

Complexidade assintótica

Jorge E. S. Souza





Complexidade assintótica

- Quando observamos tamanhos de entrada grandes o suficiente para tornar relevante apenas a ordem de crescimento do tempo de execução, estamos estudando a eficiência assintótica dos algoritmos;
- Ou seja, estamos preocupados com a maneira como o tempo de execução de um algoritmo aumenta, à medida que o tamanho da entrada da entrada aumenta INDEFINIDAMENTE;
- Em geral, um algoritmo que é assintoticamente mais eficiente será a melhor escolha para todas as entradas, exceto as muito pequenas.



Complexidade assintótica

- Em 1965 Jack Edmonds introduz a idéia de Complexidade Assintótica.
 - A função $T(x)$ é chamada de Complexidade Local.
 - A Complexidade Assintótica fornece limites para $T(x)$.





Para que serve?

Comparação entre complexidades

- Um problema pode ter mais de um algoritmo para resolvê-lo. Qual deles escolher?
- De maneira geral, consideremos um conjunto A de algoritmos para um dado problema.
- A cada algoritmo $a \in A$ temos associado a função de avaliação $aval$.
- Frequentemente, interessa-nos identificar um algoritmo ótimo, ou seja, um algoritmo o , tal que $aval(o)$ é melhor do que $aval(a)$ para todo $a \in A$.





Complexidade assintótica

⇒ A complexidade assintótica é definida pelo crescimento da complexidade para entradas suficientemente grandes;

⇒ Um algoritmo assintoticamente mais eficiente é melhor para todas as entradas, exceto para entradas relativamente pequenas;

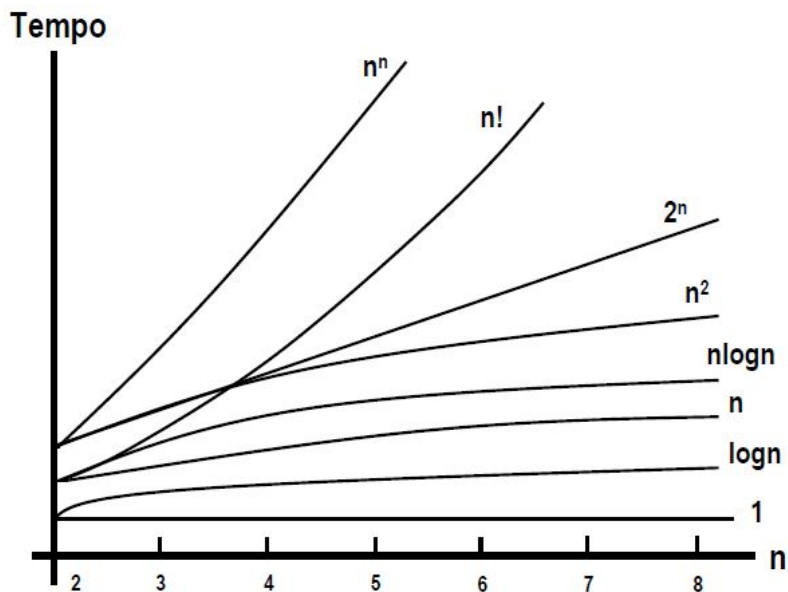
⇒ Esta complexidade é comumente denotada através de uma COTA;

⇒ As cotas são maneiras de reduzir detalhes realçando apenas os aspectos relevantes de eficiência.





Complexidades



- $O(1)$ constante
- $O(\log n)$ logarítmica
- $O(n)$ linear
- $O(n \log n)$ log linear
- $O(n^2)$ quadrática
- $O(n^3)$ cúbica
- $O(2^n)$ exponencial



Notações

⇒ A análise assintótica é um conjunto de notações que fornece uma maneira de formular conclusões sobre a comparação da complexidade de algoritmos;

⇒ são elas:

