

Algoritmos e Estruturas de Dados III

CIC210

Análise de Algoritmos

Haroldo Gambini Santos

Universidade Federal de Ouro Preto - UFOP

15 de março de 2010

Tempo de Execução de um Algoritmo

- Depende:
 - Implementação computacional (programa, linguagem, compilador)
 - Ambiente de execução
 - Processador
 - Sistema Operacional
 - Memória (real e cache)
 - Barramento
 - ..
 - Organização e conteúdo dos dados de entrada

Tempo de Execução de um Algoritmo

- Depende:
 - Implementação computacional (programa, linguagem, compilador)
 - Ambiente de execução
 - Processador
 - Sistema Operacional
 - Memória (real e cache)
 - Barramento
 - ..
 - Organização e conteúdo dos dados de entrada

Difícil de caracterizar com exatidão!

Notação Assintótica

- Problemas realmente grandes
- Preocupação com o quão rápido os limites serão atingidos
 - Ordem de crescimento das funções

Notação Assintótica

Exemplo

- Algoritmos A e B resolvem um mesmo problema
- Complexidade de tempo:
 - A: $5.000n$
 - B: $1,1^n$

Notação Assintótica

Exemplo

- Algoritmos A e B resolvem um mesmo problema
- Complexidade de tempo:
 - A: $5.000n$
 - B: $1,1^n$
- Considere $n = 10$
 - A requer 50.000 passos
 - B requer somente 3

Notação Assintótica

Exemplo

- Algoritmos A e B resolvem um mesmo problema
- Complexidade de tempo:
 - A: $5.000n$
 - B: $1,1^n$
- Considere $n = 10$
 - A requer 50.000 passos
 - B requer somente 3
- Considere $n = 1000$
 - A requer 5.000.000 passos
 - B requer 2.5×10^{41}

Notação Assintótica

Exemplo

- Algoritmos A e B resolvem um mesmo problema
- Complexidade de tempo:
 - A: $5.000n$
 - B: $1,1^n$
- Considere $n = 10$
 - A requer 50.000 passos
 - B requer somente 3
- Considere $n = 1000$
 - A requer 5.000.000 passos
 - B requer 2.5×10^{41}
- Para **todos** os casos de n , exceto $n < 142$, o tempo de A será menor do que o de B

Funções Típicas

Tempo de Execução
Estimativa em um Computador com 4 GHz

n	$f = (n!)$	$f = (2^n)$	$f = (n^5)$	$f = (n^3)$	$f = (n^2)$
15	5 minutos	< 1s	< 1s	< 1s	< 1s
20	20 anos	< 1s	< 1s	< 1s	< 1s
50	d.f.v.t.	4 dias	< 1s	< 1s	< 1s
100	d.f.v.t.	d.f.v.t.	3 segundos	< 1s	< 1s
500	d.f.v.t.	d.f.v.t.	3 horas	< 1s	< 1s
1.000	d.f.v.t.	d.f.v.t.	3 dias	< 1s	< 1s
10.000	d.f.v.t.	d.f.v.t.	8 séculos	4 minutos	< 1s
15.000	d.f.v.t.	d.f.v.t.	65 séculos	14 minutos	< 1s

Notação Assintótica

Detalhes sem importância

- Constantes multiplicativas
- Termos de mais baixa ordem

Notação Θ

■ Seja:

■ $c_1, c_2 \in \mathbb{R}^+$ e $n_0 \in \mathbb{N}$

Notação Θ

- Seja:
 - $c_1, c_2 \in \mathbb{R}^+$ e $n_0 \in \mathbb{N}$
- Para uma função $g(n)$ $\Theta(g(n))$ é o conjunto de funções:

$$\begin{aligned}\Theta(g(n)) &= f(n) : \\ &\text{exista } 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \\ &\text{para todo } n > n_0\end{aligned}$$

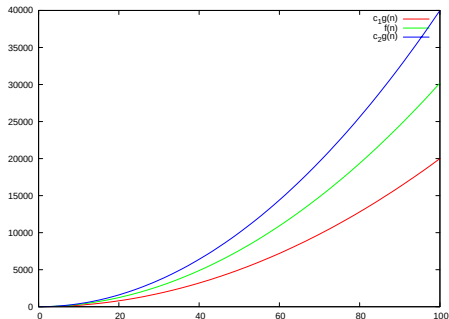
Notação Θ

- Seja:
 - $c_1, c_2 \in \mathbb{R}^+$ e $n_0 \in \mathbb{N}$
- Para uma função $g(n)$ $\Theta(g(n))$ é o conjunto de funções:

$$\begin{aligned}\Theta(g(n)) = f(n) : \\ \text{exista } 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \\ \text{para todo } n > n_0\end{aligned}$$

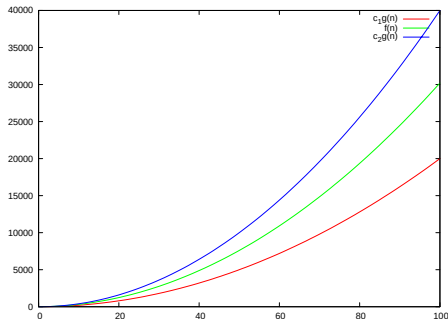
- $f(n)$ fica “imprensada” entre $c_1 g(n)$ e $c_2 g(n)$ para valores grandes de n
- para dizer que $f(n) \in \Theta(g(n))$ pode-se escrever
 $f(n) = \Theta(g(n))$

Notação Θ



$g(n)$ é um limite assintoticamente restrito para $f(n)$

Notação Θ



$g(n)$ é um limite assintoticamente restrito para $f(n)$

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

se $g(n)$ cresce na mesma taxa que $f(n)$

Notação Θ

Receita

- Descartar os termos de mais baixa ordem
- Ignorar o coeficiente inicial do termo de mais baixa ordem

Exemplo

$$3n^2 + 2n$$

...

Outras Notações

$$\begin{aligned} O(g(n)) &= f(n) : \\ &\text{exista } 0 \leq f(n) \leq c_g(n) \\ &\text{para todo } n > n_0, c \in \mathbb{R}^+ \end{aligned}$$

Outras Notações

$$\begin{aligned}\Omega(g(n)) &= f(n) : \\ &\text{exista } 0 \leq c_g(n) \leq f(n) \\ &\text{para todo } n > n_0, c \in \mathbb{R}^+\end{aligned}$$