en Tiempos de ejecución (Complejidad de tiempo del algoritmo).

¿Qué es la complejidad de un Algoritmo?

Es la medida de los recursos (tiempo y espacio), que un algoritmo requiere para su funcionamiento. Esta se expresa en función del tamaño del problema que se quiere resolver.

Si el recurso es Espacio:

- La complejidad es la cantidad de memoria que se requiere para su ejecución esto tiene que ver con las estructuras de datos utilizadas para su implementación.

Si el recurso es Tiempo:

- La complejidad se asocia a la cantidad de tiempo que el algoritmo toma para ejecutar las operaciones.

Cuando el tamaño de la entrada crece (conjunto de datos), entonces nos interesa medir el Tiempo de Ejecución. La función para medir esta complejidad se denota como $T(n)$.

Para medir esta función se realiza lo siguiente:

- Se ejecuta el algoritmo
- Se calcula contando las sentencias del algoritmo y multiplicándolas por el tiempo requerido.

La complejidad del algoritmo se describe por medio de una función, que se estima a partir del tiempo de ejecución y de espacio de memoria.

La complejidad del algoritmo se denota del siguiente modo: $O()$

El conjunto de funciones se denominan O , y existe una gran variedad de ellas:

$O(1)$	Complejidad Constante
$O(n)$	Complejidad Lineal
$O(n^2)$	Complejidad Cuadrático
$O(n^3)$	Complejidad Cúbico
$O(n^a)$	Complejidad Polinómica ($a > 3$)
$O(\log(n))$	Complejidad Logarítmica
$O(n \log(n))$	Complejidad Cuasi - lineal
$O(2^n)$	Complejidad Exponencial