



Instituto Tecnológico de Tijuana

Análisis de Algoritmo

Materia:

Estructura de Datos

Profesor(a):

Ray Brunett Parra Galaviz

Alumno(a):

Jiménez Mayoral Gloria Alejandra – 17212146

Fecha:

04 de septiembre de 2018

5. Análisis de algoritmo

Es una rama de las ciencias de la computación que estudia la eficiencia de los algoritmos.

Puede entenderse como la estimación del consumo de recursos que un algoritmo requiere, para lo cual se utilizan herramientas analíticas, de tal manera que sea posible establecer el rendimiento del programa que lo utiliza, o va a utilizar, y comparar los costos relativos de dos o más algoritmos para resolver un mismo problema. Pretende obtener el orden de magnitud de tiempo requerido para la ejecución del mismo. El análisis normalmente debe ser independiente del lenguaje o la máquina que se utilice para implementar el algoritmo.

El uso del análisis de algoritmos es el que permite evaluar las estructuras de datos de un programa, de tal manera que se pueda establecer la eficiencia de un algoritmo utilizando diferentes estructuras.

La eficiencia es un criterio que se debe utilizar en la selección de un algoritmo y su posterior implementación. Una comparación de algoritmos se debe centrar en diferencias significativas en eficiencia. ¿Cómo comparar la eficiencia de tiempo de dos algoritmos que resuelven el mismo problema? Un posible enfoque es implementar los dos algoritmos en local y ejecutar los programas. Existen al menos tres interrogantes fundamentales con este enfoque:

1. ¿Cómo se codifican los algoritmos?
2. ¿Qué computadora utilizará?
3. ¿Qué datos debe utilizar el programa?

- **Análisis del orden de magnitud**

Las técnicas matemáticas analizan algoritmos con independencia de implementaciones específicas, computadoras o datos. El enfoque matemático considera un enfoque de tiempo del algoritmo como una función del tamaño del problema. El medio para medir el tamaño del problema depende de la aplicación. A medida que crece el tamaño del programa, generalmente, crece el tiempo de ejecución.

Observando como varía el tiempo de ejecución con el tamaño de la entrada, se puede determinar la *tasa de crecimiento* del programa o algoritmo, expresado normalmente en términos de n , donde n es una medida del tamaño o complejidad de la entrada.

- **Consideraciones de eficiencia**

Las consideraciones fundamentales son: *el tiempo y el espacio*. Se deben optimizar ambos conceptos, aunque a menudo se producirá la optimización de un concepto a costa del otro. Al considerar el tiempo necesario para ordenar un archivo de tamaño n , no son interesantes las unidades del tiempo real, porque éstas varían de unas máquinas a otras, de un programa a otro y de un conjunto de datos a otro. Lo que

realmente interesa es el cambio correspondiente a la cantidad de tiempo para ordenar un archivo, inducido por un cambio en el tamaño del archivo n .

- **Análisis de rendimiento**

La medida del rendimiento o prestaciones se consigue mediante la complejidad del tiempo y espacio de un programa.

La *complejidad del espacio* de un programa es la cantidad de memoria que se necesita para ejecutar la compleción (terminación).

La *complejidad del tiempo* de un programa es la cantidad de tiempo de computadora que se necesita para ejecutar hasta la compleción (completado totalmente).

- **Tiempo de ejecución**

Es conveniente utilizar una función $T(n)$ para representar el número de unidades de tiempo tomadas por un programa o un algoritmo de cualquier entrada de tamaño n . $T(n)$ es el tiempo de ejecución de un programa. Por ejemplo, un programa puede tener un tiempo de ejecución $T(n) = c * n$, donde c es una constante. El tiempo de ejecución de un programa es linealmente proporcional al tamaño de la entrada sobre la que se ejecuta. Tal programa o algoritmo se dice que es *tiempo lineal*.

Bibliografía

- Joyanes A. L., Zahonero M. I. (2004) *Algoritmos y estructuras datos. Una perspectiva en C*. España: McGraw-Hill.