Complejidad algorítmica

Clase práctica

Algoritmos y Estructuras de Datos II

Departamento de Computación

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

1$^{\text{er}}$ cuatrimestre 2020

Menú del día

¿Complejidad algorítmica?

- ¿Qué es?
- ¿Para qué se usa?
- ¿Cuál es el tamaño de la entrada?

Análisis de algoritmos

Algoritmo (parametro_de_entrada: α)	
operación elemental	$\triangleright$ c_0
otra operación elemental	$\triangleright$ c_1
OE restantes del algoritmo	$\triangleright$ costo

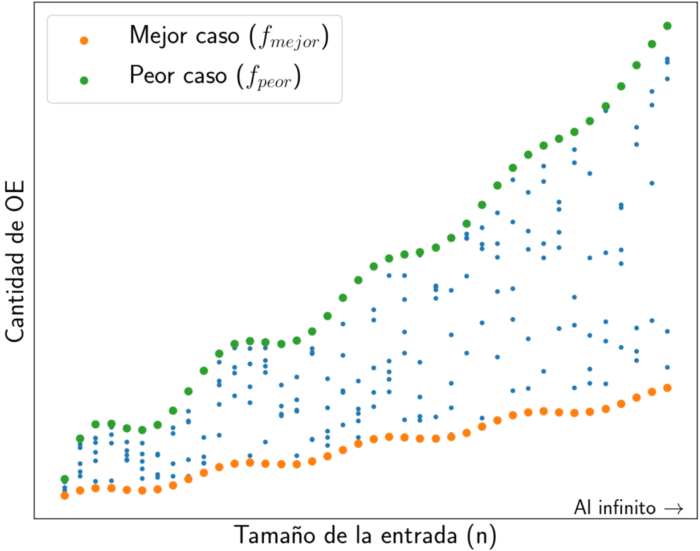
Análisis de algoritmos

Algoritmo (parametro_de_entrada: α)	
operación elemental	$\triangleright$ 1
otra operación elemental	$\triangleright$ 1
OE restantes del algoritmo	$\triangleright$ cantidad OE restantes

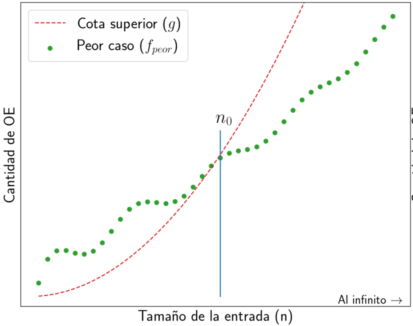
Recordar

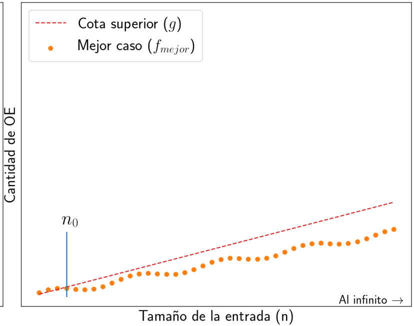
- Advertencia: el gráfico es solo para ganar intuición.
- NO ANALIZAMOS ALGORITMOS CON GRÁFICOS.
- Nos centramos en el **análisis teórico** (que se realiza con demostraciones y aplicación de propiedades).

Cantidad de OE para distintas **instancias** de tamaño *n*



Big $\mathcal{O}$ – Acotación superior





Definición

Sea $g : \mathbb{N} \to \mathbb{R}$. Entonces:

$$\mathcal{O}(g) \stackrel{\text{def}}{=} \{f : \mathbb{N} \to \mathbb{R} \mid (\exists\ n_0 \in \mathbb{N},\ c \in \mathbb{R}_{>0})\ \ f(n) \leq c \cdot g(n)\ \ \forall n \geq n_0\}$$

Big $\mathcal{O}$ – Acotación superior

Definición

Sea $g : \mathbb{N} \to \mathbb{R}$. Entonces:

$$\mathcal{O}(g) \stackrel{\text{def}}{=} \{f : \mathbb{N} \to \mathbb{R} \mid (\exists\ n_0 \in \mathbb{N},\ c \in \mathbb{R}_{>0})\ \ f(n) \leq c \cdot g(n)\ \ \forall n \geq n_0\}$$

⚠ $\mathcal{O}(g)$ denota un **conjunto de funciones**

⚠ Para poder decir que una función pertenece a una clase de funciones (ej. $f_{peor} \in \mathcal{O}(g)$), hay que **demostrarlo**.

Propiedad 1: Dada $f : \mathbb{N} \to \mathbb{R}_{>0}$, $f \in \mathcal{O}(f)$.

Propiedad 2: Dada $f : \mathbb{N} \to \mathbb{R}_{>0}$ y k cte. (positiva). Si

$$f \in \mathcal{O}(g) \Rightarrow k \cdot f \in \mathcal{O}(g)$$

Big $\mathcal{O}$ – Propiedad 1

Definición

Sea $g : \mathbb{N} \to \mathbb{R}$. Entonces:

$$\mathcal{O}(g) \stackrel{\text{def}}{=} \{f : \mathbb{N} \to \mathbb{R} \mid (\exists\ n_0 \in \mathbb{N},\ c \in \mathbb{R}_{>0})\ \ f(n) \leq c \cdot g(n)\ \ \forall n \geq n_0\}$$

qvq.(quiero ver que) dada $f : \mathbb{N} \to \mathbb{R}_{>0}$, $f \in \mathcal{O}(f)$.

$$f(n) \underset{\text{qvq}}{\leq} c \cdot f(n)$$

- Podemos elegir $c \geq 1$, por ejemplo $c = 2$.
- Luego, vale para cualquier n que tomemos. Elijo $n_0 = 3$.
- Tomando $c = 2$ y $n_0 = 3$ vale la definición.