1. notação **O** → limite assintótico superior;
2. notação **Ω** → limite assintótico inferior;
3. notação **Θ** → limite assintoticamente restrito;
4. notação **o** → limite assintoticamente superior não restrito;
5. notação **ω** → limite assintoticamente inferior não restrito;

$$\lim_{x \rightarrow \infty} \frac{x + \frac{\sin(10x)}{x}}{x} = \lim_{x \rightarrow \infty} \frac{x}{x} + \frac{\sin(10x)}{x^2} \therefore$$
$$\lim_{x \rightarrow \infty} \frac{x + \frac{\sin(10x)}{x}}{x} = \lim_{x \rightarrow \infty} 1 + 0 = 1$$

Notação O, limite superior

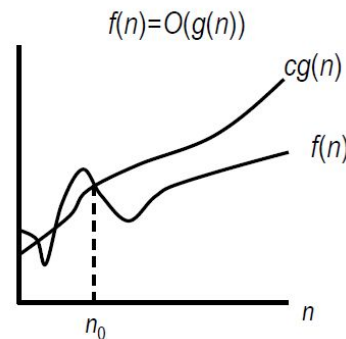
Dadas funções assintoticamente não negativas **f** e **g**, dizemos que **f** está na ordem **O** de **g** e escrevemos **f = O(g)** se:

ordem n^2 , ordem n^3

repare que aqui realmente é uma igualdade

$$f(n) \leq c \cdot g(n)$$

para algum **c** positivo e para todo **n** suficientemente grande.





Notação O

A medida de custo ou medida de complexidade relata o crescimento assintótico da operação considerada.

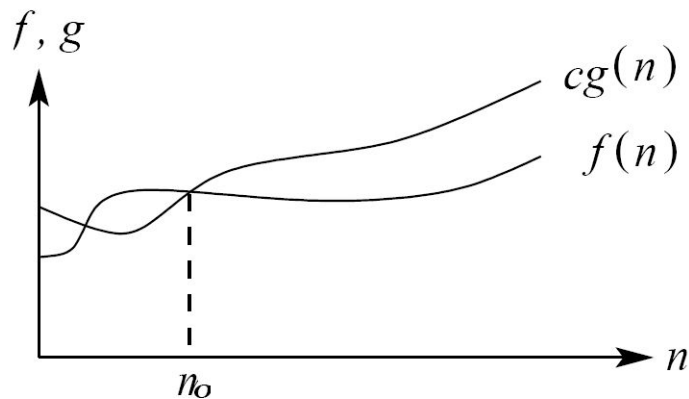
⇒ Definição: Uma função $g(n)$ domina assintoticamente outra função $f(n)$ se existem duas constantes positivas c e n_0

⇒ tais que, para qualquer

$$n \geq n_0,$$

⇒ temos

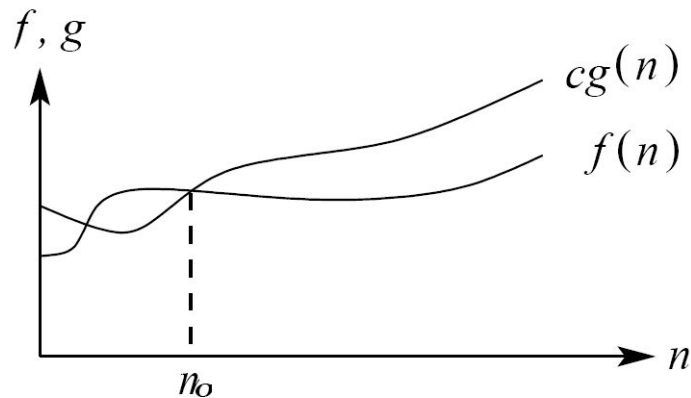
$$f(n) \leq c \cdot g(n)$$





Notação O

⇒ Limite assintótico superior



$$O(g(n)) = \{f(n): \exists c \text{ e } n_0 > 0 \mid 0 \leq f(n) \leq c.g(n) \quad \forall n \geq n_0\}$$



Formal ou informal

Escrevemos $g(n) = O(f(n))$ ou $g(n) \in O(f(n))$ para expressar que $f(n)$ domina assintoticamente $g(n)$.

