NOTACIÓN O GRANDE

1. INTRODUCCIÓN

La *O* mayúscula es una notación matemática también conocida Bachmann–Landau notation, asymptotic notation, notación o grande. Y esta nos ayuda a describir el comportamiento de una función "al límite". Esta notación tiene muchos usos en realidad, sin embargo el que nos interesa ahora es el que se le da en las ciencias de la computación.

En las ciencias de la computación esta notación nos ayuda a describir el comportamiento de un algoritmo, ya sea temporal (cuantas unidades de tiempo va a tardar en ejecutarse) o en espacio (cuanta memoria va a ocupar mientras se ejecuta), esto basado en la cantidad de entradas que recibe y sobre los que opera. A este "comportamiento" también se le conoce como complejidad de un algoritmo. Este análisis de complejidad nos sirve generalmente cuando las entradas de el algoritmo analizado son muy grandes.

Por poner un ejemplo, supón que tu amiga te va a dar un regalo, pero para hacer las cosas más interesantes tomó N cajas, las colocó en una fila y escondió el regalo en una de ellas. Entonces tu tienes que ir destapando cada caja hasta encontrar tu regalo, sabes que tendrás que abrir a lo mucho N cajas, entonces podríamos decir que tu "algoritmo" de búsqueda tiene una complejidad temporal de $O(n)$.

Algo bastante útil de este análisis es que funciona independientemente de quién o en donde se ejecute el algoritmo, siguiendo con el ejemplo de las cajas y el regalo. Puede que tu te tardes 3 segundos en abrir cada caja, pero tu amiga las pueda abrir en 2. No importa cual tarde menos, si los dos usan el mismo algoritmo, la complejidad será la misma, $O(n)$.

De igual manera si determinado algoritmo se tarda veinte segundos en mi computadora pero en los servidores de Google se tarda 2, se podría establecer la unidad de tiempo en dos segundos y decir que "en mi computadora se tarda C unidades de tiempo" en donde C es una constante que dependerá únicamente de las características del entorno en donde se ejecuta, no del algoritmo que está siendo ejecutado.

Una forma bastante rápida y generalmente útil para poder calcular la complejidad de un algoritmo es fijarse en el tipo de operaciones que realiza, por ejemplo si se trata de un cálculo matemático, como el cálculo de las soluciones de una ecuación de segundo grado, se dice que tiene una complejidad constante *O(1)*, ya que no importa que tan grandes sean los coeficientes de entrada, siempre realizará la misma cantidad de operaciones para llegar al resultado.

Por otro lado, si tu algoritmo requiere recorrer un arreglo unidimensional de N elementos (como el ejemplo de las cajas), sabes que cuando menos se va a tardar N unidades de tiempo en ejecutarse, o sea O(n), así mismo si tienes que recorrer una matriz de N x N se tardará N^2 y así sucesivamente.

Entre las complejidades más comunes también está la de *O(n log n),* que presentan algunos algoritmos que van dividiendo o descartando parte de las entradas sobre las que opera, como el Quicksort o la búsqueda binaria.

También existe una complejidad con la que debes tener mucho, pero mucho cuidado y esta es la complejidad *O(n!),* ya que con tan solo una entrada de 10 elementos el algoritmo estaría haciendo 3628800 operaciones. Un ejemplo de este problema es el del agente viajero:

Imagina que quieres visitar las capitales de todos los países de la Unión Europea, pero quieres encontrar la manera más corta para hacerlo entonces te pones a calcular todas las posibilidades que tienes… ¡mala idea! hay 3.0488834e+29 posibilidades para calcular, es entonces cuando surgen otro tipo de algoritmos que son un poco más complejos de analizar.

Una cota superior asintótica es una función que sirve de cota superior de otra función cuando el argumento tiende a infinito. Usualmente se utiliza la notación de Landau: *O(g(x)),* Orden de *g(x)*, coloquialmente llamada Notación *O* Grande, para referirse a las funciones acotadas superiormente por la función *g(x).*

Formalmente se define:

$$O(g(x)) = \left\{ \begin{array}{c} f(x) : \text{existen } x_0, c > 0 \text{ tales que} \\ \forall x \geq x_0 > 0 : 0 \leq |f(x)| \leq c|g(x)| \end{array} \right\}$$

2. DEFINICIÓN O GRANDE

Definición: Sean $f, g : N \rightarrow R^*$ dos funciones arbitrarias de los números naturales en los números reales no negativos. Se dice entonces que f está en O-grande de g, y se escribe $f \in O(g)$, si y sólo si existe una constante real positiva M y un número natural n0 tales que para todo número natural n ≥ n0 se tiene que $f(n) \leq M * g(n)$. Simbólicamente:

$$f \in O(g) \Leftrightarrow \exists M \in R + \exists\, n0 \in N : \forall n \in N : [n \geq n0 \Rightarrow f(n) \leq M * g(n)]$$

Definición: Sea $g : N \rightarrow R^*$ una función arbitraria de los números naturales en los números reales no negativos. $O(g)$ representa el conjunto de todas las funciones $t : N \rightarrow R^*$ tales que existe una constante real positiva M y un número natural n_0 de manera tal que para todo número natural $n \geq n_0$ se tiene que $t(n) \leq M * g(n)$.