Big $\mathcal{O}$ – Propiedad 2

Definición

Sea $g : \mathbb{N} \to \mathbb{R}$. Entonces:

$$\mathcal{O}(g) \stackrel{\text{def}}{=} \{f : \mathbb{N} \to \mathbb{R} \mid (\exists\ n_0 \in \mathbb{N},\ c \in \mathbb{R}_{>0})\ \ f(n) \leq c \cdot g(n)\ \ \forall n \geq n_0\}$$

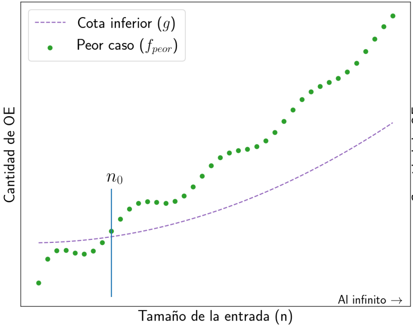
Dada $f : \mathbb{N} \to \mathbb{R}_{>0}$ y k cte. (positiva). Si

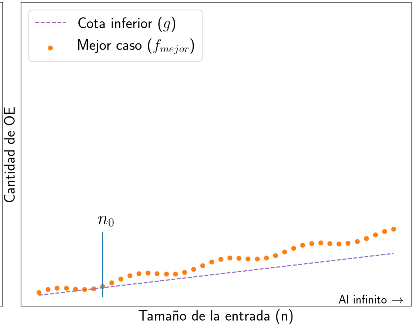
$$f \in \mathcal{O}(g) \Rightarrow k \cdot f \in \mathcal{O}(g)$$

Sabemos que $(\exists c' > 0$ y $\exists n_0')$ tq $f(n) \leq c' \cdot g(n)\ \forall n \geq n_0'$.

- $k \cdot f(n) \underset{\text{qvq}}{\leq} c \cdot g(n)$
- $k \cdot f(n) \underset{n_0'}{\leq} k \cdot c' \cdot g(n) \leq c \cdot g(n)$
- sii $k \cdot c' \leq c$ (No olvidarse que $n \geq n_0$)
- Tomando $c = k \cdot c' + 2$ y $n_0 = n_0'$ vale la definición.

Ω – Acotación inferior



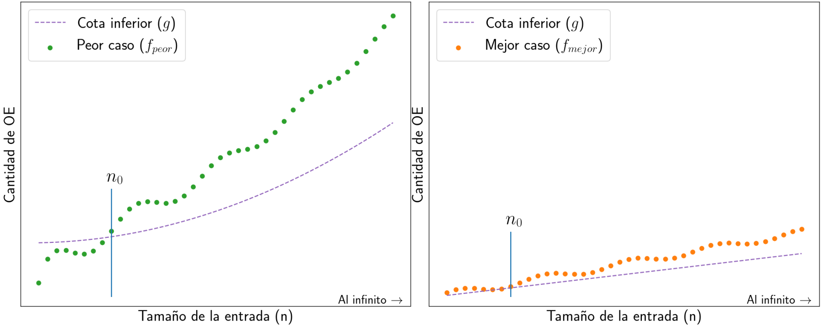


Definición

Sea $g : \mathbb{N} \to \mathbb{R}$. Entonces:

$$\Omega(g) \stackrel{\text{def}}{=} \{f : \mathbb{N} \to \mathbb{R} \mid (\exists\ n_0 \in \mathbb{N},\ c \in \mathbb{R}_{>0})\ \ f(n) \geq c \cdot g(n)\ \ \forall n \geq n_0\}$$

Ω – Acotación inferior



Definición alternativa

Sea $g : \mathbb{N} \to \mathbb{R}$. Entonces:

$$\Omega(g) \stackrel{\text{def}}{=} \{f : \mathbb{N} \to \mathbb{R} \mid \ g \in O(f)\}$$

Ω – Acotación inferior

Definición

Sea $g : \mathbb{N} \to \mathbb{R}$. Entonces:

$$\Omega(g) \stackrel{\text{def}}{=} \{f : \mathbb{N} \to \mathbb{R} \mid (\exists\ n_0 \in \mathbb{N},\ c \in \mathbb{R}_{>0})\ \ f(n) \geq c \cdot g(n)\ \ \forall n \geq n_0\}$$

Definición alternativa

Sea $g : \mathbb{N} \to \mathbb{R}$. Entonces:

$$\Omega(g) \stackrel{\text{def}}{=} \{f : \mathbb{N} \to \mathbb{R} \mid\ g \in O(f)\}$$

Propiedad 1: Dada $f : \mathbb{N} \to \mathbb{R}_{>0}$, $f \in \Omega(f)$.

Propiedad 2: Dada $f : \mathbb{N} \to \mathbb{R}_{>0}$ y k cte. (positiva). Si

$$f \in \Omega(g) \Rightarrow k \cdot f \in \Omega(g)$$

Demos de tarea.

Θ – Acotación exacta

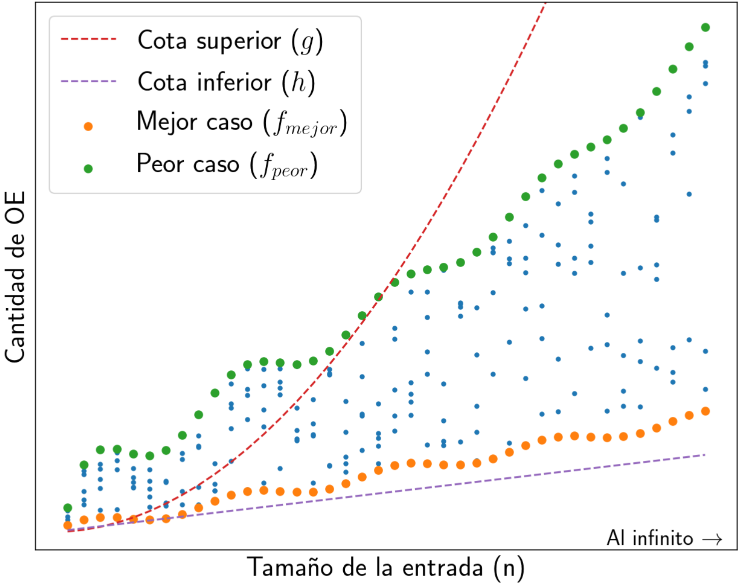
Definición

Sea $g : \mathbb{N} \to \mathbb{R}$. Entonces:

$$\Theta(g) \stackrel{\text{def}}{=} \{f : \mathbb{N} \to \mathbb{R} \mid f \in O(g) \wedge f \in \Omega(g)\}$$

- Si podemos acotar con funciones de la misma familia (solo variando el c) tanto superior como inferiormente tenemos que es Θ.
- Valen las propiedades mencionadas anteriormente para $\mathcal{O}$ y Ω.
- Observación: $\Theta(g) = O(g) \cap \Omega(g)$.

