

Análisis de algoritmo, complejidad en espacio de memoria, ejecución y eficiencia del algoritmo.

Un algoritmo es un conjunto de pasos lógicos y estructurados que nos permiten dar solución a algún problema.

La importancia de un algoritmo radica en desarrollar un razonamiento lógico matemático a través de la comprensión y aplicación de metodologías para la resolución de problemáticas, apoyando así al razonamiento crítico deductivo e inductivo.

Un algoritmo debe ser además:

- ❖ General: Es deseable que un algoritmo sea capaz de resolver una clase de problemas lo más amplia posible.
- ❖ Eficiente: Un algoritmo es eficiente cuantos menos recursos en tiempo, espacio (de memoria) y procesadores consume.

La definición de un algoritmo debe describir tres partes: Entrada, Proceso y Salida.

Características de los algoritmos:

- Un algoritmo debe ser preciso e indicar el orden de realización de cada paso.
- Un algoritmo debe estar definido. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
- Un algoritmo debe ser finito. Si se sigue un algoritmo, se debe terminar en algún momento; o sea, debe tener un número finito de pasos.

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

1. Tamaño o valor de la entrada al programa.
2. Calidad del código generado por el compilador.
3. Naturaleza y rapidez de las instrucciones en la máquina utilizada para ejecutar el programa.
4. Complejidad en el tiempo del algoritmo representado por el programa.

El hecho de que el tiempo de ejecución dependa de la entrada nos dice que el tiempo de ejecución de un programa debe definirse como una función de la entrada (o su tamaño). Un buen ejemplo es el de ordenamiento, en este caso tenemos como entrada una lista de elementos a ser ordenados y producimos como salida los mismos elementos en orden.

Complejidad de los algoritmos:

La complejidad algorítmica representa la cantidad de recursos (temporales) que necesita un algoritmo para resolver un problema y por tanto permite determinar la eficiencia de dicho algoritmo.

Los criterios que se van a emplear para evaluar la complejidad algorítmica no proporcionan medidas absolutas sino medidas relativas al tamaño del problema.

Existe un análisis de la **complejidad**, en el que se observan dos enfoques diferentes:

- Enfoque empírico: se ejecutan en máquina y se comparan resultados.
- Enfoque teórico: se determina matemáticamente a priori los recursos utilizados.

Respecto al uso eficiente de los recursos, éste suele medirse en función de dos parámetros: el espacio, es decir, memoria que utiliza, y el tiempo, lo que tarda en ejecutarse. Ambos representan los costes que supone encontrar la solución al problema planteado mediante un algoritmo. Dichos parámetros van a servir además para comparar algoritmos entre sí, permitiendo determinar el más adecuado de entre varios que solucionan un mismo problema.

Para realizar la evaluación de la **eficiencia**, se determinan:

- Eficiencia en memoria: llamada complejidad espacial, que representa la cantidad de almacenamiento necesario.
- Eficiencia en tiempo: como no se ejecuta no existe una unidad determinada para medir la eficiencia teórica, por ello se utiliza el principio de invarianza, que plantea que dos versiones del mismo algoritmo no difieren en su eficiencia en más que una constante: $T_1(n) \leq C T_2(n)$ donde n es grande y C es positiva y dependerá de la computadora.

Referencias:

Cairó Olvaldo, Metodología de la programación: Algoritmos, diagramas de flujo y programas. (2ª ed.) Editorial Alfaomega.

Joyanes Aguilar Luis. (1999). Estructura de datos: Algoritmos, abstracción y objetos. (3ª ed.). Madrid, España: Mc Graw Hill.

Galvez, Javier. Algorítmica, diseño y análisis de algoritmos. México, Addison-Wesley.

Levine, Guillermo. (2001) Computación y Programación Moderna. Perspectiva integral de la informática. México, Addison-Wesley.