Simbólicamente:

$$O(g) = \{t : N \rightarrow R^* \mid \exists M \in R + \exists\, n_0 \in N : \forall n \in N : [n \geq n_0 \Rightarrow f(n) \leq M * g(n)]$$

Si se desea comparar dos funciones, casi siempre alrededor de x=0, se dispone del siguiente Resultado: Se dice que cuando $x \rightarrow 0$, $f(x)$ es del "orden" de $g(x)$

Lo cual se escribe $f(x) = O(g(x))$

Si existen dos constantes positivas C y δ (pequeña) tales que:

$$|f(x)| \leq C|g(x)| \text{ para } |x| \leq \delta$$

3. EJEMPLO Y GRAFICA

Ejemplo: Sea f(x) = $4X^3 + 5X^6$. Recordemos que si $0 < y < 1$, entonces $y > y^2 > y^3 > \cdots$.

Entonces, si $|x| < 1$

$$|X^3| \leq |x|$$

$$|4x^3| \leq 4|x|$$

$$|x^6| \leq |x|$$

$$|5x^6| \leq 5|x|$$

$$|4x^3 + 5x^6| \leq 9|x|$$

$$4x^3 + 5x^6 = O(x).$$

Aunque lo anterior es cierto, es preferible buscar el mayor exponente posible.

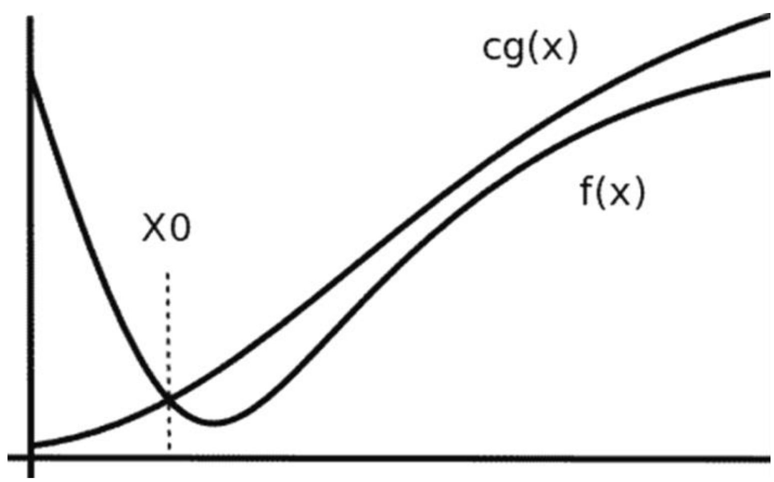
Mediante pasos semejante a los anteriores llegamos a

$$4x^3 + 5x^6 = O(x^3).$$

Obviamente no es cierto que $4x^3 + 5x^6 = O(x^4)$. 3

Según la notación *O* grande, el residuo para el polinomio de Taylor se puede expresar

$$R_n\left(\overline{x} + h\right) = O\left(h^{n+1}\right)$$



Grafica de la función *f(x) con la aproximación O grande*

4. NOTACIÓN ASINTÓTICA

Objetivo:

Simplificar el análisis eliminando la información innecesaria (como “redondeo” 1,000,001≈1,000,000)

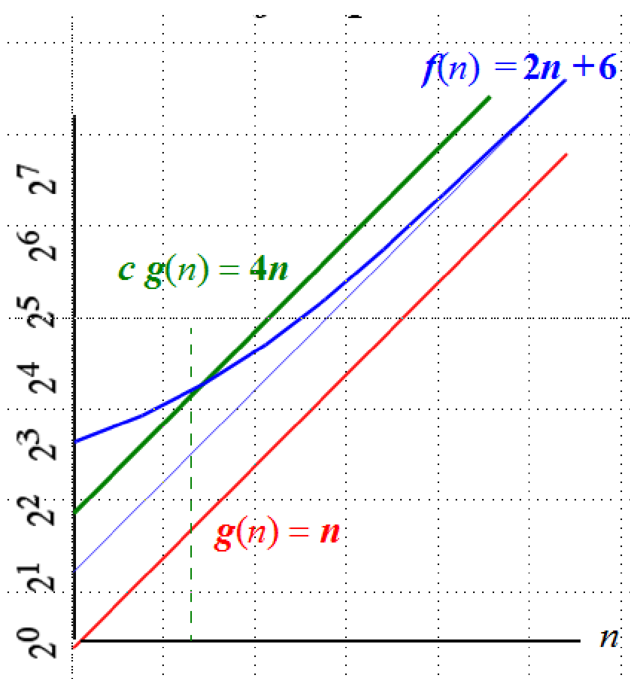
Queremos decir de manera formal $3n^2 \approx n^2$

