

TAREA #5 ANÁLISIS DE ALGORITMOS, COMPLEJIDAD EN ESPACIO DE MEMORIA Y EJECUCIÓN, EJECUCIÓN DEL ALGORITMO.

CINTHIA GUADALUPE OLIVAS CALDERON NO.17212165

Un algoritmo es un conjunto de instrucciones sencillas, claramente especificadas, que se debe seguir para resolver un problema.

Una vez que tenemos un algoritmo que resuelve un problema y podemos decir que es de alguna manera correcto, un paso importante es tener idea de la cantidad de recursos, como tiempo de procesador o espacio en la memoria principal que requerirá.

Características de un algoritmo

- Preciso (indica el orden de realización en cada paso)
- Definido (si se sigue dos veces, obtiene el mismo resultado cada vez)
- Finito (tiene fin; un número determinado de pasos).

Factores que influyen en la eficiencia de un algoritmo

Podemos tomar en cuenta muchos factores que sean externos al algoritmo como la computadora donde se ejecuta (hardware y software) o factores internos como la longitud de entrada del algoritmo. Veamos algunos de estos factores.

- El Hardware. Por ejemplo: procesador, frecuencia de trabajo, memoria, discos, etc.
- El Software. Por ejemplo: sistema operativo, lenguaje de programación, compilador, etc.
- La longitud de entrada. El enfoque matemático considera el tiempo del algoritmo como una función del tamaño de entrada. Normalmente, se identifica la longitud de entrada (tamaño de entrada), con el número de elementos lógicos contenidos en un ejemplar de entrada, por ejemplo: en un algoritmo que calcula el factorial de un número, la longitud de entrada sería el mismo número, porque no es lo mismo calcular el factorial de 4 que calcular el factorial de 1000, las iteraciones que tenga que hacer el algoritmo dependerá de la entrada. De igual manera se puede considerar como longitud de entrada: al tamaño de un arreglo, el número de nodos de una lista enlazada, el número de registros de un archivo o el número de elementos de una lista ordenada). A medida que crece el tamaño de un ejemplar

del programa, generalmente, crece el tiempo de ejecución. Observando cómo varía el tiempo de ejecución con el tamaño de la entrada, se puede determinar la tasa de crecimiento del algoritmo, expresado normalmente en términos de n , donde n es una medida del tamaño de la entrada. La tasa de crecimiento de un problema es una medida importante de la eficiencia ya que predice cuánto tiempo se requerirá para entradas muy grandes de un determinado problema. Para que un algoritmo sea eficiente, se debe optimizar el tiempo de ejecución y el espacio en la memoria, aunque se producirá la optimización de uno a costa del otro.

Tiempo de ejecución de un programa

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 la 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 programación exceda en mucho al costo de ejecución del programa, de modo que el 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. Entonces, es más ventajoso, desde el punto de vista económico, realizar un algoritmo complejo siempre que el tiempo de ejecución del programa resultante sea significativamente menor que el de un programa más evidente. Y aun en situaciones como esa, quizá sea conveniente implantar primero un algoritmo simple, con el objeto de determinar el beneficio real que se obtendría escribiendo un programa más complicado

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.

Referencias:

- Joyanes Aguilar Luis. (1999). Estructura de datos: Algoritmos, abstracción y objetos. (3a ed.). Madrid, España: Mc Graw Hill.
- Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1988). Estructuras de datos y algoritmos (Vol. 1). Addison-Wesley Iberoamericana.
- Brassard Glen, Breatley Paul, "Fundamentos de Algoritmia" España, Prentice Hall International 579 p (1997)