

La Eficiencia de los Algoritmos

Daniel Alconchel Vázquez

25 de marzo de 2021

Índice

1. Planteamiento general	3
2. Tiempo de ejecución. Notaciones para la eficiencia de los algoritmos	3
2.1. Notación Asintótica	4

1. Planteamiento general

Una definición formal y general de **algoritmo** es una secuencia finita y ordenada de pasos, extensos de ambigüedad, que dará como resultado que se realice la tarea para la que se ha diseñado con recursos limitados y en tiempo finito.

Definición 1.1. Definición de Bazaraa, Sheraly y Shetty:

Un algoritmo para resolver un problema es un proceso iterativo que genera una sucesión de puntos, conforme a un conjunto dado de instrucciones y con un criterio de parada.

Un algoritmo tiene cinco características primordiales:

- **Finitud:** ha de terminar después de un tiempo acotado.
- **Especificidad:** cada etapa debe estar precisamente definida.
- **Input:** un algoritmo tiene cero o más inputs.
- **Output:** uno o más outputs.
- **Efectividad:** todas las operaciones deben ser tan básicas como para que se puedan realizar en un periodo finito de tiempo.

2. Tiempo de ejecución. Notaciones para la eficiencia de los algoritmos

Definición 2.1. Principio de Invariancia:

Dos implementaciones diferentes de un mismo algoritmo no difieren en eficiencia más que, a lo sumo, en una constante multiplicativa.

Parece oportuno referirnos a la eficiencia teórica de un algoritmo en términos de tiempo. Algo que conocemos de antemano es el denominado **Tiempo de Ejecución** de un programa, que depende de:

- Input del programa (Depende de su tamaño).
- La calidad del código que genera el compilador.
- Naturaleza y velocidad de las instrucciones máquina.
- Complejidad en tiempo en tiempo del algoritmo.

Usamos la notación $T(n)$ para referirnos al tiempo de ejecución de un programa para un input de tamaño n .

Diremos que un algoritmo consume un tiempo de orden $t(n)$, si existe una constante positiva, c , y una implementación del algoritmo capaz de resolver cualquier caso del problema en un tiempo acotado superiormente por $ct(n)$ segundos, donde n es el tamaño del caso considerado.

Además, no habrá unidad para expresar el tiempo de ejecución de un algoritmo. Usaremos una constante (oculta) para acumular dichos factores relativos.

2.1. Notación Asintótica

Para comparar los algoritmos empleando los tiempos de ejecución y las constantes ocultas se emplea la denominada **notación asintótica**. Esta notación sirve para capturar las conductas de funciones para valores altos de x .

2.2. Definición formal para O

Sean f y g funciones definidas de $\mathbb{N} \rightarrow \mathbb{R}^+$. Si dice que f es de orden g , $O(g(n))$, si existen dos constantes positivas C y k tales que

$$\forall n \geq k, f(n) \leq C \cdot g(n)$$

Para probar que no pueden existir constantes C , k tales que pasado k , siempre se verifique la in-ecuación, podemos usar límites. Por ejemplo:

$$x^4 \neq O(3x^3 + 5x^2 - 9) \quad \lim_{x \rightarrow \infty}$$