Análisis de un algoritmo – Observaciones



Análisis de un algoritmo

Para analizar algoritmos, podemos:

- **encontrar** las funciones de **peor caso** (f_{peor}) y **mejor caso** (f_{mejor})
- buscar y **proponer funciones para acotar** superior y/o inferiormente.
- **demostrar** que existe un múltiplo de la función propuesta que es efectivamente una cota asintótica.

Demostrando y usando **propiedades** simplificamos el análisis de funciones y por ende los algoritmos.

Primeros pasos – Análisis del **mejor** caso

Precondición: $|A| > 0$ (arreglo no vacío)

$\textsc{BusquedaSecuencial}(A : \texttt{arreglo(nat)}, e : \texttt{nat}) \longrightarrow \texttt{bool}$

```
1: var i : nat, n : nat
2: n ← tam(A)
3: i ← 0                                  ▷ línea 2+3: 2+1
4: mientras i < n ∧ A[i] ≠ e hacer        ▷ evaluar la guarda: 4
5:     i ← i + 1                          ▷ no ejecuta
6: devolver (i < n)                       ▷ 2
```

- $f_{mejor}(n)$ y $f_{peor}(n)$: **cantidad de operaciones** realizadas para un arreglo de tamaño n en mejor y peor caso respectivamente.
- $f_{mejor}(n) = 3 + 4 + 2 = 9 = k_{mejor}$

Análisis del **mejor** caso

$$f_{mejor}(n) = 9 = k_{mejor}$$

- Acotemos inferiormente. $f_{mejor}(n) \in \Omega(1)$
- Acotemos superiormente. $f_{mejor}(n) \in \mathcal{O}(1)$
- $f_{mejor}(n) \in \Theta(1)$

Definición

Sea $g : \mathbb{N} \to \mathbb{R}$. Entonces:

$$\Omega(g) \stackrel{\text{def}}{=} \{f : \mathbb{N} \to \mathbb{R} \mid (\exists\ n_0 \in \mathbb{N},\ c \in \mathbb{R}_{>0})\ \ f(n) \geq c \cdot g(n)\ \ \forall n \geq n_0\}$$

Análisis del **mejor** caso

$$f_{mejor}(n) = 9 = k_{mejor}$$

- Acotemos inferiormente. $f_{mejor}(n) \in \Omega(1)$
- Acotemos superiormente. $f_{mejor}(n) \in \mathcal{O}(1)$
- $f_{mejor}(n) \in \Theta(1)$

Definición

Sea $g : \mathbb{N} \to \mathbb{R}$. Entonces:

$$\mathcal{O}(g) \stackrel{\text{def}}{=} \{f : \mathbb{N} \to \mathbb{R} \mid (\exists\ n_0 \in \mathbb{N},\ c \in \mathbb{R}_{>0})\ \ f(n) \leq c \cdot g(n)\ \ \forall n \geq n_0\}$$

Análisis del **mejor** caso

$$f_{mejor}(n) = 9 = k_{mejor}$$

- Acotemos inferiormente. $f_{mejor}(n) \in \Omega(1)$
- Acotemos superiormente. $f_{mejor}(n) \in \mathcal{O}(1)$
- $f_{mejor}(n) \in \Theta(1)$

Definición

Sea $g : \mathbb{N} \to \mathbb{R}$. Entonces:

$$\Theta(g) \stackrel{\text{def}}{=} \{f : \mathbb{N} \to \mathbb{R} \mid f \in O(g) \land f \in \Omega(g)\}$$

Primeros pasos – Análisis del **peor** caso

Precondición: $|A| > 0$ (arreglo no vacío)

$\textsc{BusquedaSecuencial}(A : \texttt{arreglo(nat)},\ e : \texttt{nat}) \longrightarrow \texttt{bool}$

```
1: var i : nat, n : nat
2: n ← tam(A)
3: i ← 0                                   ▷ línea 2-3: 3
4: mientras i < n ∧ A[i] ≠ e hacer         ▷ ciclo: (n + 1) · 4 + 2 · n
5:     i ← i + 1                           ▷ 2
6: devolver (i < n)                        ▷ 2
```

- $f_{mejor}(n)$ y $f_{peor}(n)$: **cantidad de operaciones** realizadas para un arreglo de tamaño n en mejor y peor caso respectivamente.
- $f_{peor}(n) = 3 + (n+1)\ 4 + 2n + 2 = \ 3 + 4 + 2 + \ (4 + 2)\ n$
  $= 9 \ + \ 6\ n$

Análisis del **peor** caso

$$f_{peor}(n) = 9 \ + \ 6 \ n$$

- Acotemos superiormente. $f_{peor}(n) \in \mathcal{O}(n)$
- Acotemos inferiormente. $f_{peor}(n) \in \Omega(n)$
- $f_{peor}(n) \in \Theta(n)$

Definición

Sea $g : \mathbb{N} \to \mathbb{R}$. Entonces:

$$\mathcal{O}(g) \stackrel{\text{def}}{=} \{f : \mathbb{N} \to \mathbb{R} \mid (\exists\ n_0 \in \mathbb{N},\ c \in \mathbb{R}_{>0})\ \ f(n) \leq c \cdot g(n)\ \ \forall n \geq n_0\}$$

Análisis del **peor** caso

$$f_{peor}(n) = 9 \ + \ 6 \ n$$

- Acotemos superiormente. $f_{peor}(n) \in \mathcal{O}(n)$
- Acotemos inferiormente. $f_{peor}(n) \in \Omega(n)$
- $f_{peor}(n) \in \Theta(n)$

Definición

Sea $g : \mathbb{N} \to \mathbb{R}$. Entonces:

$$\Omega(g) \stackrel{\text{def}}{=} \{f : \mathbb{N} \to \mathbb{R} \mid (\exists \ n_0 \in \mathbb{N}, \ c \in \mathbb{R}_{>0}) \ \ f(n) \geq c \cdot g(n) \ \ \forall n \geq n_0\}$$

Análisis del **peor** caso

$$f_{peor}(n) = 9 + 6\ n$$

- Acotemos superiormente. $f_{peor}(n) \in \mathcal{O}(n)$
- Acotemos inferiormente. $f_{peor}(n) \in \Omega(n)$
- $f_{peor}(n) \in \Theta(n)$

Definición

Sea $g : \mathbb{N} \to \mathbb{R}$. Entonces:

$$\Theta(g) \stackrel{\text{def}}{=} \{f : \mathbb{N} \to \mathbb{R} \mid f \in O(g) \land f \in \Omega(g)\}$$

Análisis de BusquedaSecuencial

$$f_{mejor}(n) \in \Theta(1)$$
$$f_{peor}(n) \in \Theta(n)$$

¿Una función siempre es Θ de su cota?

Solamente si se trata de la cota (superior o inferior) más ajustada, es decir, del mismo orden asintótico que la función.

(Abusos de) notación

En lugar de "$f \in \mathcal{O}(g)$" a veces notamos:

$$f(n) = \mathcal{O}(g(n)) \qquad f(n) = \mathcal{O}(g) \qquad f \in \mathcal{O}(g(n))$$

En lugar de "$f \notin \mathcal{O}(g)$" a veces notamos:

$$f(n) \neq \mathcal{O}(g(n)) \qquad f(n) \neq \mathcal{O}(g) \qquad f \notin \mathcal{O}(g(n))$$

Álgebra de órdenes

1. **Suma.** $\mathcal{O}(f) + \mathcal{O}(g) \overset{\text{def}}{=} \mathcal{O}(f+g) \overset{\text{prop}}{=} \mathcal{O}(\text{máx}\{f, g\})$.
2. **Producto.** $\mathcal{O}(f) \cdot \mathcal{O}(g) \overset{\text{def}}{=} \mathcal{O}(f \cdot g)$.

Vale también para Ω, Θ.

Analicemos nuevamente el peor caso

BusquedaSecuencial(A : `arreglo(nat)`, e : `nat`) $\longrightarrow$ `bool`

1: **var** i : nat, n : nat

2: $n \leftarrow tam(A)$

3: $i \leftarrow 0$ $\quad\triangleright$ línea 2-3: 3

4: **mientras** $i < n \land A[i] \neq e$ **hacer** $\quad\triangleright$ ciclo: $(n+1) \cdot 4 + n \cdot 2$

5: $\quad i \leftarrow i + 1$ $\quad\triangleright$ 2

6: **devolver** $(i < n)$ $\quad\triangleright$ 2

Analicemos nuevamente el peor caso

BusquedaSecuencial(A : `arreglo(nat)`, e : `nat`) $\longrightarrow$ `bool`

1: **var** $i : nat, n : nat$
2: $n \leftarrow tam(A)$
3: $i \leftarrow 0$ ▷ línea 2-3: $\Theta(1)$
4: **mientras** $i < n \wedge A[i] \neq e$ **hacer** ▷ ciclo: $\Theta(n)$
5: $\quad i \leftarrow i + 1$ ▷ $\Theta(1)$
6: **devolver** $(i < n)$ ▷ $\Theta(1)$

$$f_{ciclo}(n) = (n+1) \cdot f_{guarda}(n) + n \cdot f_{interiorc}(n) =$$
$$\Theta(max\{n, 1\}) \cdot \Theta(1) + \Theta(n) \cdot \Theta(1) = \Theta(n) + \Theta(n) = \Theta(n)$$

Finalmente:

$$f_{peor}(n) = \Theta(1) + \Theta(n) + \Theta(1) = \Theta(n)$$

Busqueda secuencial en matrices – Análisis de **peor caso**

Precondición: M es matriz cuadrada de **n** filas por **n** columnas, $n > 0$.

BusquedaSecMatriz(M : `arreglo(arreglo(nat))`, e : `nat`)

1: **var** i : *nat*, j : *nat*, *encontrado* : *bool*, n : *nat*
2: $n \leftarrow tam(M[0])$
3: $i \leftarrow 0$
4: $encontrado \leftarrow false$ ▷ líneas 2 a 4: $\Theta(1)$
5: **mientras** $i < n \ \wedge \neg encontrado$ **hacer** ▷ ciclo: $n \cdot \Theta(n) = \Theta(n^2)$
6: $\quad encontrado = $ busquedaSecuencial(M[i], e) ▷ $\Theta(n)$
7: $\quad i \leftarrow i + 1$ ▷ $\Theta(1)$
8: **devolver** *encontrado* ▷ $\Theta(1)$

Abstraemos y reusamos la función y análisis anterior.

$$f_{peor}(n) = \Theta(1) + \Theta(n^2) + \Theta(1) = \Theta(n^2)$$

Propiedades – $\diamondsuit$ es "comodín" de $\mathcal{O}, \Omega, \Theta$

1. Toda f cumple $f \in \diamondsuit(f)$. Reflexiva
2. $f \in \diamondsuit(g) \implies k \cdot f \in \diamondsuit(g)$ (con k cte.)
3. Regla de la suma:
   $f_1 \in \diamondsuit(g) \wedge f_2 \in \diamondsuit(h) \implies f_1 + f_2 \in \diamondsuit(g+h) = \diamondsuit(\text{máx}\{g, h\})$
4. Regla del producto:
   $f_1 \in \diamondsuit(g) \wedge f_2 \in \diamondsuit(h) \implies f_1 \cdot f_2 \in \diamondsuit(g \cdot h)$

3 y 4 corresponden al **álgebra de órdenes**. Además 4 implica 2.

- $f \in \diamondsuit(g) \wedge g \in \diamondsuit(h) \implies f \in \diamondsuit(h)$ Transitiva
- $f \in \diamondsuit(g) \implies \diamondsuit(f) \subseteq \diamondsuit(g)$
- $\diamondsuit(f) = \diamondsuit(g) \iff f \in \diamondsuit(g) \wedge g \in \diamondsuit(f)$

  Como $f \in \Theta(g) \implies g \in \Theta(f)$ Simétrica
  - $\Theta(f) = \Theta(g) \iff f \in \Theta(g)$

Ejercicios

Decidir si son verdaderas o falsas y justificar:

1. $2^n = \mathcal{O}(1)$
2. Dados $i, j \in \mathbb{N}$ fijos se tiene que $i \cdot n = \mathcal{O}(j \cdot n)$.
3. $n + \log^2 n \in \mathcal{O}(n + \log n)$
4. $\Omega(n) \subset \mathcal{O}(n^2)$
5. $\mathcal{O}(n^2) \subset \Omega(n)$
6. Si $f(n) = \mathcal{O}(n)$ entonces $2^{f(n)} = \mathcal{O}(2^n)$
7. $\Theta(n^2) = \mathcal{O}(n^2) \setminus \mathcal{O}(n)$
8. $\Theta(n \log(n)) = \Theta(\log(n!))$

1) $2^n \in \mathcal{O}(1)$ **Falso**

$$
\begin{aligned}
2^n \in \mathcal{O}(1) &\Leftrightarrow (\exists\ n_0 \in \mathbb{N},\ c \in \mathbb{R}_{>0})\ \ 2^n \leq c \cdot 1 \ \ \forall n \geq n_0 \\
&\Leftrightarrow (\exists\ n_0,\ c)\ \ n \cdot \log(2) \leq \log(c)\ \ \forall n \geq n_0 \\
&\Leftrightarrow (\exists\ n_0,\ c)\ \ n \leq \frac{\log(c)}{\log(2)}\ \ \forall n \geq n_0 \ \textbf{Falso}
\end{aligned}
$$

Para cualquier par de c, n_0 la desigualdad deja de cumplirse para $n = \left\lfloor \frac{\log(c)}{\log(2)} \right\rfloor + 1$.

Ejercicios

Decidir si son verdaderas o falsas y justificar:

1. $2^n = \mathcal{O}(1)$ **F**
2. Dados $i, j \in \mathbb{N}$ fijos se tiene que $i \cdot n = \mathcal{O}(j \cdot n)$.
3. $n + \log^2 n \in \mathcal{O}(n + \log n)$
4. $\Omega(n) \subset \mathcal{O}(n^2)$
5. $\mathcal{O}(n^2) \subset \Omega(n)$
6. Si $f(n) = \mathcal{O}(n)$ entonces $2^{f(n)} = \mathcal{O}(2^n)$
7. $\Theta(n^2) = \mathcal{O}(n^2) \setminus \mathcal{O}(n)$
8. $\Theta(n \log(n)) = \Theta(\log(n!))$

2) $in = \mathcal{O}(jn)$

Por definición de $\mathcal{O}$, q.v.q.

$$in \in \mathcal{O}(jn) \Leftrightarrow (\exists\ n_0,\ c)\ \ \underbrace{in \leq cjn}_{\Downarrow}\ \ \forall n \geq n_0$$

$$\Leftrightarrow in \leq cjn \underset{n \neq 0}{\Leftrightarrow} i \leq cj$$

$$\Leftrightarrow \frac{i}{j} \leq c$$

¿Por qué puedo tomar $n > 0$?

Propiedad

$$\forall k \in \mathbb{N} \qquad k \cdot f(n) \in \mathcal{O}(f(n))$$

Ejercicios

Decidir si son verdaderas o falsas y justificar:

1. $2^n = \mathcal{O}(1)$ **F**
2. Dados $i, j \in \mathbb{N}$ fijos se tiene que $i \cdot n = \mathcal{O}(j \cdot n)$. **V**
3. $n + \log^2 n \in \mathcal{O}(n + \log n)$
4. $\Omega(n) \subset \mathcal{O}(n^2)$
5. $\mathcal{O}(n^2) \subset \Omega(n)$
6. Si $f(n) = \mathcal{O}(n)$ entonces $2^{f(n)} = \mathcal{O}(2^n)$
7. $\Theta(n^2) = \mathcal{O}(n^2) \setminus \mathcal{O}(n)$
8. $\Theta(n \log(n)) = \Theta(\log(n!))$

3) $n + \log^2 n \in \mathcal{O}(n + \log n)$ - Ideas

Sabemos que

$$\mathcal{O}(n + \log(n)) = \mathcal{O}(max_{\infty}\{n, \log(n)\}) = \mathcal{O}(n)$$

Como $log(n) \in \mathcal{O}(n)$ (P.2 - Ej.2)

$$¿\text{Vale } n + \log^2 n \in \mathcal{O}(n)?$$

Propiedad

Dadas las funciones $f,\ g : \mathbb{N} \to \mathbb{R}$. Si existe

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

Podemos decir:

- Si $k \neq 0$ y $k < \infty$ entonces $\Theta(f) = \Theta(g)$.
- Si $k = 0$ entonces:
  - $f \in \mathcal{O}(g) \wedge g \notin \mathcal{O}(f)$ i.e. $\mathcal{O}(f) \subset \mathcal{O}(g)$ estrictamente.
  - $g \in \Omega(f) \wedge f \notin \Omega(g)$ i.e. $\Omega(g) \subset \Omega(f)$ estrictamente.
  - $\Theta(f) \neq \Theta(g)$.

