Algoritmo

Un algoritmo es un método, un proceso, un conjunto de instrucciones utilizadas para resolver un problema específico. Un problema puede ser resuelto mediante muchos algoritmos. Un algoritmo dado correcto, resuelve un problema definido y determinado (por ejemplo, calcula una función determinada).

Analisis de Algoritmo

La ventaja de conocer varias solucione un problema es que las diferentes soluciones pueden ser más eficientes para variaciones específicas del problema o para diferentes entradas del mismo problema. Por ejemplo, un algoritmo de ordenación puede ser el mejor para ordenar conjuntos pequeños de números, otro puede ser el mejor para ordenar conjuntos grandes de números y un tercero puede ser el mejor para ordenar cadenas de caracteres de longitud variable. Desde un punto de vista más formal y riguroso, un algoritmo es “un conjunto ordenado de pasos o instrucciones ejecutables y no ambiguas”. Obsérvese en la definición que las etapas o pasos que sigue el algoritmo deben tener una estructura bien establecida en términos del orden en que se ejecutan las etapas. Esto no significa que las etapas se deban ejecutar en secuencia: una primera etapa,después una segunda, etc. Algunos algoritmos, conocidos como algoritmos paralelos, por ejemplo, contienen más de una secuencia de etapas, cada una diseñada para ser ejecutada por procesadores diferentes en una máquina multiprocesador. En tales casos, los algoritmos globales no poseen un único hilo conductor de etapas que conforman el escenario de primera etapa, segunda etapa, etc. En su lugar, la estructura del algoritmo es el de múltiples hilos conductores que se bifurcan y se reconectan a medida que los diferentes procesadores ejecutan las diferentes partes de la tarea global. Durante el diseño de un algoritmo, los detalles de un lenguaje de programación específico se pueden obviar frente a la simplicidad de una solución. Generalmente, el diseño se escribe en español (o en inglés, o en otro idioma

hablado). También se utiliza un tipo de lenguaje mixto entre el español y un lenguaje de programación universal que se conoce como pseudocódigo.

Complejidad en Espacio y ejecución

La medida del rendimiento de un programa se consigue mediante la complejidad de espacio y del tiempo de un programa. La complejidad del espacio de un programa es la cantidad de memoria que se necesita para ejecutarlo hasta la compleción (terminación). El avance tecnológico proporciona hoy en día memoria abundante; por esa razón, el análisis de algoritmos se centra fundamentalmente en el tiempo de ejecución. La complejidad del tiempo de un programa es la cantidad de tiempo de computadora que se necesita para ejecutarlo. Se utiliza una función, T(n), para representar el número de unidades de tiempo tomadas por un programa o algoritmo para cualquier entrada de tamaño n. Si la función T(n) de un programa es T(n) = c * n, entonces el tiempo de ejecución es linealmente proporcional al tamaño de la entrada sobre la que se ejecuta. Tal programa se dice que es de tiempo lineal o, simplemente lineal.

Eficiencia del algoritmo

Eficiencia de un algoritmo 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. Un buen programador buscará el algoritmo más eficiente dentro del conjunto de aquellos que resuelven con exactitud un problema dado. ¿Cómo medir la eficiencia de un algoritmo o programa informático? Uno de los métodos más sobresalientes es el análisis de algoritmos, que permite medir la dificultad inherente de un problema. Los restantes capítulos utilizan con frecuencia las técnica de análisis de algoritmos siempre que estos se diseñan. Esta característica le permitirá comparar algoritmos para la resolución de problemas en términos de eficiencia. Aunque las máquinas actuales son capaces de ejecutar millones de instrucciones por segundo, la eficiencia permanece como un reto o preocupación a resolver. Con frecuencia, la elección entre algoritmos eficientes e ineficientes pueden mostrar la diferencia entre una solución práctica a un problema y una no práctica. En los primeros tiempos de la informática moderna (décadas de los 60 a los 80), las computadoras eran muy lentas y tenían una capacidad de memoria pequeña. Los programas tenían que ser diseñados cuidadosamente para hacer uso de los recursos escasos, como almacenamiento y tiempo de ejecución. Los programadores gastan horas intentando recortar radicalmente segundos a los tiempos de ejecución de sus programas o intentando comprimir los programas en un pequeño espacio en memoria utilizando todo tipo de tecnologías de comprensión y reducción de tamaño. La eficiencia de un programa se medía en aquella época como un factor dependiente del binomio espacio-tiempo. Hoy, la situación ha cambiado radicalmente. Los costes del hardware han caído drásticamente, mientras que los costes humanos han aumentado considerablemente. El tiempo de ejecución y el espacio de memoria ya no son factores críticos como lo fueron anteriormente. Hoy día, el esfuerzo considerable que se requería para conseguir la eficiencia máxima no es tan acusado, excepto en algunas aplicaciones como, por ejemplo, sistemas tiempo real con factores críticos de ejecución. Pese a todo, la eficiencia sigue siendo un factor decisivo en el diseño de algoritmos y construcción posterior de programas. 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 el menor tiempo posible. Sin embargo, estos métodos presentan varias dificultades, ya que

cuando se trata de generalizar la medida hecha, 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.

Bibliografias

Luis Joyanes Aguilar, Estructura de datos en Java.2007