**Órdenes**

**Tiempo de Ejecución de un Programa**

Al resolver un problema, debemos elegir el algoritmo a utilizar, siguiendo los siguientes objetivos:

Algoritmo fácil de entender, codificar y depurar,

Uso eficiente de los recursos y ejecución veloz.

En la mayoría de los casos, estos objetivos son contrapuestos.

Depende, en gran forma, de la cantidad de veces que el programa se va a ejecutar.

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

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

* Los datos de entrada al programa,
* La calidad del código generado por el compilador,
* La naturaleza y rapidez de las instrucciones usadas en el programa,
* La complejidad de tiempo del algoritmo base del programa.

Muchas veces el tiempo depende del tamaño de la entrada más que de la entrada en sí misma.

La longitud de la entrada es una medida apropiada de tamaño.

Se denomina ***T(n)*** el tiempo de ejecución de un programa con entrada de tamaño ***n***.

Cuando el tiempo de ejecución es función de una entrada específica y no sólo del tamaño, entonces se define ***T(n)*** como el tiempo de ejecución del *peor caso*.

También se podría calcular el ***Tprom(n)***, o sea, el tiempo promedio de ejecución, pero suele ser bastante más difícil.

El hecho de que el tiempo de ejecución dependa del compilador y de la máquina impide que se pueda expresar en unidades de tiempo normales.

Se dirá entonces, que, por ejemplo, “el algoritmo es proporcional a ***n2***”.

**Notación asintótica – Orden del tiempo de ejecución**

Se dice que el tiempo de ejecución de un programa es de ***orden n***, (***T(n)*** es ***O(n)***), si existen dos constantes enteras positivas ***c*** y ***no*** tales que,

Para ***todo n >= no***, ***T(n) <= c \* n***

Se dice que ***T(n)*** es ***O(f(n))*** si existen dos constantes enteras positivas ***c***  y ***no*** tales que,

Para ***todo n >= no, T(n) <= c \* f(n)***

En este caso, se dice que el programa tiene *velocidad de crecimiento* ***f(n)***.

Si ***T(n)*** es ***O(f(n)), f(n)*** es una cota superior para la velocidad de crecimiento de ***T(n)***.

La cota inferior se especifica notando ***T(n)*** es Ω***(g(n))***. Esto significa que existe una constante **c** tal que ***T(n) >= c \* g(n)*** para un número infinito de valores de n.

Suponemos que podemos evaluar programas comparando sus funciones de tiempo de ejecución, sin considerar las constantes de proporcionalidad. De acuerdo a esto, un programa con tiempo de ejecución O(n2) es mejor que uno con tiempo de ejecución O(n3).

Además de los factores del compilador y la máquina, existe un factor debido al programa mismo.

Ej: 100 \* n2 milisegundos vs 5 \* n3 milisegundos.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

**Tiempo de ejecución**

Si un programa se va a ejecutar poco, el costo de su desarrollo es el dominante.

Si el programa se ejecuta sólo con entradas pequeñas, la velocidad de crecimiento puede ser menos importante que el factor constante.

Un algoritmo eficiente pero complicado aún puede no ser adecuado, por razones de mantenimiento.

Algunos algoritmos eficientes requieren demasiado espacio, debiéndose utilizar almacenamiento secundario, con lo cual pierden su eficiencia.

En los algoritmos numéricos, la precisión y la estabilidad son tan importantes como la eficiencia.

**Cálculo del tiempo de ejecución de un programa**

Reglas de la suma y producto en notación asintótica.

El tiempo de ejecución de cada proposición de asignación puede tomarse como O(1).

El tiempo de ejecución de una secuencia de proposiciones se determina por la regla de la suma.

El tiempo de ejecución de una proposición condicional if es el costo de las proposiciones que se ejecutan condicionalmente, más el tiempo para evaluar la condición (se toma O(1)).

El tiempo para ejecutar un ciclo es la suma sobre todas las iteraciones del ciclo, del tiempo de ejecución del cuerpo y del empleado para evaluar la condición de terminación. Normalmente este tiempo es el producto del número de iteraciones del ciclo por el mayor tiempo posible para una ejecución del cuerpo.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto, Carta

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente