Aula 02 Notação Assintótica

Notação O, Ω, Θ e Exemplos

Prof. Marco Aurélio Stefanes

marco em dct.ufms.br

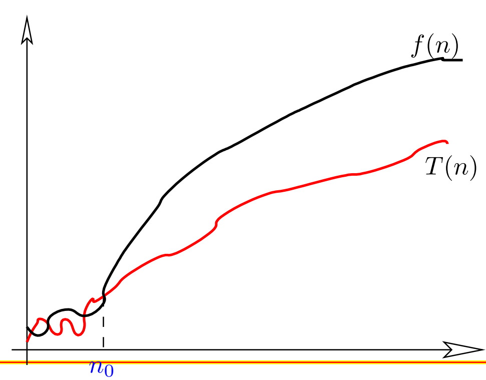
www.dct.ufms.br/ ˜ marco

Notação O

Sejam $T(n)$ e $f(n)$ funções dos inteiros nos reais.
Dizemos que $T(n)$ é $O(f(n))$ se existem constantes positivas c e n_0 tq

$$T(n) \leq c\, f(n)$$

para todo $n \geq n_0$.
i.e. para n grande, $T(n)$ não cresce mais rápido que $f(n)$



Notação O

Intuitivamente...

$O(f(n))$ $\approx$ funções que não crescem mais rápido que $f(n)$
$\approx$ funções menores ou iguais a um múltiplo de $f(n)$

n^2 $\quad (3/2)n^2 \quad 9999n^2 \quad n^2/1000$ etc.
crescem todas com a mesma velocidade, são todas $O(n^2)$.

- $n^2 + 99n$ é $O(n^2)$
- $33n^2$ é $O(n^2)$
- $9n + 2$ é $O(n^2)$
- $0{,}00001n^3 - 200n^2$ não é $O(n^2)$

Uso da notação O

Notação O: Usada para análise de pior caso de algoritmos, Também chamado limite superior.

"$T(n)$ é $O(f(n))$" deve ser entendido como "$T(n) \in O(f(n))$".

"$T(n) = O(f(n))$" deve ser entendido como "$T(n) \in O(f(n))$".

"$T(n)$ é $g(n) + O(f(n))$" significa que existem constantes positivas c e n_0 tais que

$$T(n) \leq g(n) + c\, f(n)$$

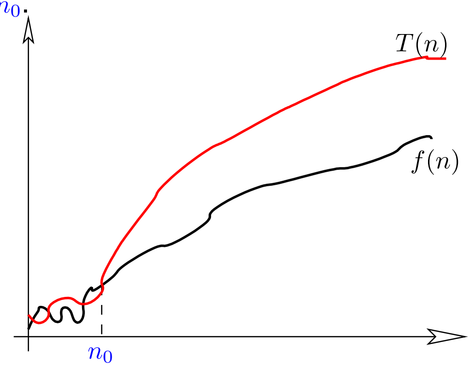
para todo $n \geq n_0$.

Notação Ω

Dizemos que $T(n)$ é $\Omega(f(n))$ se existem constantes positivas c e n_0 t.q.

$$T(n) \geq \; c\,f(n)$$

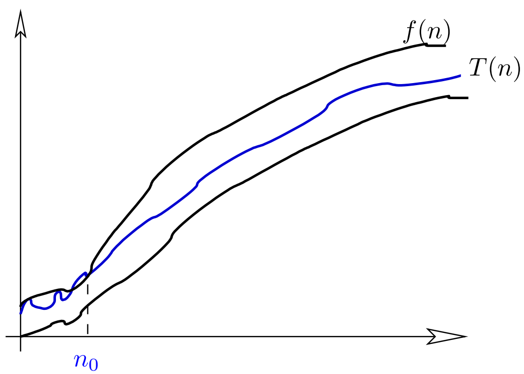
para todo $n \geq n_0$.



Usada para análise de melhor caso ou estabelecer complexidade de problemas.

Notação Θ

Dizemos que $T(n)$ é $\Theta(f(n))$ se
$f(n) \in O(g(n))$ e
$f(n) \in \Omega(g(n))$



Limite assintótico justo!

