

Notación Asintótica

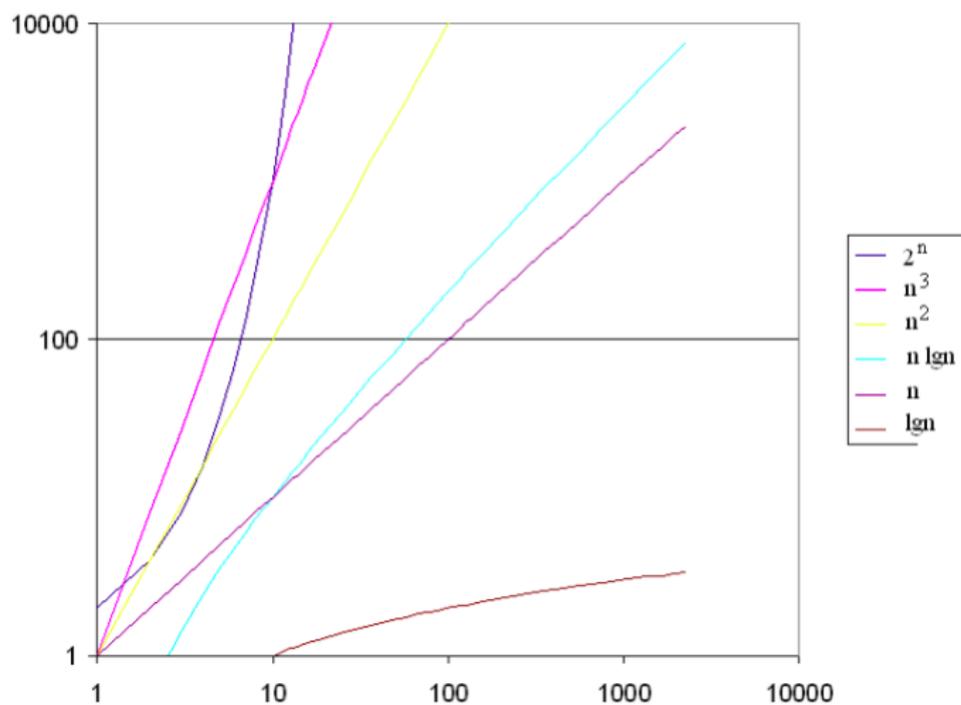
Hernán Gómez

Universidad del Valle
(Facultad de Ingeniería)

Febrero, 2020



Crecimiento de funciones



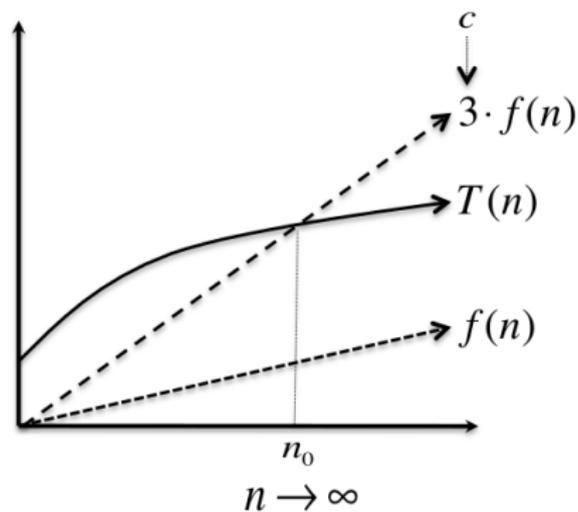
Notación Asintótica

- Es una forma de acotar un conjunto de funciones
- Sirve para analizar un algoritmo en su complejidad temporal
- Sirve para limitar y caracterizar un algoritmo



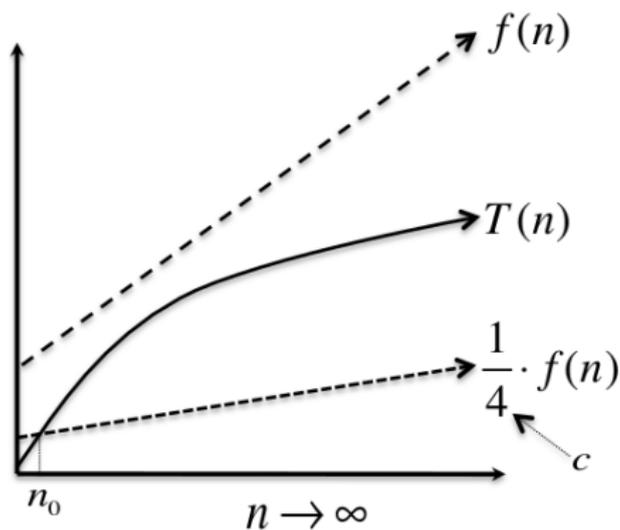
Notación Big O

$T(n)$ es $O(f(n))$ si y solo si hay una constantes positivas c y n tal que
 $\forall_n n \geq n_0 \quad T(n) \leq c \cdot f(n)$