TAREA: Convencerse de que los tres items para $k = 0$ son equivalentes.

3) $n + \log^2 n \in \mathcal{O}(n)$ **Verdadero**

$$\lim_{n\to\infty} \frac{n+\log^2 n}{n} \underset{l'h}{=} \lim_{n\to\infty} 1 + \frac{2\log n}{n}$$

$$= \lim_{n\to\infty} 1 + \frac{2\log n}{n} = 1$$

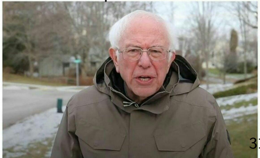
Veamos por separado el segundo término:

$$\underset{l'h}{=} \lim_{n\to\infty} \frac{2}{n} = 0$$

Luego $n + \log^2 n \in \mathcal{O}(n)$.

not getting a simplified form after using L'Hopital once

me to L'Hopital:



Ejercicios

Decidir si son verdaderas o falsas y justificar:

1. $2^n = \mathcal{O}(1)$ **F**
2. Dados $i, j \in \mathbb{N}$ fijos se tiene que $i \cdot n = \mathcal{O}(j \cdot n)$. **V**
3. $n + \log^2 n \in \mathcal{O}(n + \log n)$ **V**
4. $\Omega(n) \subset \mathcal{O}(n^2)$
5. $\mathcal{O}(n^2) \subset \Omega(n)$
6. Si $f(n) = \mathcal{O}(n)$ entonces $2^{f(n)} = \mathcal{O}(2^n)$
7. $\Theta(n^2) = \mathcal{O}(n^2) \setminus \mathcal{O}(n)$
8. $\Theta(n \log(n)) = \Theta(\log(n!))$

4 y 5) Ideas

¿Es cierto que toda función de $\Omega(n)$ está en $\mathcal{O}(n^2)$?
¿Qué pasa con n^3?

¿Es cierto que toda función de $\mathcal{O}(n^2)$ está en $\Omega(n)$?
¿Qué pasa con $\log(n)$?

Ejercicios

Decidir si son verdaderas o falsas y justificar:

1. $2^n = \mathcal{O}(1)$ **F**
2. Dados $i, j \in \mathbb{N}$ fijos se tiene que $i \cdot n = \mathcal{O}(j \cdot n)$. **V**
3. $n + \log^2 n \in \mathcal{O}(n + \log n)$ **V**
4. $\Omega(n) \subset \mathcal{O}(n^2)$ **F**
5. $\mathcal{O}(n^2) \subset \Omega(n)$ **F**
6. Si $f(n) = \mathcal{O}(n)$ entonces $2^{f(n)} = \mathcal{O}(2^n)$
7. $\Theta(n^2) = \mathcal{O}(n^2) \setminus \mathcal{O}(n)$
8. $\Theta(n \log(n)) = \Theta(\log(n!))$

6) Si $f(n) = \mathcal{O}(n)$ entonces $2^{f(n)} = \mathcal{O}(2^n)$ **Falso**

- **V:** entonces para **toda** $f \in \mathcal{O}(n) \rightarrow 2^{f(n)} = \mathcal{O}(2^n)$
- **F:** entonces para **alguna** $f \in \mathcal{O}(n) \rightarrow 2^{f(n)} \neq \mathcal{O}(2^n)$

$$\mathcal{O}(n) = \{\, 1, 3, 10, \ldots, \log_b(n), \ldots, \log_b^2(n), \ldots, n, n+5, , 3n, \ldots\}$$

- **3:** $2^3 \in \mathcal{O}(2^n)$
- **$\log_2(n)$:** $2^{log_2(n)} = n \in \mathcal{O}(2^n)$

6) Si $f(n) = \mathcal{O}(n)$ entonces $2^{f(n)} = \mathcal{O}(2^n)$ **Falso**

- **V:** entonces para **toda** $f \in \mathcal{O}(n) \to 2^{f(n)} = \mathcal{O}(2^n)$
- **F:** entonces para **alguna** $f \in \mathcal{O}(n) \to 2^{f(n)} \neq \mathcal{O}(2^n)$

$$\mathcal{O}(n) = \{1, , 10, \ldots, , \ldots, \log_b^2(n), \ldots, n, n+5, 2n, 3n, \ldots\}$$

- **2n:** ¿$2^{2n} \stackrel{?}{=} \mathcal{O}(2^n)$?

$$\lim_{n\to\infty} \frac{\overbrace{2^n}^{g(n)}}{\underbrace{2^{2n}}_{h(n)}} = \lim_{n\to\infty} \frac{\cancel{2^n}}{\cancel{2^n} \cdot 2^n} = \lim_{n\to\infty} \frac{1}{2^n} = 0$$

Caso $k = 0$**:** $h \notin \mathcal{O}(g)$. 2n es un contraejemplo. Es **Falso**

Ejercicios

Decidir si son verdaderas o falsas y justificar:

1. $2^n = \mathcal{O}(1)$ **F**
2. Dados $i, j \in \mathbb{N}$ fijos se tiene que $i \cdot n = \mathcal{O}(j \cdot n)$. **V**
3. $n + \log^2 n \in \mathcal{O}(n + \log n)$ **V**
4. $\Omega(n) \subset \mathcal{O}(n^2)$ **F**
5. $\mathcal{O}(n^2) \subset \Omega(n)$ **F**
6. Si $f(n) = \mathcal{O}(n)$ entonces $2^{f(n)} = \mathcal{O}(2^n)$ **F**
7. $\Theta(n^2) = \mathcal{O}(n^2) \setminus \mathcal{O}(n)$
8. $\Theta(n \log(n)) = \Theta(\log(n!))$