Propriedades

- Transitiva
  - $f = \Theta(g)$ e $g = \Theta(h) \Rightarrow f = \Theta(h)$
  - $f = O(g)$ e $g = O(h) \Rightarrow f = O(h)$
  - $f = \Omega(g)$ e $g = \Omega(h) \Rightarrow f = \Omega(h)$
- Reflexiva
  - $f = \Theta(f)$
  - $f = (f)$
  - $f = \Omega(g)$
- Simetria
  - $f = \Theta(g) \Leftrightarrow g = \Theta(f)$
  - $f = O(g) \Leftrightarrow g = \Omega(f)$

Ordem de Crescimento

$\lg n$	n	n^2	n^5	2^n
1	2	4	32	4
2	4	16	1024	16
3	8	64	32768	256
4	16	256	1048576	65536
5	32	1024	33554432	4.2E9
6	64	4096	1.0E9	1.8E19
7	128	16384	3.4E10	3.4E38
8	256	65536	1.1E12	1.6E77
9	512	262144	3.5E13	1.3E154
10	1024	1048576	1.1E15	$###$

Onde está $n \lg n$?

Ordem de Crescimento

Tempo de Execução	tamanho máximo do problema (n)		
	1 segundo	1 minuto	1 hora
$400n$	2500	150000	9000000
$20n \log n$	4096	166666	7826087
$2n^2$	707	5477	42426
n^4	31	88	44
2^n	19	25	31

Classificação de funções

- Θ define uma relação de equivalência
- $\Theta(f)$ é uma classe de equivalência - Classe de complexidade
- $O(1)$ conjunto de funções limitadas por uma constante (Independe do tamanho da entrada n)
- $\Theta(n)$ Linear
- $\Theta(n^2)$ Quadrática
- $\log n \in O(n^\alpha)$, para qualquer $\alpha > 0$
- $n^k \in O(c^n)$, para qualquer $0 < c \neq 1$

Notações Assintóticas

Duas outras notações assintóticas

- Notação o - oh pequeno - $T(n) = o(f(n))$
  - Limite superior não justo
  - Para todo c, há n_o t.q. $T(n) \leq cf(n)$, para $n \geq n_0$
  - $2n = o(n^2)$, mas $2n^2 \neq o(n^2)$
- Notação ω - $T(n) = \omega(f(n))$
  - Limite inferior não justo
  - Para todo c, há n_o t.q. $T(n) \leq cf(n)$, para $n \geq n_0$
  - $n^2/2 = \omega(n)$, mas $n^2/2 \neq o(n^2)$

Notação assintótica e limite

Podemos usar limites para a taxa de crecimento assintótico

- $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c \Leftrightarrow f(n) = \Theta(g(n))$
- $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \Leftrightarrow f(n) = O(g(n))$
- $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \Leftrightarrow f(n) = \Omega(g(n))$

Reais $\times$ Notação Assintótica

Podemos fazer uma analogia entre notação assintótica e números reais

Sejam f, g funções e $a, b \in R$

$$\begin{aligned} f(n) = O(g(n)) &\approx a \le b \\ f(n) = \Omega(g(n)) &\approx a \ge b \\ f(n) = \Theta(g(n)) &\approx a = b \\ f(n) = o(g(n)) &\approx a < b \\ f(n) = \Omega(g(n)) &\approx a > b \end{aligned}$$

Reais $\times$ Notação Assintótica

Podemos fazer uma analogia entre notação assintótica e números reais

Sejam f, g funções e $a, b \in R$

$$\begin{aligned} f(n) = O(g(n)) &\approx a \le b \\ f(n) = \Omega(g(n)) &\approx a \ge b \\ f(n) = \Theta(g(n)) &\approx a = b \\ f(n) = o(g(n)) &\approx a < b \\ f(n) = \Omega(g(n)) &\approx a > b \end{aligned}$$

Considere a funções $f(n) = n$ e $g(n) = n^{1+\sin n}$

$f(n)$ é $O(g(n))$?

$f(n)$ é $\Omega(g(n))$?