Encontramos leituras nos modos:

- $g(n)$ é da ordem no máximo $f(n)$; // formal
- $g(n)$ é O de $f(n)$; // informal
- $g(n)$ é igual a O de $f(n)$; // informal
- $g(n)$ pertence a O de $f(n)$; // formal





Exemplo 1:

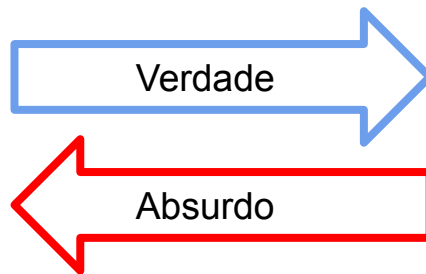
▫ $f(n) = n$

▫ $g(n) = n+34$

▫ $g(n) \in O(f(n))$???

▫ ou

▫ $n+34 \in O(n)$???





Exemplo 1:

▫ $f(n) = n$

▫ $g(n) = n+34$

▫ $g(n) \in O(f(n))$???

▫ ou

▫ $n+34 \in O(n)$???

$$n + 34 \leq c \cdot n, \quad c = 35 \text{ e } n_0 = 1.$$

Testando para $n = 1$

$$1 + 34 \leq 35 * 1 \Rightarrow 35 \leq 35 \therefore \text{verdade}$$

Daria para baixar o valor de c ???

Qual é o custo disso ??? sim



Exemplo 1:

▫ $f(n) = n$

▫ $g(n) = n+34$

▫ $g(n) \in O(f(n))$???

▫ ou

▫ $n+34 \in O(n)$???

$$n + 34 \leq c \cdot n, \quad c = 2 \text{ e } n_0 = ?.$$

Testando para $n = 1$

$$1 + 34 \leq 2 * 1 \Rightarrow 35 \leq 2$$

∴ falso para esse valor de n

Mas quando $n \rightarrow \infty$, se mantém falso?

o que acontece ??? não se mantem falso

Exemplo 1:

▫ $f(n) = n$

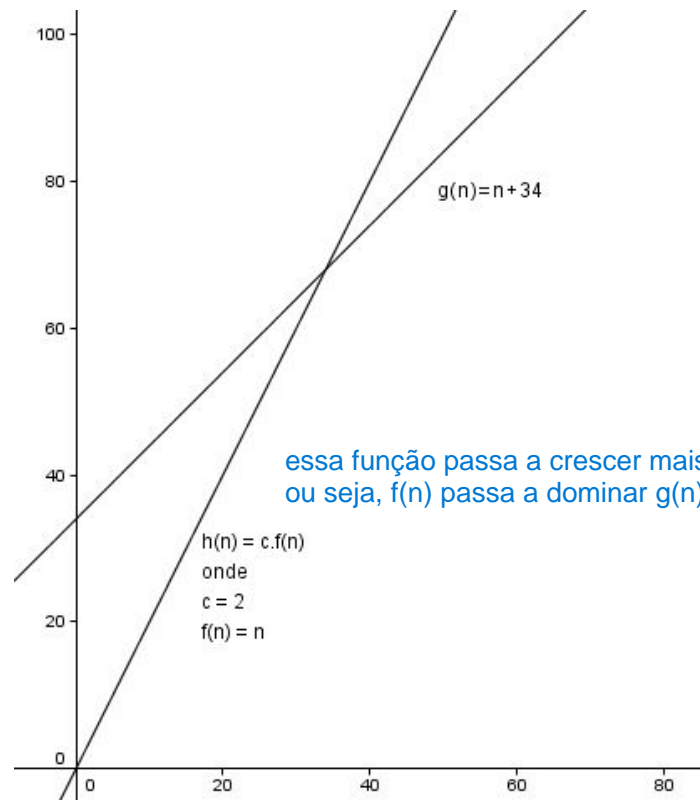
▫ $g(n) = n+34$

▫ $g(n) \in O(f(n))$???