7) $\Theta(n^2) = \mathcal{O}(n^2) \setminus \mathcal{O}(n)$

$$\Theta(n^2) = \mathcal{O}(n^2) \cap \Omega(n^2)$$

$$\mathcal{O}(n) = \{1, 3, 5, 10, \ldots, \log_b(n), \ldots, \log_b^2(n), \ldots, n, n{+}5, 2n, 3n, \ldots\}$$

$$\mathcal{O}(1) \subset \mathcal{O}(\log(n)) \subset \mathcal{O}(\log^2(n)) \subset \mathcal{O}(\sqrt{n}) \subset \mathcal{O}(n) \subset \mathcal{O}(n^2) \subset \cdots$$

$$\mathcal{O}(n^2) = \mathcal{O}(n) \cup \underbrace{\{\ldots, n^{3/2}, \ldots, \frac{n^2}{2} + 1, n^2, 3n^2 + 3, \ldots\}}_{¿\Theta(n^2)?}$$

$$\lim_{n\to\infty} \frac{n^{3/2}}{n^2} = \lim_{n\to\infty} \frac{1}{\sqrt{n}} = 0$$

Caso $k = 0$: $n^{3/2} \notin \Theta(n^2)$

Ejercicios

Decidir si son verdaderas o falsas y justificar:

1. $2^n = \mathcal{O}(1)$ **F**
2. Dados $i, j \in \mathbb{N}$ fijos se tiene que $i \cdot n = \mathcal{O}(j \cdot n)$. **V**
3. $n + \log^2 n \in \mathcal{O}(n + \log n)$ **V**
4. $\Omega(n) \subset \mathcal{O}(n^2)$ **F**
5. $\mathcal{O}(n^2) \subset \Omega(n)$ **F**
6. Si $f(n) = \mathcal{O}(n)$ entonces $2^{f(n)} = \mathcal{O}(2^n)$ **F**
7. $\Theta(n^2) = \mathcal{O}(n^2) \setminus \mathcal{O}(n)$ **F**
8. $\Theta(n \log(n)) = \Theta(\log(n!))$ **Veamos que es V**

8) ¿$\Theta(n\log(n)) = \Theta(\log(n!))$? **Verdadero**

Para esto, basta probar que,

$$n\log(n) \in \Theta(\log(n!))$$

o que

$$\log(n!) \in \Theta(n\log(n))$$

Veamos lo último, que consiste en:

$$\log(n!) \in \mathcal{O}(n\log(n)) \;\wedge\; \log(n!) \in \Omega(n\log(n))$$

8) $\log(n!) \in \mathcal{O}(n\log(n))$

Por propiedades: $\log(n!) = \log(\prod_{i=1}^{n}(i)) = \sum_{i=1}^{n}\log(i)$

Tenemos la siguiente cota:

$$\log(n!) = \sum_{i=1}^{n}\log(i) \quad \leq \quad n \cdot \max_{1\leq i\leq n}\{\log(i)\} = n\log(n)$$

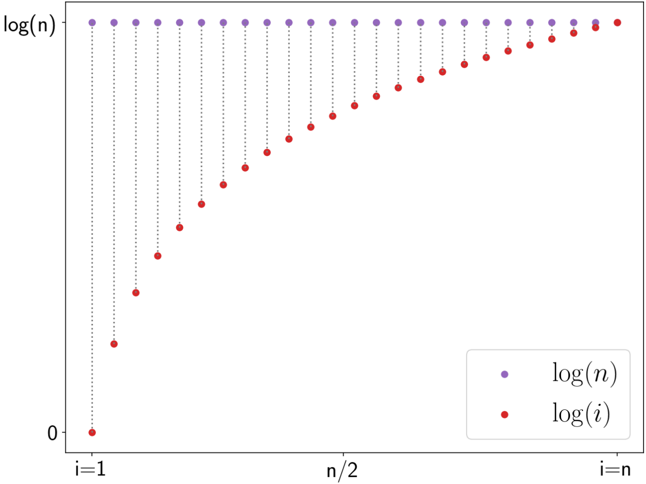
Entonces, tomando la definición de $\mathcal{O}(n\log(n))$:

$$\log(n!) \underset{cota}{\leq} n\log(n) \leq c \cdot n\log(n)$$

Tomamos $c = 1$ y vale para todo n natural en particular para $n_0 = 1$.

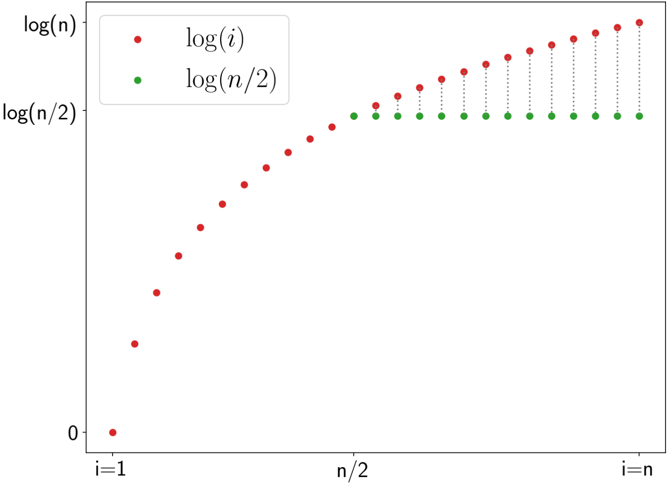
8) $\log(n!) \in \mathcal{O}(n \log(n))$

$$\log(n!) = \sum_{i=1}^{n} \log(i)$$



$$\log(n!) \leq n \log(n)$$

8) $\log(n!) \in \Omega(n \log(n))$



$$\log(n!) \geq \frac{n}{2} \log(\frac{n}{2})$$

8) $\log(n!) \in \Omega(n\log(n))$

Miremos la siguiente cota:

$$\log(n!) = \log(1)+\log(2)+\cdots+\log\left(\frac{n}{2}\right)+\log\left(\frac{n}{2}+1\right)+\cdots+\log(n)$$

$$\geq \frac{n}{2}\log\left(\frac{n}{2}\right) = \frac{n}{2}(\log(n)-\log(2)) = \frac{n}{2}\log(n) - \frac{n}{2}\log(2)$$

$$\underset{q.v.q.}{\geq} \frac{n}{2}\log(n) - \frac{n}{4}\log(n) = \frac{n}{4}\log(n)$$

Podríamos tomar $c = 1/4$. Pero pusimos una condición sobre n. Para encontrar un n_0 recordemos que habíamos pedido

$$\frac{n}{2}\log(2) \leq \frac{n}{4}\log(n)$$

8) $\log(n!) \in \Omega(n\log(n))$ – demo 2^{do} galerazo

Recordemos que habíamos pedido $\frac{n}{2}\log(2) \leq \frac{n}{4}\log(n)$.
¿Para que n vale?

