Algoritmos

Instituto Federal de Educação, Ciência e Tecnologia de Goiás - Formosa

Prof. Waldeyr Mendes Cordeiro da Silva

2019

Agenda

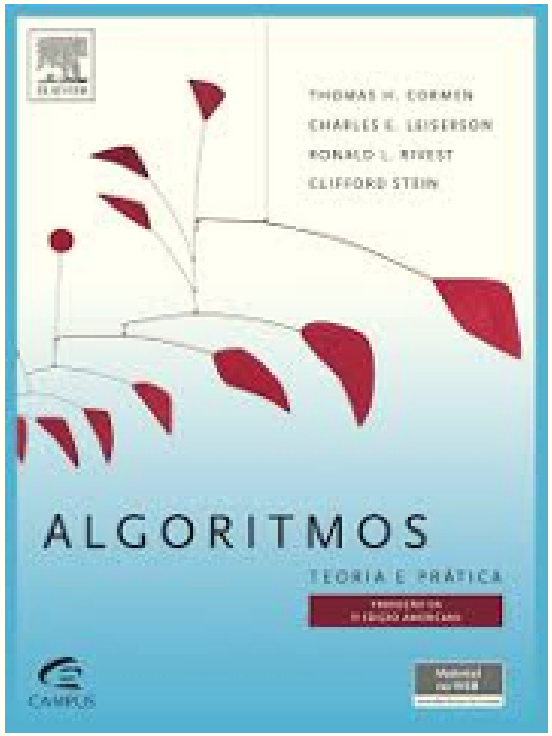
- Aulas, Avaliações, Bibliografia, Ementa
- Introdução

Aulas, Avaliações, Bibliografia, Ementa

Aulas e Avaliações

- Aulas (>= 75% presença):
  - FEV(14, 21, 27);
  - MAR(07, 14, 21, 28);
  - ABR(04, 11, 18, 25);
  - MAI(02, 09, 16, 23, 30);
  - JUN(06, 13, 27);
- Avaliações (50% nota):
  - ABR(04);
  - JUN(06);
- Trabalhos individuais (50% nota):
  - FEV(14, 21, 27);
  - MAR(07, 14, 21, 28);
  - ABR(11, 18, 25);
  - MAI(02, 09, 16, 23, 30);
- