Elementos de complejidad algorítmica

Análisis asintótico

Verónica E. Arriola-Rios

Facultad de Ciencias, UNAM

11 de marzo de 2021

Introducción

Temas

1. Introducción
   - Algoritmos
   - Evaluación de un algoritmo

Algoritmo

Definición (Algoritmo)

Un *algoritmo* es una secuencia de pasos que transforma un valor o conjunto de valores, conocidos como *entradas*, en un valor o conjunto de valores, conocidos como *salidas*, de tal modo que éstas satisfagan un conjunto de *relaciones* previamente especificadas.

entradas $\longrightarrow$ $\boxed{\mathcal{Algoritmo(Relaciones)}}$ $\longrightarrow$ salidas

Figura: Diagrama de bloques del concepto *algoritmo*.

Temas

1. Introducción
   - Algoritmos
   - Evaluación de un algoritmo

Características a evaluar en un algoritmo

- Calcula el resultado deseado **correctamente**.
- **Robustez** (se defiende bien de datos erróneos).
- **Tiempo** de ejecución (como función de sus entradas)
- **Espacio** en memoria ocupado total o máximo.
- **Tamaño del código** (número de instrucciones).
- **Complejidad** (fácil de leer, entender, modificar).
  - Modularidad.
  - Extensibilidad.
  - Qué tan complicado es el algoritmo.

Complejidad algorítmica

Objetivo

Se desea predecir el comportamiento del algoritmo ***sin implementarlo en una máquina específica****.*

– Udi Manber

- Para ello se modela a las computadoras como máquinas de operaciones elementales.
- Un modelo de cómputo es un conjunto de suposiciones sobre:
    - las operaciones elementales que puede realizar una computadora,
    - el tiempo que le tomará ejecutarlas,
    - la forma en que almacenará los datos en memoria.

Complejidad algorítmica

Definición (Complejidad algorítmica)

El tiempo de ejecución $T_A(E)$ de un algoritmo A sobre un ejemplar E, de un problema P, es el número de operaciones elementales que requiere A para resolver E.

- Al número de operaciones elementales calculadas se le llama *tiempo de ejecución* o *desempeño computacional*.

Tamaño de un ejemplar

Definición (Tamaño de E)

El tamaño $n = |E|$ de un ejemplar E, es una medida natural del número de elementos que poseé E.

Ejemplos:

- El número de elementos en una lista.
- El número de bits requeridos para almacenar los datos.

Dependencia del tamaño

- Los criterios empleados para evaluar la complejidad algorítmica proporcionan medidas **relativas** al tamaño del problema.
- El desempeño computacional de un algoritmo T, es una función que depende del tamaño del ejemplar n.

$$T = T(n)$$

Dependencia de la estructura del problema

- Para un mismo tamaño de E, es posible tener diferentes desempeños.
- Se distinguen:
  - El mejor caso.
  - El caso promedio.
  - El peor caso.
- Por ejemplo: al buscar un dato en una lista desordenada.
  - Mejor caso: el dato se encuentra al inicio.
  - Caso promedio: Realizar varias búsquedas, en promedio, es equivalente a buscar un dato cerca de la mitad de la lista.
  - Peor caso: El dato se encuentra al final.

Notación asintótica

Notación asintótica

- Lo que más nos interesa es la **magnitud** del desempeño computacional: ¿Cómo se comporta $T(n)$ cuando n es "grande"?

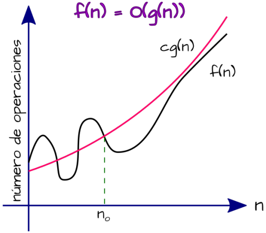
Notación asintótica

Sea $f(n) : f(n) \geqslant 0, \forall n \geqslant 0, n \in \mathbb{Z}$.

Definición ($O(n)$)

$f(n)$ es "O grande" de $g(n)$ ($f(n) = O(g(n))$) si

- $\exists n_0 \in \mathbb{Z}$ y una constante $c > 0$ tal que
- $\forall n | n \in \mathbb{Z}, n \geqslant n_0 \Rightarrow f(n) \leqslant cg(n)$



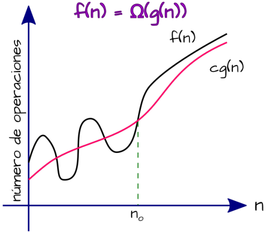
Notación asintótica

Sea $f(n) : f(n) \geqslant 0, \forall n \geqslant 0, n \in \mathbb{Z}$.

Definición ($\Omega(g(n))$)

$f(n)$ es "omega" de $g(n)$ ($f(n) = \Omega(g(n))$) si

- $\exists n_0 \in \mathbb{Z}$ y una constante $c > 0$ tal que
- $\forall n | n \in \mathbb{Z}, n \geqslant n_0 \Rightarrow f(n) \geqslant cg(n)$

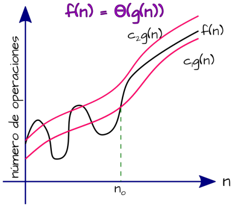


Notación asintótica

Definición ($\Theta(g(n))$)

$$\Theta(g(n)) = \{f(n) : \exists c_1, c_2 \text{ constantes con } 0 \leqslant c_1, 0 \leqslant c_2, \\ \text{y } n_0 \text{ tal que }, \forall n \geqslant n_0, c_1 g(n) \leqslant f(n) \leqslant c_2 g(n)\}$$

Es decir, $f(n)$ es $O(g(n))$ y $\Omega(g(n))$.



Notación asintótica

Sea $f(n) : f(n) \geqslant 0, \forall n \geqslant 0, n \in \mathbb{Z}$.

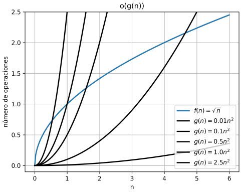
Definición ($o(n)$)

$f(n)$ es "o pequeña" de $g(n)$ ($f(n) = o(g(n))$) si

- $\forall c > 0, \exists n_0(c) \in \mathbb{Z}$ tal que
- $\forall n | n \in \mathbb{Z}, n \geqslant n_0 \Rightarrow f(n) \leqslant cg(n)$

Consecuencias:

- $\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$



Notación asintótica o(g(n)) cont.

Consecuencias (cont.):

- Si $g(n)$ es una **cota justa** de $f(n)$, se cumple $f(n) = O(g(n))$ pero no $f(n) = o(g(n))$.
  Ejemplo: Sea $f(n) = 2n^2 + 1$ y $g(n) = n^2$.
  - $f(n)$ es $O(n^2)$, sea $c = 3 \Rightarrow$ se calcula n_0:

$$\begin{aligned} 2n^2 + 1 &\leqslant 3n^2 \\ 1 &\leqslant n^2 \\ n &\geqslant \sqrt{1} = 1 \Rightarrow n_0 = 1 \quad \square \end{aligned}$$

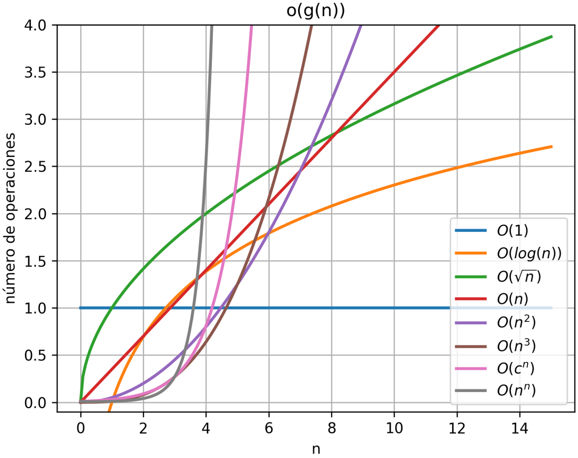
  - $f(n)$ no es $o(n^2)$, sea $c = 1 \Rightarrow$

$$\begin{aligned} 2n^2 + 1 &\leqslant n^2 \\ n^2 + 1 &\leqslant 0 \\ n^2 &\leqslant -1, \text{ pero } n_0 \in \mathbb{Z}^+! \quad \square \end{aligned}$$

Categorías de orden o(g(n))

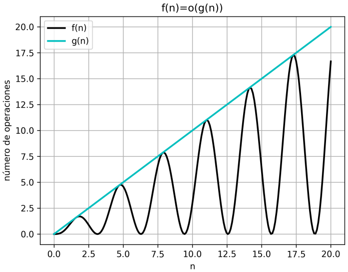
- Es posible clasificar categorías de orden con funciones características:
$$\Theta(1)\Theta(\log n)\Theta(n)\Theta(n\log n)\Theta(n^2)\underbrace{\Theta(n^j)\Theta(n^k)}_{2<j<k}\underbrace{\Theta(a^n)\Theta(b^n)}_{0<a<b}\Theta(n!)\Theta(n^n)$$
$$\xrightarrow[\text{cota superior}]{}$$
$$\xleftarrow[\text{en esta dirección no}]{}$$
- Si una función de complejidad $f(n)$ está en una categoría a la izquierda de la categoría que contiene a $g(n)$ entonces $f(n)$ es $o(g(n))$.

Categorías de orden o(g(n))



¿Definición alternativa?

- $f(n)$ es $o(g(n))$ si $f(n)$ es $O(g(n))$ pero no $\Omega(g(n))$



$$x\sin^2(x) \text{ es } O(x) \text{ pero ¿} x\sin^2(x) \text{ es } o(x)?$$

Haciéndo coincidir ambas definiciones: $x\sin^2(x)$ es $o(x^2)$, pero esta no es una cota justa.

Bibliografía

Bibliografía I

- Álvarez Bravo, José Vicente (s.f.). *Tema 5: Complejidad algorítmica*. Universidad de Valladolid, Campus de Segovia. URL: http://www.infor.uva.es/~jvalvarez/docencia/tema5.pdf.
- Cormen, Thomas H. y col. (2009). *Introduction to Algorithms*. 3rd. The MIT Press.

Licencia