▫ ou

▫ $n+34 \in O(n)$???

sim





Exemplo 2

n^2 é $O(n)$?

não

$$n^2 \leq c \cdot n, \quad c = 1 \text{ e } n_0 = 1.$$

Testando para $n = 1$

$$1^2 \leq 1 * 1 \Rightarrow 1 \leq 1 \quad \therefore \text{verdadeiro}$$

Mas quando $n \rightarrow \infty$, se mantém verdadeiro?

$$n^2 \leq c \cdot n, \quad c = 1 \text{ e } n = 2.$$

$$2^2 \leq 1 * 2 \Rightarrow 4 \leq 2 \quad \therefore \text{Absurdo}$$

Exemplo 2

n^2 é $O(n)$?

Suponha: $\exists c, n_0$

$\forall n \geq n_0$

$$\begin{aligned} n^2 \leq c \cdot n &\Rightarrow \frac{n^2}{n} \leq c \cdot \frac{n}{n} \\ &\Rightarrow n \leq c \end{aligned}$$

Absurdo, pois c é uma constante
e n assume valores até o infinito.

Portanto, $\nexists c, n_0$

$$n^2 \leq c \cdot n, \quad c = 1 \text{ e } n_0 = 1.$$

Testando para $n = 1$

$$1^2 \leq 1 * 1 \Rightarrow 1 \leq 1 \quad \therefore \text{verdadeiro}$$

Mas quando $n \rightarrow \infty$, se mantém verdadeiro?

$$n^2 \leq c \cdot n, \quad c = 1 \text{ e } n = 2.$$

$$2^2 \leq 1 * 2 \Rightarrow 4 \leq 2 \quad \therefore \text{Absurdo}$$



Operações básicas

$$f(n) = O(f(n))$$

$$c \times O(f(n)) = O(f(n)) \quad c = \text{constante}$$

$$O(f(n)) + O(f(n)) = O(f(n))$$

$$O(O(f(n))) = O(f(n))$$

$$O(f(n)) + O(g(n)) = O(\max(f(n), g(n)))$$

$$O(f(n))O(g(n)) = O(f(n)g(n))$$

$$f(n)O(g(n)) = O(f(n)g(n))$$



Outras notações

Assim como a notação O fornece uma maneira assintótica de dizer que uma função é “menor ou igual a” outra, existem outras notação que fornecem outras conclusões sobre a complexidade de algoritmos;

⇒ são elas:

1. notação Ω → limite assintótico inferior;
2. notação Θ → limite assintoticamente restrito;
3. notação o → limite assintoticamente superior não restrito;
4. notação ω → limite assintoticamente inferior não restrito;



Notação Ω

A notação Ω é bem parecida com a notação O ;

$\Rightarrow O$ define um limite assintótico superior, e;

$\Rightarrow \Omega$ define um limite assintótico inferior.

Exemplos:

$$n^4 \in \Omega(n^3)$$

$$n \in \Omega(1)$$

$$3 \cdot \log(n) \in \Omega(\log(n))$$

$$1 \in \Omega(1)$$

$$n! \in \Omega(2^n)$$

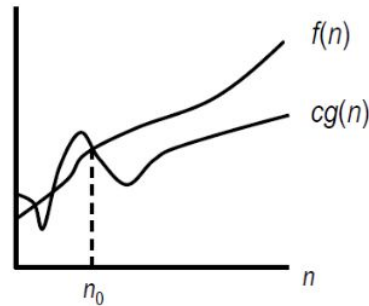


Notação Ω , limite inferior

Dadas funções assintoticamente não negativas **f** e **g**, dizemos que **f** está em Ω de **g** e escrevemos **f** = $\Omega(g)$ se:

$$f(n) \geq c.g(n)$$

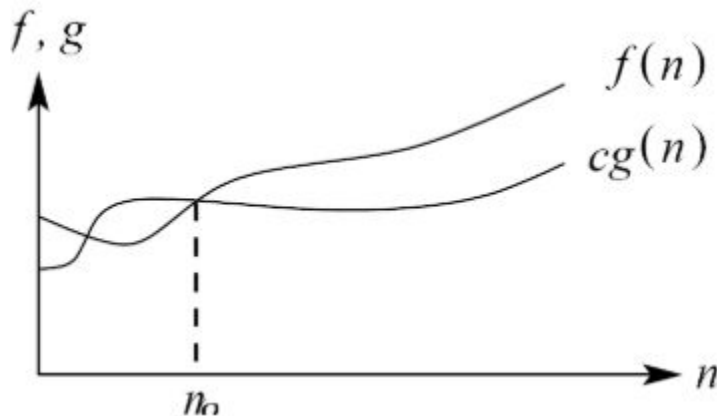
para algum **c** positivo e para todo **n** suficientemente grande.