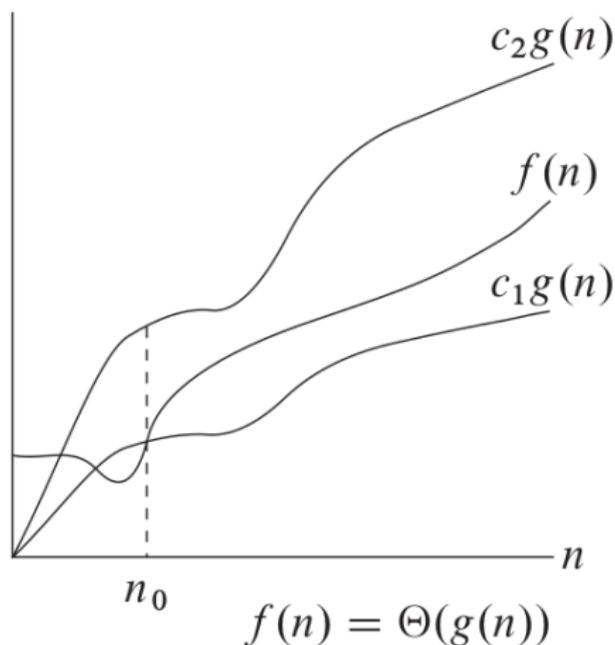
Notación Big Omega

$T(n)$ es $\Omega(f(n))$ si y solo si hay una constantes positivas c y n tal que $\forall_n n \geq n_0 \quad T(n) \geq c \cdot f(n)$



Notación Theta

$T(n)$ es $\theta(f(n))$ si y solo si hay una constantes positivas c_1 , c_2 y n tal que $\forall_n n \geq n_0 \quad c_1 \cdot f(n) \leq T(n) \leq c_2 \cdot f(n)$



Notaciones Small Omega y Small O

- Small Omega $T(n)$ es $\omega(f(n))$ si y solo si hay una constantes positivas c ,y n tal que $n_0 > 0$, $T(n) > c.f(n)$
- Small o(op) $T(n)$ es $o(f(n))$ si y solo si hay una constantes positivas c ,y n tal que $n_0 > 0$, $T(n) < c.f(n)$

Ejercicio # 1

Searching One Array

Input: array A of n integers, and an integer t .

Output: Whether or not A contains t .

```
for  $i := 1$  to  $n$  do
  if  $A[i] = t$  then
    return TRUE
return FALSE
```

Figura: Ejemplo 1; Búsqueda de un entero t en un arreglo A



Ejercicio # 2

Searching Two Arrays

Input: arrays A and B of n integers each, and an integer t .

Output: Whether or not A or B contains t .

```
for  $i := 1$  to  $n$  do
  if  $A[i] = t$  then
    return TRUE
for  $i := 1$  to  $n$  do
  if  $B[i] = t$  then
    return TRUE
return FALSE
```

Figura: Ejemplo 2; búsqueda de un número t en 2 arreglos de tamaño n



Ejercicio # 3

Checking for a Common Element

Input: arrays A and B of n integers each.

Output: Whether or not there is an integer t contained in both A and B .

```
for  $i := 1$  to  $n$  do
  for  $j := 1$  to  $n$  do
    if  $A[i] = B[j]$  then
      return TRUE
return FALSE
```

Figura: Ejemplo 3; encontrar en 2 arreglos A y B si contienen un dato en común



Ejercicio # 4

Checking for Duplicates

Input: array A of n integers.

Output: Whether or not A contains an integer more than once.

```
for  $i := 1$  to  $n$  do
  for  $j := i + 1$  to  $n$  do
    if  $A[i] = A[j]$  then
      return TRUE
return FALSE
```

Figura: Ejemplo 4; verificar si un arreglo A hay datos repetidos

