

Classificação de Dados

Aula 03
Introdução à Análise de Complexidade de Algoritmos;
Notações Theta, O e Omega

UFRGS

INF01124

Complexidade

- Porquê o estudo da Complexidade?
 - Performance
 - Escolher entre vários algoritmos o mais eficiente para implementar;
 - Desenvolver novos algoritmos para problemas que já têm solução;
 - Desenvolver algoritmos mais eficientes (melhorar os algoritmos), devido ao aumento constante do "tamanho" dos problemas a serem resolvidos.
- O estudo da Complexidade Computacional torna possível determinar se a implementação de determinado algoritmo é viável.

3

Instituto de Informática - UFRGS

Análise de Algoritmos

- Análise de Algoritmo**
 - Tempo de processamento em função dos dados de entrada;
 - Espaço de memória total requerido para os dados;
 - Comprimento total do código;
 - Correla obtenção do resultado pretendido;
 - Robustez (como comporta-se com as entradas inválidas ou não previstas).
- Análise de Algoritmos é medição de complexidade de algoritmo**
 - Quantidade de "trabalho" necessária para a sua execução, expressa em função das **operações fundamentais**, as quais variam de acordo com o algoritmo, e em função do volume de dados.

Instituto de Informática - UFRGS

Análise de Algoritmos

- Medidas de Análise**
 - Devem ser independentes da tecnologia (hardware/software)
 - Modelos Matemáticos simplificados baseados nos fatores relevantes:
 - Tempo de Execução
Uma função que relaciona o tempo de execução com o tamanho de entrada:
 $t = F(n)$
 - Conjunto de operações a serem executadas.
 - Custo associado à execução de cada operação.
 - Ocupação de Espaço em Memória

4

Instituto de Informática - UFRGS

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

5

Instituto de Informática - UFRGS

Tamanho do problema

Complexidade de tempo	Tamanho de problema executável em		
	1 segundo	1 minuto	1 hora
$\log_2 n$	2^{10^6}	$2^{6 \cdot 10^7}$	$2^{3,6 \cdot 10^9}$
n	10^6	$6 \cdot 10^7$	$3,6 \cdot 10^9$
$n \cdot \log_2 n$	62.746	$2,8 \cdot 10^6$	$1,3 \cdot 10^8$
n^2	10^3	$7,746 \cdot 10^3$	60.000
n^3	10^2	$3,9 \cdot 10^2$	$1,5 \cdot 10^3$
2^n	20	25	32
3^n	13	16	20

Tabela 1.1.3: Complexidade do algoritmo x Tempo de execução

Do Livro: Complexidade de Algoritmos, Laíra Toscani & Paulo Veloso, Série Livros Didáticos, II-UFRGS

Instituto de Informática - UFRGS

Custo do cálculo de determinantes

n	Método de Cramer	Método de Gauss
2	22 μ s	50 μ s
3	102 μ s	159 μ s
4	456 μ s	353 μ s
5	2,35 ms	666 μ s
10	1,19 min	4,95 ms
20	15 225 séculos	38,63 ms
40	$5 \cdot 10^{33}$ séculos	0,315 s

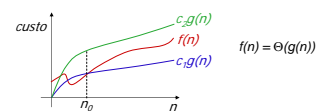
Tabela 1.1.1: Tamanho do Problema x Tempo de Execução

Do Livro: Complexidade de Algoritmos, Laíra Toscani & Paulo Veloso, Série Livros Didáticos, II-UFRGS

Instituto de Informática - UFRGS

A Notação Θ (Theta)

- A notação Θ define um limite assintótico exato, a menos de constantes
- 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 $f(n)$



Instituto de Informática - UFRGS

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
$$p(n) = \sum_{i=0}^d a_i n^i, \text{ onde } a_d > 0, p(n) = \Theta(n^d)$$
- ◆ $\Theta(1)$ representa uma constante ou uma função constante com relação a alguma variável

Instituto de Informática - UFRGS

Notações: Θ versus O

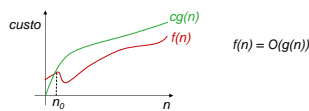
- ◆ A notação O define um limite para o tempo de execução do pior caso de um algoritmo para quaisquer entradas
 - ◆ $O(n^2)$ é o limite para o pior caso para o algoritmo de classificação por inserção direta
- ◆ A notação Θ define limites para o tempo de execução de um algoritmo, mas depende dos valores 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 $f(n) \neq \Theta(n^2)$

11

Instituto de Informática - UFRGS

A Notação O (omicron ou “O grande”)

- ◆ 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

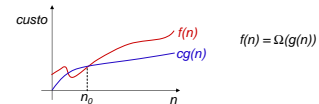


10

Instituto de Informática - UFRGS

A Notação Ω (Omega)

- ◆ 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



12

Instituto de Informática - UFRGS

Métodos de Classificação de Dados

- Classificação interna (*internal sorting*) é realizada em memória de acesso direto
- Caso o volume de dados ultrapasse a capacidade da memória é necessário utilizar classificação externa (*external sorting*), com armazenamento normalmente em disco (ou outro dispositivo de armazenamento externo)

Instituto de Informática - UFRGS