Notação Ω

⇒ Limite assintótico inferior



$$\Omega(g(n)) = \{f(n): \exists c \text{ e } n_0 > 0 \mid 0 \leq c \cdot g(n) \leq f(n) \quad \forall n \geq n_0\}$$



Notação Ω

Na prática a notação Ω não é vista sozinha em análises de algoritmos;

- Pelo motivo de não interessar para a análise de algoritmos;
- A notação O possui sua importância, pois o programador conclui que seu algoritmo é no máximo tão complexo a uma função.
- Mas no mínimo tão complexo, como a notação Ω descreve, não é importante para conclusões práticas sobre algoritmos.
- Ω vem na maioria das vezes acompanhada a notação Θ ;
- Como um complemento na análise...



Notação Θ

Conhecida também como “limite firme” ou “limite assintoticamente restrito”.

- A notação O , apesar de fornecer informações sobre a complexidade do algoritmo, nem sempre nos revela algo importante;
- Não faz sentido, para algum algoritmo ordem n , dizer que sua complexidade é por exemplo $O(n!)$. Ou faz?

⇒ Exemplos da falta de precisão de O : $n \in O(n^2)$

$$n \in O(n^5)$$

$$n \in O(2^n)$$

$$n \in O(n^n)$$



Notação Θ

Conhecida também como “limite firme” ou “limite assintoticamente restrito”.

- A notação O , apesar de fornecer informações sobre a complexidade do algoritmo, nem sempre nos revela algo importante;
- Não faz sentido, para algum algoritmo ordem n , dizer que sua complexidade é por exemplo $O(n!)$. Ou faz?

⇒ Exemplos da falta de precisão de O :

$$n \in O(n^2)$$

$$n \in O(n^5)$$

$$n \in O(2^n)$$

$$n \in O(n^n)$$

Verdadeiras e
imprecisas



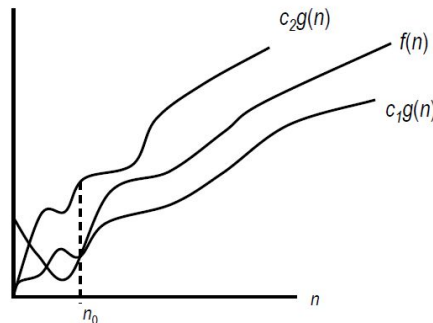
Notação Θ , limite firme

Dizemos que as funções f e g estão na mesma ordem e escrevemos $f = \Theta(g)$ se $f = O(g)$ e $f = \Omega(g)$.

Em outras palavras, $f = \Theta(g)$ significa que existe números positivos c_1 e c_2 tais que:

$$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$$

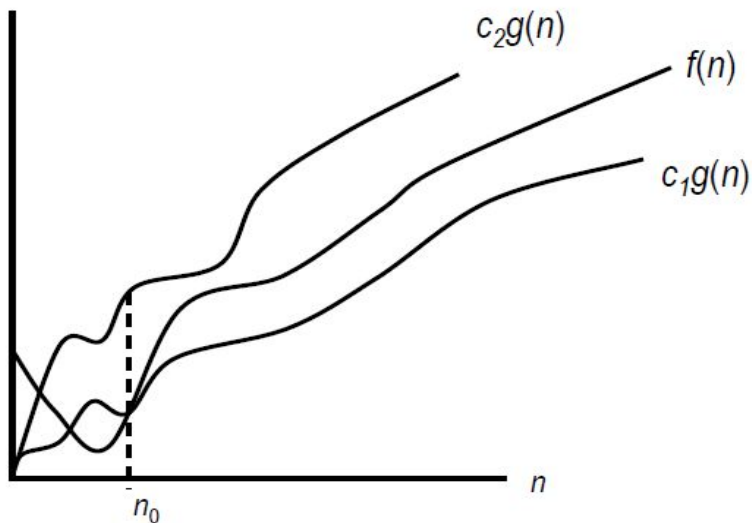
para todo n suficientemente grande.





