

Análisis de Algoritmos

(Complejidad)

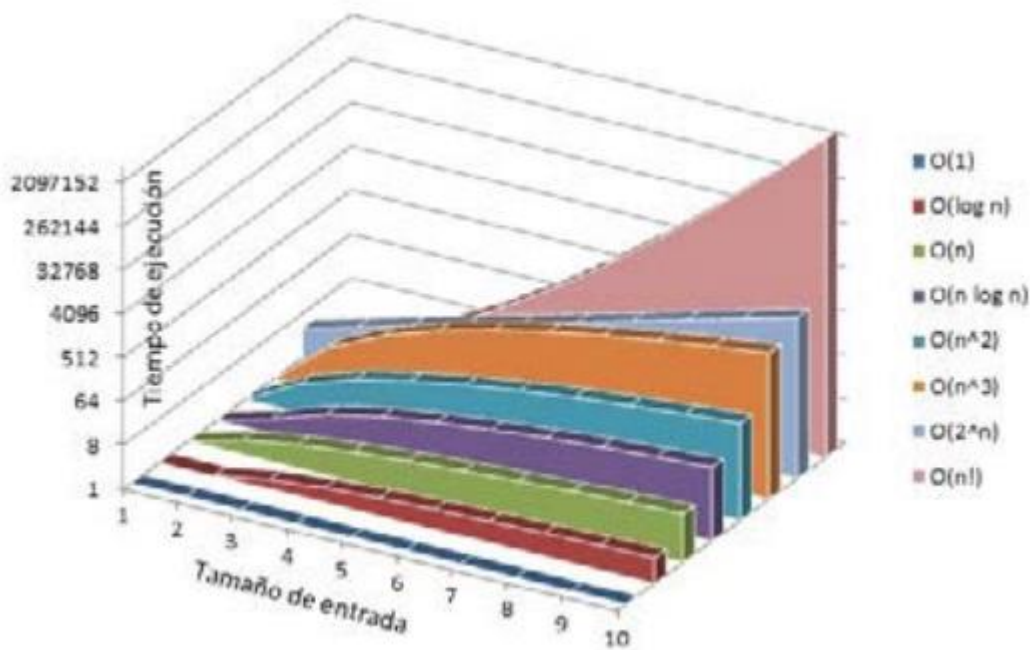
Enfoques para el análisis de algoritmos

Hay dos enfoques:

- Empírico (o a posteriori): Tengo un algoritmo A y un algoritmo B, los ejecuto mido el tiempo y determino una vez que termino cual es el mejor. El problema es que tengo que correr pruebas en cada uno, en donde tienen distintos tamaños, en donde lleva un montón de tiempo para finalmente tratar de determinar cuál es el mejor algoritmo.
- Teórico (o a priori): Hacemos una medida de la complejidad del algoritmo basado en una notación $O()$, que me da un valor, un orden de magnitud de ese algoritmo. En donde el orden de ese algoritmo se puede comparar con otro algoritmo para determinar cuál es más rápido o cual es más complejo.

¿De qué manera podemos medir la eficiencia de un algoritmo?:

Notación **O** grande



¿Qué caso es el más útil?

- Mejor Caso
- Caso Medio
- Peor caso

El caso que es más útil es el peor caso, ya que garantizo que en el **peor de los casos** mi algoritmo va a tener esa complejidad, por lo tanto, la **máxima** complejidad va a ser esa.

Reglas para el análisis de algoritmos:

- Operación elemental: es una operación donde su costo operacional es de 1, una operación elemental es sumar dos números, hacer un print, hacer un input, hacer una asignación.

•**Secuencia:** conjunto de instrucciones, en donde la complejidad de esa secuencia de instrucciones es la suma de cada una de esas instrucciones, por ejemplo: si tengo un input, después una operación aritmética y después un print, tendría bloque de complejidad de 3. Si tengo dos bloques secuenciales la complejidad es la suma de ambos

•**Condicional:** El costo de complejidad va a ser 1 por cada condición que yo tenga, más el máximo coste que tenga tanto por la rama verdadera tanto por la rama falsa.

•**Ciclo:**

-**Ciclo for:** la cantidad de iteraciones multiplicado por la complejidad del bloque que tengo adentro. Si tengo otro for a dentro tendría que multiplicar el for por el bloque de código de cada ciclo.

-**Ciclo while:** Es prácticamente imposible saber la cantidad exacta, pero están asociados a una complejidad, y se multiplican al igual que el for.

•**Recursividad:** Para realizar este cálculo existen dos técnicas comúnmente utilizadas: ecuaciones recurrentes y teoremas maestros. Para realizar la primera de ellas se busca una ecuación que elimine la recursividad para poder obtener el orden de dicha función, muchas veces esto no es una tarea sencilla. En cambio, el segundo utiliza funciones condicionales y condiciones de regularidad para realizar el cálculo del coste.

Ejemplos:

Hay que ver en el peor de los casos, el que me indica más o menos normal es n . La búsqueda secuencial es igual a n .

-Cuando tengo un resultado: $n \cdot n = n^2$, pero cuando tengo n^4 se saca el 4 y queda solo el n porque la constante es insignificante.

Para ver ejemplos de ejercicios entrar al siguiente ENLACE o CODIGO QR (clase grabada):

ENLACE: <https://youtu.be/UsLdl3qRy-g>

CODIGO QR:

