**ESTRUCTURA DE DATOS II**

**Tiempo de Ejecución**

Una medida que suele ser útil conocer es el tiempo de ejecución de un programa en función de N, lo que denominaremos T(N). Esta función se puede medir físicamente (ejecutando el programa, reloj en mano), o calcularse sobre el código contando instrucciones a ejecutar y multiplicando por el tiempo requerido por cada instrucción. Así, un trozo sencillo de programa como:

S1; for (int i= 0; i < N; i++) S2;

Requiere

T(N)= t1 + t2\*N

Siendo t1 el tiempo que lleve ejecutar la serie "S1" de sentencias, y t2 el que lleve la serie "S2".

Prácticamente todos los programas reales incluyen alguna sentencia condicional, haciendo que las sentencias efectivamente ejecutadas dependan de los datos concretos que se le presenten. Esto hace que más que un valor T(N) debamos hablar de un rango de valores

Tmin(N) <= T(N) <= Tmax(N)

Los extremos son habitualmente conocidos como "caso peor" y "caso mejor". Entre ambos se hallara algún "caso promedio" o más frecuente.

Cualquier fórmula T(N) incluye referencias al parámetro N y a una serie de constantes "Ti" que dependen de factores externos al algoritmo como pueden ser la calidad del código generado por el compilador y la velocidad de ejecución de instrucciones del ordenador que lo ejecuta. Dado que es fácil cambiar de compilador y que la potencia de los ordenadores crece a un ritmo vertiginoso (en la actualidad, se duplica anualmente), intentaremos analizar los algoritmos con algún nivel de independencia de estos factores; es decir, buscaremos estimaciones generales ampliamente válidas.

***Complejidad Espacial***

Memoria que utiliza un programa para su ejecución, La eficiencia en memoria de un algoritmo indica la cantidad de espacio requerido para ejecutar el algoritmo; es decir, el espacio en memoria que ocupan todas las [variables](http://www.monografias.com/trabajos12/guiainf/guiainf.shtml#HIPOTES) propias al algoritmo. Para calcular la *memoria estática* de un algoritmo se suma la memoria que ocupan las variables declaradas en el algoritmo. Para el caso de la *memoria dinámica*, el cálculo no es tan simple ya que, este depende de cada ejecución del algoritmo.  
  
Cálculo de Costos de Algoritmos Queremos saber la eficiencia de los algoritmos, no del computador. Por ello, en lugar de medir el tiempo de ejecución en microsegundos o algo por el estilo, nos preocuparemos del número de veces que se ejecuta una operación primitiva (de tiempo fijo). Para estimar la eficiencia de este algoritmo, podemos preguntarnos, "si el argumento es una frase de N números, ¿cuántas multiplicaciones realizaremos?" La respuesta es que hacemos una multiplicación por cada número en el argumento, por lo que hacemos N multiplicaciones. La cantidad de tiempo que se necesitaría para el doble de números sería el doble.