Reais $\times$ Notação Assintótica

Podemos fazer uma analogia entre notação assintótica e números reais

Sejam f, g funções e $a, b \in R$

$$\begin{aligned} f(n) = O(g(n)) &\approx a \le b \\ f(n) = \Omega(g(n)) &\approx a \ge b \\ f(n) = \Theta(g(n)) &\approx a = b \\ f(n) = o(g(n)) &\approx a < b \\ f(n) = \Omega(g(n)) &\approx a > b \end{aligned}$$

Considere a funções $f(n) = n$ e $g(n) = n^{1+\sin n}$

$f(n)$ é $O(g(n))$?

Exemplos

Considere a função

$$T(n) = \begin{cases} n & \text{se } n \text{ é impar,} \\ n^2 & \text{se } n \text{ é par} \end{cases}$$

$T(n)$ é $O(n^2)$?

$T(n)$ é $\Omega(n^2)$?

$T(n)$ é $\Theta(n^2)$?

$T(n)$ é $\Theta(n)$?

Logaritmos

- $\lg n = \log_2 n$
- $\ln n = \log_e n$
- $\log_b,\ b > 1$ é estritamente crescente
- $\log_b 1 = 0$
- $\log_b b = 1$
- $\log_b n^a = a \log_b n$
- $\log_b(nm) = \log_b n + \log_b m$
- $n^{\log m} = m^{\log n}$
- $\log_c n = \frac{\log_b n}{\log_b c}$

Exemplos

Mostrar que $\log_a n = \Theta(\log_b n)$ para qualquer $a, b > 1$ constante.
Resp. Devemos mostrar que existem c, c' t.q.

i) $\log_a \leq c \log_b n$

ii) $\log_a \geq c' \log_b n$

Exemplos

Mostrar que $\log_a n = \Theta(\log_b n)$ para qualquer $a, b > 1$ constante.

Exemplos

Mostrar que $\log_a n = \Theta(\log_b n)$ para qualquer $a, b > 1$ constante.
Resp. Devemos mostrar que existem c, c' t.q.

i) $\log_a \leq c \log_b n$

ii) $\log_a \geq c' \log_b n$

i) $log_a n \leq c \log_b n \Leftrightarrow log_a n \leq c \frac{\log_a n}{\log_a b}$

$$\Leftrightarrow log_a n \leq c \frac{\log_a n}{\log_a b} = \frac{c}{\log_a b} \log_a n$$

Tomando $c = \log_a b$ temos o resultado desejado.

Exemplos

Mostrar que $\log_a n = \Theta(\log_b n)$ para qualquer $a, b > 1$ constante.
Resp. Devemos mostrar que existem c, c' t.q.

i) $\log_a \leq c \log_b n$

ii) $\log_a \geq c' \log_b n$

i) $log_a n \leq c \log_b n \Leftrightarrow log_a n \leq c \frac{\log_a n}{\log_a b}$

$$\Leftrightarrow log_a n \leq c \frac{\log_a n}{\log_a b} = \frac{c}{\log_a b} \log_a n$$

Tomando $c = \log_a b$ temos o resultado desejado.

ii) Análogo ao anterior.

Exemplos

a) Sejam $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$
Mostrar que $f(n) = \Theta(g(n))$.
Resp. Encontrar c_1, c_2 e n_0 tq

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

Exemplos

a) Sejam $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$
Mostrar que $f(n) = \Theta(g(n))$.

Exemplos

a) Sejam $f(n) = \frac{1}{2}n^2 - 3n$ e $g(n) = n^2$
Mostrar que $f(n) = \Theta(g(n))$.
Resp. Encontrar c_1, c_2 e n_0 tq

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

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

$$c_1 = \frac{1}{14}, \ c_2 = \frac{1}{2}, n_0 = 7$$

Exemplos

b) Mostrar que $6n^3 \neq \Theta(n^2)$.

Exemplos

b) Mostrar que $6n^3 \neq \Theta(n^2)$.
Por contradição, supor que $\exists\ c_1, c_2, n_0 > 0$ tq

$$c_1 n^2 \leq 6n^3 \leq c_2 n^2$$

$$c_1 \leq 6n \leq c_2$$

Absurdo, pois $c_2 \geq 6n$ não vale para n arbitrariamente grande!