Notação Θ

⇒ Limite assintótico firme



$$\Theta(g(n)) = \{f(n): \exists c_1, c_2 \text{ e } n_0 > 0 \mid 0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n) \forall n \geq n_0\}$$



Exemplo:

Prove que: $\frac{n^2}{2} - 3n \in \Theta(n^2)$

⇒ Para isso, devemos definir constantes c_1 , c_2 e n_0 tais que:

$$c_1 n^2 \leq \frac{1}{2}n^2 - 3n \leq c_2 n^2$$

⇒ Encontre constantes que satisfaça as duas desigualdades...



Exemplo:

Dividindo por n^2 :

$$\frac{c_1 n^2}{n^2} \leq \frac{1/2 n^2}{n^2} - \frac{3n}{n^2} \leq \frac{c_2 n^2}{n^2}$$

$$= c_1 \leq 1/2 - 3/n \leq c_2$$

é fácil achar

$$\Rightarrow c_1 = 1/14, \quad c_2 = 1/2 \quad \text{e} \quad n_0 = 7$$

Portanto, se existem tais constantes:

$$\frac{n^2}{2} - 3n \in \Theta(n^2)$$

é Verdadeiro



Observação:

$$f(x) \in \Theta(g(x))$$

sse

$$f(x) \in O(g(x))$$

e

$$f(x) \in \Omega(g(x))$$



Notação o

⇒ Limite assintoticamente superior não restrito;

Todas as funções de **O (ó-zão)** que não definem um limite assintoticamente restrito pertencem a "o" (ó-zinho).

se $f(n) \in O(g(n))$ e $f(n) \notin \Omega(g(n))$ então $f(n) \in o(g(n))$

Exemplos:

Assintoticamente restrito:

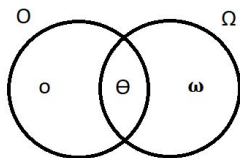
$$2n^2 \in O(n^2)$$

(ó-zão)

Assintoticamente não restrito:

$$2n \in O(n^2)$$

(ó-zão) e (ó-zinho)



$$\log(n) \in O(c^n)$$

(ó-zão) e (ó-zinho)



Comparativo: notação O x notação o

Definição: $O(g(n)) = \{f(n): \exists c \text{ e } n_0 > 0 \mid 0 \leq f(n) \leq c \cdot g(n) \quad \forall n \geq n_0\}$

Definição: $o(g(n)) = \{f(n): \exists c \text{ e } n_0 > 0 \mid 0 \leq f(n) < c \cdot g(n) \quad \forall n \geq n_0\}$

$f(n) \in O(g(n))$, o limite $0 \leq f(n) \leq c \cdot g(n)$ se mantém válido para alguma constante $c > 0$.

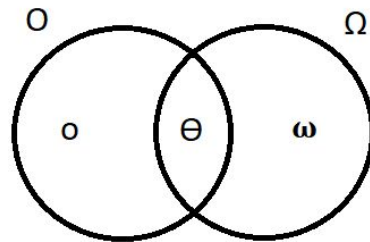
$f(n) \in o(g(n))$, o limite $0 \leq f(n) \leq c \cdot g(n)$ se mantém válido para todas as constantes $c > 0$.



Facilitando o entendimento...