Notación “O-grande (Big-Oh)”: dadas las funciones ***f(n)*** and *g(n),* decimos que ***f(n)*** es O(***g(n)***) si y solo si hay constantes positivas ***c*** y n_0 tal que ***f(n)≤ c g(n)*** para $n \geq n_0$

Ejemplo

Para funciones ***f(n)*** y ***g(n)*** (derecha) hay constantes positivas ***c*** y n_0 tales que:

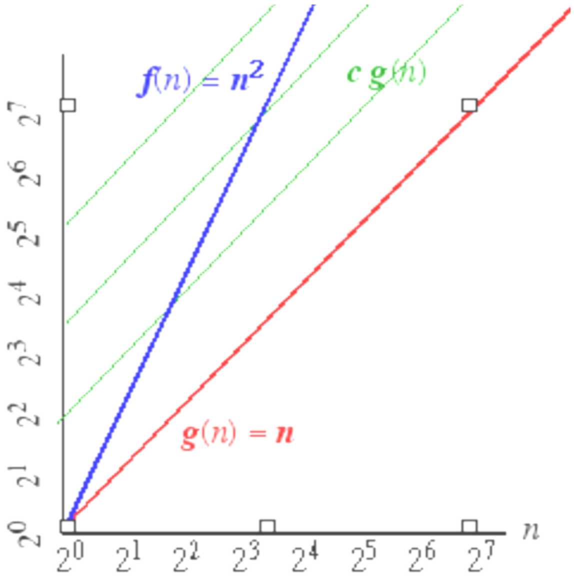
f(n)≤c g(n) for $n \geq n_0$ conclusión: 2n+6 is *O(n).*



Otro caso…

$\boldsymbol{n^2}$ no es O(***n***) debido a que no hay ***c*** y $\boldsymbol{n_0}$ tal que: $\boldsymbol{n^2} \leq \boldsymbol{cn}$ para $n \geq n_0$

(como se ve en el gráfico, no importa cuan grande se escoge c hay un n suficientemente grande que $\boldsymbol{n^2} > \boldsymbol{cn}$)



Nota: Aun cuando es correcto decir que "7n - 3 es $O(n^3)$", es mejor decir "7n - 3 es O(n)", esto es, se debe hacer la aproximación lo más cerca posible

Regla simple: Eliminar los términos de bajo orden y las constantes 7n-3 es O(n)

$8n^2\log n + 5n^2 + n$ es $O(n^2\log n)$

Clases especiales de algoritmos:

logaritmico: **O(log n)**

lineal: **O(n)**

cuadratico: **$O(n^2)$**

polinomico: **$O(n^k)$, k ≥ 1**

exponencial: **$O(a^n)$, n > 1**

- "Alternativos" de Big-Oh

  Ω (f(n)): Big Omega-- cota *inferior* asintótica

  Θ (f(n)): Big Theta-- cota *promedio* asintótica

5. ORDEN DE CONVERGENCIA

Sea $\{x_k\}$ una sucesión de números reales con Límite L, Se dice que la sucesión tiene un orden de convergencia $p \geq 1$, si p es el mayor valor tal que el siguiente límite existe:

$$\lim_{k\to\infty} \frac{|x_{k+1} - L|}{|x_k - L|^p} = \beta < \infty$$

En este caso se dice que β es la tasa de convergencia. Cuando el orden es 1, se dice que la convergencia es lineal.

La convergencia se llama "***Superlineal***", si:

$$\lim_{k\to\infty} \frac{|x_{k+1} - L|}{|x_k - L|^p} = 0$$

Cuando el orden de convergencia es 2, se dice que la convergencia es cuadrática. A mayor el orden de convergencia, la sucesión converge más rápidamente hacia el límite.

Lo ideal es tener ordenes de convergencia altos con tasas pequeñas.

Cuando el orden es 1, se dice que la convergencia es lineal.

Por lo tanto:

$$\lim_{k\to\infty} \frac{|x_{k+1} - L|}{|x_k - L|^p} = \lim_{k\to\infty} \frac{y_{k+1}}{y_k^p} = \lim_{k\to\infty} \frac{y_k^2}{y_k^p} = \lim_{k\to\infty} y_k^{2-p}$$

Entonces:

Si p=1, el límite es 0, entonces la convergencia es por lo menos lineal, y se puede afirmar que es "Superlineal".

Si p=2, el límite es 1, entonces la convergencia es cuadrática. (Vea en el libro que la convergencia no puede ser mayor que 2). Así, la convergencia es cuadrática con tasa 1.

6. ENLACES SUGERIDOS

https://es.wikipedia.org/wiki/Cota_superior_asint%C3%B3tica

7. BIBLIOGRAFÍA

Métodos Numéricos Con SCILAB, Héctor Manuel Mora Escobar, Abril 2010.

8. GLOSARIO

9. PREGUNTAS DE AUTOEVALUACIÓN

1. ¿Qué es la Notación *O grande*?
2. ¿Qué es la cota superior asintótica?
3. La convergencia se llama "*Superlineal*", si: