

ESTRUTURA DE DADOS

ANÁLISE DE ALGORITMOS

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

1

BIBLIOGRAFIA

- GOODRICH, Michael. T., TAMASSIA, Roberto. **Estruturas de Dados e Algoritmos em Java**. São Paulo: Bookman, 2002.
 - Capítulo 3: Ferramentas de Análise

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

2

ALGORITMO

- Um algoritmo é um procedimento, passo a passo, para realizar uma tarefa em um tempo finito.



17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

3

TEMPO DE EXECUÇÃO

- Depende de uma série de fatores:
 - Cresce com o tamanho da entrada.
 - Afetado pelo hardware (processador, frequência, memória, disco, etc.)
 - Afetado pelo software (sistema operacional, linguagem de programação, compilador, interpretador, etc.).

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

4

ESTUDOS EXPERIMENTAIS

- **Problema:** experimentos têm limitações:
 - Podem ser feitos em um número limitado de entradas de teste;
 - É difícil comparar a eficiência de dois algoritmos a não ser que os experimentos tenham sido feitos com o mesmo hardware e software;
 - É necessário implementar e executar um algoritmo.

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

5

ESTUDOS EXPERIMENTAIS

- **Solução:** metodologia geral para analisar o tempo de execução:
 - Leva em conta todas as entradas possíveis;
 - Permite-nos avaliar a eficiência relativa de dois algoritmos de forma independente de hardware e do software;
 - Pode ser feita através do estudo de uma descrição em alto nível do algoritmo.

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

6

ESTUDOS EXPERIMENTAIS

- ❑ **Solução:** metodologia geral para analisar o tempo de execução:
 - objetiva associar a cada algoritmo uma função $f(n)$ que caracteriza o tempo de execução do algoritmo com uma função do tamanho n da entrada;

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

7

ESTUDOS EXPERIMENTAIS

- ❑ **Exemplo:**
 - “O algoritmo A é executado em tempo proporcional a n ”.
 - Querendo dizer que se fôssemos realizar experimento contaríamos que o tempo de execução do algoritmo A para qualquer entrada de tamanho n nunca excede cn , onde c é uma constante que depende do hardware e do software usados no experimento.

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

8

PSEUDOCÓDIGO

- ❑ **Pseudocódigo:**
 - mistura de linguagem natural e estruturas de programação;
 - usada para descrever as idéias principais da implementação genérica;
- ❑ **Principais estruturas de programação:**
 - expressões
 - declarações de métodos
 - estruturas de decisão
 - estruturas de repetição
 - indexação de arranjos
 - chamadas de métodos
 - retorno de métodos

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

9

EXEMPLO

- ❑ **Problema:** descobrir qual o maior elemento em um vetor V
- ❑ **Entrada:** um vetor V com $n \geq 1$ elementos inteiros
- ❑ **Saída:** o maior elemento de V

```
Algoritmo maiorElemento(V, n)
    maior ← V[0]
    para i de 1 até n - 1 faça
        se (V[i] > maior) então
            maior ← V[i]
    fimse
    fimpara
    retorna maior
```

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

10

ANÁLISE DE ALGORITMOS

- ❑ Definimos um conjunto de **operações primitivas** de alto nível que são independentes da linguagem de programação usada e podem ser identificadas no pseudocódigo;
- ❑ As seguintes operações estão incluídas entre as operações primitivas:
 - atribuição de valores a variáveis
 - chamadas de métodos
 - operações aritméticas
 - comparação de dois números
 - acesso a um arranjo
 - seguir uma referência a um objeto
 - retorno de um método

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

11

CONTANDO OPERAÇÕES

Algoritmo maiorElemento(V, n)	# operações
maior ← V[0]	2
para i de 1 até n - 1 faça	1 + n
se (V[i] > maior) então	2(n - 1)
maior ← V[i]	2(n - 1)
fimse	
{incrementa contador i}	2(n - 1)
fimpara	
retorna maior	1
	Total 7n - 2

- ❑ Número mínimo de operações:
 $2 + 1 + n + 4(n - 1) + 1 = 5n$
- ❑ Número máximo de operações:
 $2 + 1 + n + 6(n - 1) + 1 = 7n - 2$

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

12

ANÁLISE DO CASO MÉDIO E DO PIOR CASO

- Um algoritmo pode ser mais rápido sobre algumas entradas do que sobre outras.
- Podemos desejar expressar o tempo de execução desse algoritmo como uma média calculada com todas as entradas possíveis (**caso médio**).
- Uma análise de caso médio requer tipicamente que sejam calculados os tempos de execução baseados em uma distribuição de probabilidade.
- Portanto, a não ser que especificado de outra forma, caracterizaremos os tempos de execução em termos de **pioor caso**.

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

13

NOTAÇÃO ASSINTÓTICA

- Na análise de algoritmos, é importante concentrar-se na taxa de crescimento do tempo de execução com uma função do tamanho da entrada n , obtendo-se um quadro geral do comportamento, em vez de concentrar-se nos detalhes menores.
- Freqüentemente basta saber que o tempo de execução de um algoritmo como `maiorElemento` **crece proporcionalmente a n** , como o verdadeiro tempo de execução sendo n vezes algum pequeno fator constante que depende do ambiente de hardware e software e varia em faixa de valores dependendo da entrada específica.

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

14

NOTAÇÃO O (BIG-O)

- Sejam $f(n)$ e $g(n)$ funções mapeando inteiros não-negativos em números reais.
 - Dizemos que $f(n)$ é $O(g(n))$ se existe uma constante real $c > 0$ e uma constante inteira $n_0 \geq 1$ tais que $f(n) \leq cg(n)$ para todo inteiro $n \geq n_0$.
 - Esta definição é geralmente chamada de notação **big-O**, pois geralmente se diz " $f(n)$ é big-O de $g(n)$ ".

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

15

EXEMPLO: $7n - 2$ é $O(n)$

- Justificativa:** pela definição da notação O , precisamos achar uma constante $c > 0$ e uma constante inteira $n_0 > 1$ tais que $7n - 2 \leq cn$ para todo inteiro $n \geq n_0$: $c = 7$ e $n_0 = 1$.
- Esta é uma das infinitas escolhas possíveis, pois qualquer número real maior ou igual a 7 será um escolha possível para c e qualquer inteiro maior ou igual a 1 é uma escolha possível para n_0 .
- Podemos dizer que o tempo de execução do algoritmo `maiorElemento` para determinar o maior elemento de um arranjo de n inteiros é $O(n)$.

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

16

FUNÇÕES COMUNS

- Algumas funções aparecem com freqüência na análise de algoritmos e estruturas de dados, e seguidamente usamos termos especiais para nos referir a elas.
- A tabela abaixo descreve este e outros termos comumente usados na análise de algoritmos.

Logaritmica	Linear	Quadrática	Polinomial	Exponencial
$O(\log n)$	$O(n)$	$O(n^2)$	$O(n^k)$ com $k \geq 1$	$O(a^n)$ com $a > 1$

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

17

CRESCIMENTO DE VÁRIAS FUNÇÕES

n	$\log n$	\sqrt{n}	n	$n \log n$	n^2	n^3	2^n
2	1	1,4	2	2	4	8	4
4	2	2	6	8	16	64	16
8	3	2,8	8	24	64	512	256
16	4	4	16	64	256	4.096	65.536
32	5	5,7	32	160	1.024	32.768	4.294.967.296
64	6	8	64	384	4.096	262.144	$1,84 \times 10^{19}$
128	7	11	128	896	16.384	2.097.152	$3,40 \times 10^{38}$
256	8	16	256	2.048	65.536	16.777.216	$1,15 \times 10^{77}$
512	9	23	512	4.608	262.144	134.217.728	$1,34 \times 10^{154}$
1.024	10	32	1.024	10.240	1.048.576	1.073.741.824	$1,79 \times 10^{308}$

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

18

PARENTES DE BIG-O

- A notação **Big-O** fornece uma maneira assintótica de dizer que uma função é “menor ou igual a” outra função.
- Outras notações fornecem maneiras assintóticas de fazer outros tipos de comparações.

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

19

PARENTES DE O (BIG-ÔMEGA)

- Sejam $f(n)$ e $g(n)$ funções mapeando número inteiros em números reais.
 - Dizemos que $f(n)$ é $\Omega(g(n))$ (dito “ $f(n)$ é ômega de $g(n)$ ”) se $g(n)$ é $O(f(n))$.
 - Ou seja, se existe uma constante $c > 0$ e uma constante inteira $n_0 \geq 1$ tais que $f(n) \geq cg(n)$ para $n \geq n_0$.
- Esta função nos permite dizer que uma função é assintoticamente maior que ou igual a outra, exceto por uma fator constante.