Exemplos

b) Mostrar que $6n^3 \neq \Theta(n^2)$.
Por contradição, supor que $\exists\ c_1, c_2, n_0 > 0$ tq

$$c_1 n^2 \leq 6n^3 \leq c_2 n^2$$

$nO(n) = O(n^2)$

"$nO(n) = O(n^2)$" significa "$nO(n) \subset O(n^2)$".
Ou seja, se $T(n)$ é $O(n)$, então $nT(n)$ é $O(n^2)$.

De fato, se $T(n)$ é $O(n)$ então existem constantes, digamos 10 e 10^{100}, tais que

$$T(n) \;\leq\; 10\, n$$

para todo $n \geq 10^{100}$. Desta forma,

$$n\, T(n) \;\leq\; n\, 10\, n \leq 10\, n^2$$

para todo $n \geq 10^{100}$. Logo $n\, T(n)$ é $O(n^2)$.

$nO(n) = O(n^2)$

"$nO(n) = O(n^2)$" significa "$nO(n) \subset O(n^2)$".
Ou seja, se $T(n)$ é $O(n)$, então $nT(n)$ é $O(n^2)$.

De fato, se $T(n)$ é $O(n)$ então existem constantes positivas c e n_0, tais que

$$T(n) \ \leq \ c\,n$$

para todo $n \geq n_0$. Desta forma,

$$n\,T(n) \ \leq \ n\,c\,n \leq c\,n^2$$

para todo $n \geq n_0$. Logo $n\,T(n)$ é $O(n^2)$.

Resumo da análise

Etapa para analisar um algoritmo (nesta ordem!)

- Correção
- Analise o pior caso
- Enfoque o termo principal do tempo de execução
- Melhore a ordem de magnitude
- Reduza os fatotes constantes

Propriedade Transitiva

Se $T(n) = O(f(n))$ e $f(n) = O(g(n))$ então $T(n) = O(g(n))$.

Propriedade Transitiva

Se $T(n) = O(f(n))$ e $f(n) = O(g(n))$ então $T(n) = O(g(n))$.
Rascunho da prova: Suponha, por exemplo, que

$0 \leq T(n) \leq 11 f(n)$ para todo $n \geq 3$

$0 \leq f(n) \leq 10\,g(n)$ para todo $n \geq 5$.

Defina $c := 11 \cdot 10 = 110$ e $n_0 := 5$.
Para todo $n \geq n_0$, temos $T(n) \geq 0$ e

$$T(n) \ \leq \ 11 f(n) \ \leq \ 11(10\,g(n)) \ = \ 110\,g(n)\,.$$

Exemplo: Se $f = O(3n^2 + 4n)$ então $f = O(n^2)$, pois $3n^2 + 4n = O(n^2)$. Faça caso geral!

Regra da Soma

Se $T(n) = O(f(n))$ então $T(n) + f(n) = O(f(n))$.
Prova:

Exercícios

- Prove que $O(1) = O(2)$

Regra da Soma

Se $T(n) = O(f(n))$ então $T(n) + f(n) = O(f(n))$.
Prova: Existem $c \geq 1$ e n_0 tq $0 \leq T(n) \leq cf(n)$ para todo $n \geq n_0$. Então

$$\begin{aligned} 0 &\leq T(n) + f(n) \\ &\leq cf(n) + f(n) \\ &\leq cf(n) + cf(n) \\ &= 2cf(n) \end{aligned}$$

para todo $n \geq n_0$.
Exemplo: Se $T(n) = O(n)$ então $T(n) + 3n^2 = O(3n^2)$.

Exercícios

- Prove que $O(1) = O(2)$
- Qual o menor valor t.q. $\lg \lg n > 10$

Exercícios

- Prove que $O(1) = O(2)$
- Qual o menor valor t.q. $\lg \lg n > 10$
- Mostre que $n \lg n = O(n^{3/2})$

Exercícios

- Prove que $O(1) = O(2)$
- Qual o menor valor t.q. $\lg \lg n > 10$
- Mostre que $n \lg n = O(n^{3/2})$
- Mostre que $\lg n = O(n)$
- Mostre que $3 \lg n + \lg \lg n = O(\lg n)$

Exercícios

- Prove que $O(1) = O(2)$
- Qual o menor valor t.q. $\lg \lg n > 10$
- Mostre que $n \lg n = O(n^{3/2})$
- Mostre que $\lg n = O(n)$