

UT2 - RAT

Ordenes

Un algoritmo debe ser fácil de entender, codificar y depurar, tiene que hacer uso eficiente de los recursos y tener una ejecución veloz. En la mayoría de los casos estos objetivos se contraponen, depende de la cantidad de veces que se ejecutará el programa cual se prioriza.

El tiempo de ejecución depende de factores como:

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

$T(n)$ es 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.

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 n_0 tales que:

para todo $n \geq n_0$, $T(n) \leq c * n$

Se dice que $T(n)$ es $O(f(n))$ si existen dos constantes enteras positivas c y n_0 tales que,

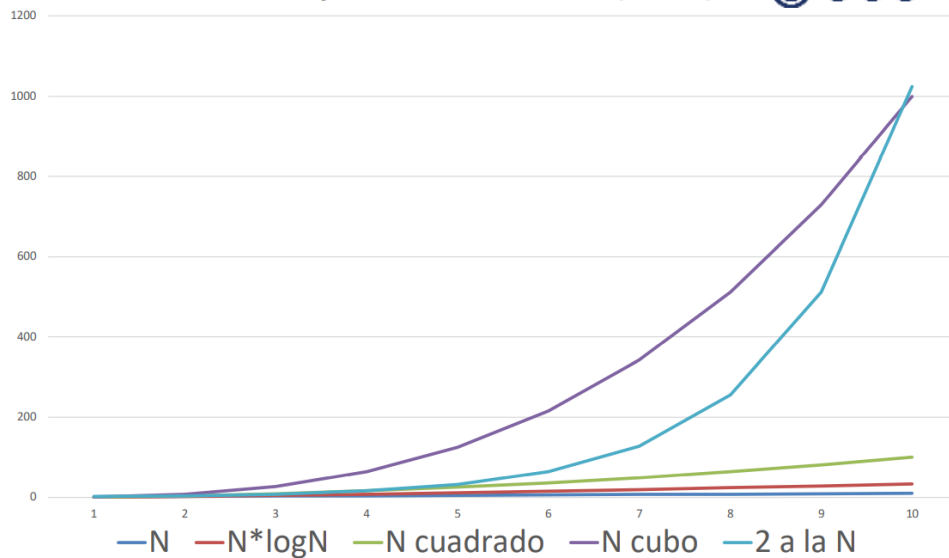
para todo $n \geq n_0$, $T(n) \leq c * f(n)$

– 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 $T(n)$.

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 $O(n^2)$ es mejor que uno con tiempo de ejecución $O(n^3)$.

Comparación de órdenes (todos)



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

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 ($O(1)$).

El tiempo para ejecutar un ciclo es la suma sobre todas las iteraciones del ciclo, del tiempo de ejecución del cuerpo y del usado 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.