

Classificação de Dados

Aula 03

Introdução à Análise de Complexidade de Algoritmos;
Notações Theta, O e Omega

UFRGS

INF01124

1

Tempo de Execução de Algoritmos

- Em alguns casos pode-se calcular exatamente o tempo de execução de um algoritmo, mas esta precisão pode não justificar o esforço
- Para entradas suficientemente grandes, constantes multiplicativas e termos de mais baixa ordem podem ser desconsiderados, *i.e.*, o termo de mais alta ordem domina o custo do algoritmo
 - $3n^2 + 10n + 7 = \Theta(n^2)$
 - Eficiência assintótica*
- Um algoritmo assintoticamente mais eficiente apresenta menor tempo de execução para todas as entradas a partir de um certo tamanho

2

Eficiência Assintótica

- Exemplo: Resolução de Sistemas Lineares

$$\begin{aligned} ax + by + cz &= j \\ dx + ey + fz &= k \\ gx + hy + iz &= l \end{aligned}$$

- Regra de Cramer

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} j \\ k \\ l \end{bmatrix} \quad x = \frac{\begin{vmatrix} j & b & c \\ k & e & f \\ l & h & i \end{vmatrix}}{\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}} \quad y = \frac{\begin{vmatrix} a & j & c \\ d & k & f \\ g & l & i \end{vmatrix}}{\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}} \quad z = \frac{\begin{vmatrix} a & b & j \\ d & e & k \\ g & h & l \end{vmatrix}}{\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}}$$

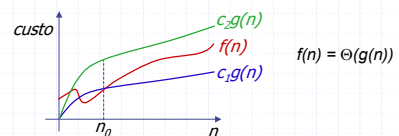
- Método de Eliminação de Gauss

Exercício: Pesquisar o custo destes dois algoritmos

3

A Notação Θ

- Para uma dada função $g(n)$, representando o custo assintótico de um algoritmo, define-se:
 - $\Theta(g(n)) = \{ f(n) : \exists c_1, c_2, n_0 > 0 \mid 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n), \forall n \geq n_0 \}$
 - $f(n) = \Theta(g(n))$ ou $f(n) \in \Theta(g(n))$
 - $g(n)$ é dito um limite assintoticamente justo (*tight*) para $f(n)$



4

A Notação Θ : Exemplo 1

- $f(n) = 3n^2 + 5n = \Theta(n^2)$
 - $c_1 n^2 \leq 3n^2 + 5n \leq c_2 n^2 \quad \forall n \geq n_0$ (dividindo por n^2)
 - $c_1 \leq 3 + 5/n \leq c_2$
 - Para $n \geq 1$, $c_1 \leq 3$
 - $n \geq 1$, $c_2 \geq 8$, $n_0 = 1$
 - $3n^2 \leq 3n^2 + 5n \leq 8n^2, \forall n \geq 1$
- Existem outras opções de constantes
- As constantes selecionadas dependem da função

5

A Notação Θ : Exemplo 2

- $f(n) = 0.5n^3 - 6n = \Theta(n^3)$
 - $c_1 n^3 \leq 0.5n^3 - 6n \leq c_2 n^3 \quad \forall n \geq n_0$ (dividindo por n^3)
 - $c_1 \leq 0.5 - 6/n^2 \leq c_2$
 - Para $n = 1, 2, 3$, $f(n) \leq 0$
 - $n = 4$, $c_1 \leq 0.5 - 6/16 = 1/8$
 - $c_2 \geq 0.5$, $n_0 = 4$
 - $(1/8)n^3 \leq 0.5n^3 - 6n \leq 0.5n^3, \forall n \geq 4$

6

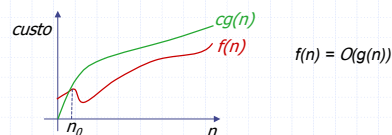
A Notação Θ : Observações

- ◆ A inequação pode ser satisfeita fazendo-se c_1 assumir um valor ligeiramente menor que o coeficiente do termo de mais alta ordem e fazendo-se c_2 assumir um valor ligeiramente maior que este mesmo coeficiente
- ◆ Para todo polinômio $p(n) = \sum_{i=0}^d a_i n^i$, onde $a_d > 0$, $p(n) = \Theta(n^d)$
- ◆ $\Theta(1)$ representa uma constante ou uma função constante com relação a alguma variável

7

A Notação O

- ◆ Utiliza-se a notação O quando dispõe-se apenas de um limite assintótico superior (limite para o tempo de execução do pior caso):
 $O(g(n)) = \{ f(n) : \exists c, n_0 > 0 \mid 0 \leq f(n) \leq cg(n), \forall n \geq n_0 \}$
- ◆ Para todos os valores de $n > n_0$, o valor de $f(n)$ é menor que $g(n)$
- ◆ $f(n) = \Theta(g(n)) \rightarrow f(n) = O(g(n))$, i.e., Θ é mais forte que O