Se $f(n) \in o(g(n))$ então

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





Notação ω

⇒ Limite assintoticamente inferior não restrito;

Todas as funções de Ω (ômega-zão) que não definem um limite assintoticamente restrito pertencem a “ ω ” (ômega-zinho).

se $f(n) \notin O(g(n))$ e $f(n) \in \Omega(g(n))$ então $f(n) \in \omega(g(n))$

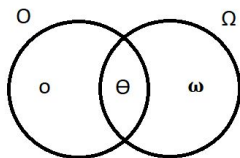
Exemplos:

Assintoticamente restrito:

$$2n^2 \in \Omega(n^2) \quad (\text{ômega-zão})$$

Assintoticamente não restrito:

$$2n^2 \in \Omega(n) \quad (\text{ômega-zão}) \text{ e } (\text{ômega-zinho})$$



$$c^n \in \Omega(\log(n)) \quad (\text{ômega-zão}) \text{ e } (\text{ômega-zinho})$$



Comparativo: notação Ω x notação ω

Definição: $\Omega(g(n)) = \{f(n): \exists c \text{ e } n_0 > 0 \mid 0 \leq c \cdot g(n) \leq f(n) \quad \forall n \geq n_0\}$

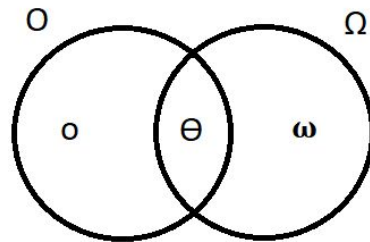
Definição: $\omega(g(n)) = \{f(n): \exists c \text{ e } n_0 > 0 \mid 0 \leq c \cdot g(n) < f(n) \quad \forall n \geq n_0\}$



Facilitando o entendimento...

Se $f(n) \in \omega(g(n))$ então

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





Limites para comparar ordens de grandeza

O limite da razão das funções em questão pode levar a três casos:

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} := \begin{cases} 0, & f(n) \text{ tem ordem de grandeza menor que } g(n) \\ c > 0, & f(n) \text{ tem mesma ordem de grandeza que } g(n) \\ \infty, & f(n) \text{ tem ordem de grandeza maior que } g(n) \end{cases}$$

Caso tenhamos uma indeterminação da forma $\frac{\infty}{\infty}$, usamos a regra de L'Hôpital:

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} := \lim_{n \rightarrow \infty} \frac{f'(n)}{g'(n)}$$



Comparação de funções

Muitas das propriedades relacionais de números reais também se aplicam a comparações assintóticas. No caso das propriedades seguintes, suponha $f(n)$ e $g(n)$ sejam assintoticamente positivas.

Transitividade:

| | | | | |
|-----------------------|---|-----------------------|----------|-------------------------|
| $f(n) = \Theta(g(n))$ | e | $g(n) = \Theta(h(n))$ | implicam | $f(n) = \Theta(h(n))$, |
| $f(n) = O(g(n))$ | e | $g(n) = O(h(n))$ | implicam | $f(n) = O(h(n))$, |
| $f(n) = \Omega(g(n))$ | e | $g(n) = \Omega(h(n))$ | implicam | $f(n) = \Omega(h(n))$, |
| $f(n) = o(g(n))$ | e | $g(n) = o(h(n))$ | implicam | $f(n) = o(h(n))$, |
| $f(n) = \omega(g(n))$ | e | $g(n) = \omega(h(n))$ | implicam | $f(n) = \omega(h(n))$. |



Reflexividade:

$$f(n) = \Theta(f(n))$$

$$f(n) = O(f(n))$$

$$f(n) = \Omega(f(n))$$

Simetria:

$$f(n) = \Theta(g(n)) \text{ se e somente se } g(n) = \Theta(f(n))$$

Simetria de transposição:

$$f(n) = O(g(n)) \text{ se e somente se } g(n) = \Omega(f(n))$$

$$f(n) = o(g(n)) \text{ se e somente se } g(n) = \omega(f(n))$$



Sabor

Pelo fato dessas propriedades se manterem para notações assintóticas, é possível traçar uma analogia entre a comparação assintótica de duas funções **f** e **g** e a comparação de dois números reais **a** e **b**:

$$\begin{array}{lll} f(n) = O(g(n)) & \approx & a \leq b, \\ f(n) = \Omega(g(n)) & \approx & a \geq b, \\ f(n) = \Theta(g(n)) & \approx & a = b, \\ f(n) = o(g(n)) & \approx & a < b, \\ f(n) = \omega(g(n)) & \approx & a > b. \end{array}$$



Obrigado

jorge@imd.ufrn.br