- $\frac{n}{2}\log(2) \overset{?}{\leq} \frac{n}{4}\log(n)$
- sii $\log(2) \leq \frac{1}{2}\log(n)$
- sii $2 \leq \log_2(n)$
- sii $2^2 \leq 2^{\log_2(n)}$
- sii $4 \leq n$

Vale para $n \geq 4$

$\log(n!) \in \Omega(n\log(n))$ – Final

Miremos la siguiente cota:

$$\log(n!) = \log(1) + \log(2) + \cdots + \log\left(\frac{n}{2}\right) + \log\left(\frac{n}{2}+1\right) + \cdots + \log(n)$$

$$\geq \frac{n}{2}\log\left(\frac{n}{2}\right) = \frac{n}{2}(\log(n) - \log(2)) = \frac{n}{2}\log(n) - \frac{n}{2}\log(2)$$

$$\underset{q.v.q.}{\geq} \frac{n}{2}\log(n) - \frac{n}{4}\log(n) = \frac{n}{4}\log(n)$$

Tomando $c = 1/4$ y $n_0 = 4$ probamos que vale la definición.

Ejercicios

Decidir si son verdaderas o falsas y justificar:

1. $2^n = \mathcal{O}(1)$ **F**
2. Dados $i, j \in \mathbb{N}$ fijos se tiene que $i \cdot n = \mathcal{O}(j \cdot n)$. **V**
3. $n + \log^2 n \in \mathcal{O}(n + \log n)$ **V**
4. $\Omega(n) \subset \mathcal{O}(n^2)$ **F**
5. $\mathcal{O}(n^2) \subset \Omega(n)$ **F**
6. Si $f(n) = \mathcal{O}(n)$ entonces $2^{f(n)} = \mathcal{O}(2^n)$ **F**
7. $\Theta(n^2) = \mathcal{O}(n^2) \setminus \mathcal{O}(n)$ **F**
8. $\Theta(n \log(n)) = \Theta(\log(n!))$ **Veamos que es V**

Ejercicio de parcial (items c y d)

Parcial 2^{do} cuatrimestre 2019

Sea A un arreglo de enteros. $n = tam(A)$.
Dado el algoritmo de la página siguiente, determinar

(c) ... una $f : \mathbb{N} \to \mathbb{R}_{>0}$ tal que $T_{\text{mejor}}(n) \in \Theta(f(n))$.

(d) ... una $g : \mathbb{N} \to \mathbb{R}_{>0}$ tal que $T_{\text{peor}}(n) \in \Theta(g(n))$.

Notar que la primera posición del arreglo es $A[1]$.

Observación: Consideramos que las operaciones de suma, resta, multiplicación y comparación entre naturales son operaciones elementales.

ALGORITMOQUEHACEALGO(A : `arreglo(nat)`)

```
i ← 1; j ← 1
suma ← 1; count ← 0
mientras i ≤ tam(A) hacer
    si i ≠ A[i] entonces
        count ← count + 1
    j ← 1
    mientras j ≤ count hacer
        k ← 1
        mientras k ≤ tam(A) hacer
            suma ← suma + A[k]
            k ← k · 2
        j ← j + 1
    i ← i + 1
devolver suma
```

Observación: $A[1]$ es la primera posición del arreglo.

- ¿Cuándo es mejor caso?
- ¿Cuándo es peor caso?

ALGQUEHACEALGO(A)

```
i ← 1; j ← 1
suma ← 1; count ← 0
mientras i ≤ tam(A) hacer
    si i ≠ A[i] entonces
        count ← count + 1
    j ← 1
    mientras j ≤ count hacer
        k ← 1
        mientras k ≤ tam(A) hacer
            suma ← suma + A[k]
            k ← k · 2
        j ← j + 1
    i ← i + 1
devolver suma
```

$A[1]$ es la primera posición del arreglo.

Mejor caso: $A[i] = i$ siempre

$$T_{mejor} = \sum_{i=1}^{n} (\Theta(1)) = \Theta(n)$$

AlgQueHaceAlgo(A)

```
i ← 1; j ← 1
suma ← 1; count ← 0
mientras i ≤ tam(A) hacer
    si i ≠ A[i] entonces
        count ← count + 1
    j ← 1
    mientras j ≤ count hacer
        k ← 1
        mientras k ≤ tam(A) hacer
            suma ← suma + A[k]
            k ← k · 2
        j ← j + 1
    i ← i + 1
devolver suma
```

$A[1]$ es la primera posición del arreglo.

Peor caso: $A[i] \neq i$ siempre

$$
\begin{aligned}
T_{peor}(n) &= \sum_{i=1}^{n} \sum_{j=1}^{i} \log n \\
&= \log n \sum_{i=1}^{n} i \\
&= \log n \frac{n(n+1)}{2} \\
&= \frac{\log n}{2}(n^2 + n)
\end{aligned}
$$

$$T_{peor} \in \Theta(n^2 \log n)$$

AlgQueHaceAlgo(A)

```
i ← 1; j ← 1
suma ← 1; count ← 0
mientras i ≤ tam(A) hacer
    si i ≠ A[i] entonces
        count ← count + 1
    j ← 1
    mientras j ≤ count hacer
        k ← 1
        mientras k ≤ tam(A) hacer
            suma ← suma + A[k]
            k ← k · 2
        j ← j + 1
    i ← i + 1
devolver suma
```

$A[1]$ es la primera posición del arreglo.

¿Preguntas?

¡Anótenlas y las vemos en clase!