8

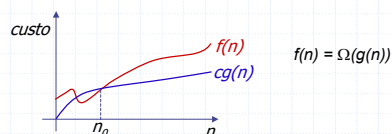
Notações: Θ versus O

- ◆ A notação O define um limite para o tempo de execução do pior caso de um algoritmo para quaisquer entradas
 - ◆ $O(n^2)$ é o limite para o pior caso para o algoritmo de classificação por inserção direta
- ◆ A notação Θ define limites para o tempo de execução de um algoritmo, mas depende dos valores de entrada utilizados
 - ◆ Por exemplo, caso a sequência já se encontre ordenada, o algoritmo de classificação por inserção direta executa em $\Theta(n)$
- ◆ Se $f(n) = an^2 + bn + c$, com $a > 0$, então $f(n) = O(n^2)$ e $f(n) = \Theta(n^2)$
- ◆ Se $f(n) = an + b$, com $a > 0$, então $f(n) = O(n)$, mas $f(n) \neq \Theta(n^2)$

9

A Notação Ω

- ◆ Utiliza-se a notação Ω quando dispõe-se apenas de um limite assintótico inferior (limite para o tempo de execução do melhor caso):
 $\Omega(g(n)) = \{ f(n) : \exists c, n_0 > 0 \mid 0 \leq cg(n) \leq f(n), \forall n \geq n_0 \}$
- ◆ Para todos os valores de $n > n_0$, o valor de $f(n)$ é maior que $g(n)$
- ◆ $f(n) = \Theta(g(n)) \rightarrow f(n) = \Omega(g(n))$, i.e., Θ é mais forte que Ω



10

Teorema

- ◆ $f(n) = \Theta(g(n))$ se e somente se $f(n) = O(g(n))$ e $f(n) = \Omega(g(n))$

11

Exercício

- ◆ Quais os limites superior e inferior do algoritmo de classificação por inserção direta? Utilize as notações O e Ω .

12

A Notação o

- Utiliza-se a notação o quando o limite superior não é assintoticamente justo (tight)
- $\alpha(g(n)) = \{ f(n) : \forall c > 0, \exists n_0 > 0 \mid 0 \leq f(n) < cg(n), \forall n \geq n_0 \}$
- Para todos os valores de $n > n_0$, $f(n)$ é bem menor que $g(n)$
- Exemplo: $2n = o(n^2)$, mas $2n^2 \neq o(n^2)$
- Na notação o , quando n tende a infinito, $f(n)$ torna-se insignificante em relação a $g(n)$

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

13

A Notação ω

- Utiliza-se a notação ω quando o limite inferior não é assintoticamente justo (tight)
- $\omega(g(n)) = \{ f(n) : \forall c > 0, \exists n_0 > 0 \mid 0 \leq cg(n) < f(n), \forall n \geq n_0 \}$
- Para todos os valores de $n > n_0$, $f(n)$ é bem maior que $g(n)$
- Exemplo: $2n^2 = \omega(n)$, mas $2n^2 \neq \omega(n^2)$
- Na notação ω , quando n tende a infinito, $g(n)$ torna-se insignificante em relação a $f(n)$

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

14

Propriedades Relacionais

Transitividade

$$\begin{aligned} f(n) = \Theta(g(n)) \text{ e } g(n) = \Theta(h(n)) &\rightarrow f(n) = \Theta(h(n)) \\ f(n) = O(g(n)) \text{ e } g(n) = O(h(n)) &\rightarrow f(n) = O(h(n)) \\ f(n) = \Omega(g(n)) \text{ e } g(n) = \Omega(h(n)) &\rightarrow f(n) = \Omega(h(n)) \\ f(n) = o(g(n)) \text{ e } g(n) = o(h(n)) &\rightarrow f(n) = o(h(n)) \\ f(n) = \omega(g(n)) \text{ e } g(n) = \omega(h(n)) &\rightarrow f(n) = \omega(h(n)) \end{aligned}$$

Reflexividade

$$f(n) = \Theta(f(n)) \quad f(n) = O(f(n)) \quad f(n) = \Omega(f(n))$$

15

Propriedades Relacionais (Cont.)

Simetria

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

Simetria Transposta

$$\begin{aligned} f(n) = O(g(n)) \text{ se e somente se } g(n) = \Omega(f(n)) \\ f(n) = o(g(n)) \text{ se e somente se } g(n) = \omega(f(n)) \end{aligned}$$

Obs.: Dadas duas funções $f(n)$ e $g(n)$, é possível que $f(n) \neq O(g(n))$ e $f(n) \neq \Omega(g(n))$

16

Analogia com Números Reais

- Sejam a e b dois números reais

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

17