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

20

PARENTES DE O (BIG-THETA)

- Da mesma forma:
 - Dizemos que $f(n)$ é $\Theta(g(n))$ (dito “ $f(n)$ é theta de $g(n)$ ”) se $f(n)$ é $O(g(n))$ e $f(n)$ é $\Omega(g(n))$.
 - Ou seja, existem constantes $c' > 0$ e $c'' > 0$ e uma constante inteira $n_0 \geq 1$ tais que $c'g(n) \leq f(n) \leq c''g(n)$ para $n > n_0$.
- A definição de Θ permite-nos dizer que duas funções são assintoticamente iguais, exceto por um fator constante.

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

21

RESUMINDO

- As notações O , Ω e Θ fornecem uma linguagem conveniente para a análise de estruturas de dados e algoritmos.
- Estas notações são convenientes porque permitem que nos concentremos nos aspectos gerais do comportamento de um algoritmo, em vez de seu detalhes.

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

22

EXEMPLO: MÉDIA PREFIXADA

- Se tivermos um vetor X armazenando n números, desejamos compor um vetor V tal que $V[i]$ seja a média dos elementos $X[0], \dots, X[i]$ para $i = 0, \dots, n - 1$.

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

23

ALGORITMO: TEMPO QUADRÁTICO

Entrada: um vetor X com $n \geq 1$ elementos.
Saída: um vetor V com n elementos tal que $V[i]$ é a média $X[0], \dots, X[i]$.

```
Algoritmo medial(X, n)
  para i de 0 até n - 1 faça
    soma ← 0
    para j de 0 até i faça
      soma ← soma + X[j]
    fimpara
    V[i] ← soma / (i + 1)
  fimpara
  retorna vetor V
```

- Ele calcula cada elemento de A separadamente.
- O tempo de execução deste algoritmo é $O(n^2)$.

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

24

ALGORITMO: TEMPO LINEAR

Entrada: um vetor X com $n \geq 1$ elementos.

Saída: um vetor V com n elementos tal que $V[i]$ é a média $X[0], \dots, X[i]$.

```
Algoritmo media2(X, n)
  soma ← 0
  para i de 0 até n - 1 faça
    soma ← soma + X[i]
    V[i] ← soma / (i + 1)
  fimpara
  retorna vetor V
```

- Se denotarmos com S_i a soma prefixada $X[0] + X[1] + \dots + X[i]$, podemos calcular as médias prefixadas como sendo $A[i] = S_i / (i + 1)$.
- O tempo de execução de `media2` é $O(n)$.

EXEMPLO

- Um algoritmo para testar se o primeiro elemento de um vetor é igual ao segundo.

ALGORITMO

Entrada: um vetor V com $n \geq 2$ elementos.

Saída: mensagem acusando a igualdade ou não dos elementos.

```
Algoritmo primeirosIguais(V, n)
  se (V[0] = V[1]) então
    Escreva "são iguais"
  senão
    Escreva "não são iguais"
  fimse
  fim
```

ANÁLISE

- Completamente independente do número de elementos no vetor.
- Diz-se que este algoritmo tem um tempo de execução **constante**.
- Representado na notação Big-O como $O(1)$.

EXEMPLO

- Um algoritmo que testa se o primeiro elemento de um vetor é igual a qualquer um dos outros elementos do vetor.

ALGORITMO

Entrada: um vetor V com $n \geq 2$ elementos.

Saída: mensagem acusando a igualdade de um elemento do vetor com o primeiro elemento.

```
Algoritmo temIgual(V, n)
  para i de 1 até n - 1 faça
    se (V[0] = V[i]) então
      Escreva "tem um igual"
  fimse
  fimpara
  fim
```

ANÁLISE

- Exigirá no máximo $n - 1$ comparações.
- Diz-se que um algoritmo que exige um total de $n - 1$ comparações é $O(n)$.
- Diz-se que um algoritmo $O(n)$ tem um tempo de execução **linear**.

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

31

EXEMPLO

- Um algoritmo que testa se qualquer elemento de um vetor é duplicado em outra parte do vetor.

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

32

ALGORITMO

Entrada: um vetor V com $n \geq 1$ elementos.
Saída: mensagem acusando elementos duplicados.

```
Algoritmo temDuplicados (V, n)
  para i de 0 até n - 2 faça
    para j de 0 até n - 1 faça
      se (V[i] = V[j]) então
        Escreva V[i] "está duplicado"
      fimse
    fimpara
  fimpara
fim
```

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

33

ANÁLISE

- Algoritmos para resolver este problema aumentam conforme o quadrado de n .
- Assim o Big-O ignora a constante e os algoritmos são considerados $O(n^2)$.
- $O(n^2)$ é denominado tempo de execução **quadrático**.

17/2/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

34