

# **Análisis de algoritmos complejidad en espacio y tiempo, y ejecución y eficiencia de algoritmos**

## **Complejidad de algoritmos**

Un algoritmo es un procedimiento no necesariamente secuencial, que toma un conjunto de valores de entrada y los transforma para producir un conjunto de valores de salida. El análisis de un algoritmo consiste en predecir la cantidad de recurso (tiempo, memoria, comunicación) que un algoritmo requerirá para cualquier entrada.

## **Complejidad en el tiempo**

### **-Tiempo de ejecución**

Cuando se resuelve un problema con frecuencia hay necesidad de elegir entre varios algoritmos. ¿Cómo se debe elegir? Hay dos objetivos que suelen contradecirse:

- 1.- Que el algoritmo sea fácil de entender, codificar y depurar.
- 2.- Que el algoritmo use eficientemente los recursos del computador y, en especial que se ejecute con mayor rapidez posible.

Cuando se escribe un programa que se va a usar una o pocas veces, el primer objetivo es el más importante. En tal caso, es muy probable que el costo del tiempo de costo a optimar es el de escritura del programa. En cambio, cuando se presenta un problema cuya solución se va a utilizar muchas veces, el costo de ejecución del programa puede superar en mucho al de escritura, en especial si en la mayor parte de las ejecuciones se dan entradas de gran tamaño.

### **-Medición del tiempo de ejecución de un programa**

El tiempo de ejecución de un programa depende de factores como:

- 1.- Los datos de entrada al programa.
- 2.- La calidad del código generado por el compilador utilizado para crear el programa objeto.
- 3.- La naturaleza y rapidez de las instrucciones de máquina empleadas en la ejecución del programa.
- 4.- La complejidad del tiempo del algoritmo base del programa

## **Eficiencia de Algoritmos**

Existen diferentes métodos con los que se trata de medir la eficiencia de los algoritmos; entre ellos están los que se basan en el número de operaciones que debe efectuar un algoritmo para realizar una tarea, y otros métodos se enfocan en tratar de medir el tiempo que se emplea en llevar a cabo

una determinada tarea, pues lo que más importa al usuario final es que ésta se efectúe de forma correcta y en el menor tiempo posible.

### **-Notación asintótica**

Esta es una de las formas de expresar la eficiencia de los algoritmos y denomina así porque se trata de expresar el tiempo que se requiere para efectuar la tarea en función del tamaño de la muestra, en el límite cuando el tamaño es muy grande. Esto hace que, en muchos casos, cuando la muestra es pequeña no sean válidos los resultados.

Ejemplo:

Si  $f(N) = N^3 + \log_2(N)$  la función asintótica para N muy grande es:  $f(N) = N^3$

### **-Notación orden de magnitud**

Esta notación es asintótica, también se denomina notación O y es una de las más usadas; por ello se utiliza en el libro para determinar la eficiencia de los algoritmos. Como su nombre lo indica, se trata de encontrar el orden de *magnitud*<sup>4</sup> para predecir el comportamiento, en vez de fórmulas que aparentemente son más aproximadas, pero que en muchos casos no se pueden generalizar

Ejemplo:

$$\text{Si } f(N) = \frac{5}{6} N^3 \text{ ó } (f(N)) = O(N^3)$$

Lo que implica que el tiempo requerido es del orden de  $N^3$

### **Referencias:**

- Sánchez, Jesús (1998). *Introducción al análisis de algoritmos*. México: trillas
- Sisa, A. (2002). *Estructuras de datos y algoritmos: Con énfasis en programación orientada a objetos*. Colombia: Pearson Educación.
- Torres, C. (1992). *Diseño y análisis de algoritmos*. Paraninfo
- Aho, A., Hopcroft, J., Ullman, Jeffrey. (1998). *Estructura de datos y algoritmos* (2da edición). México: S.A Alhambra Mexicana.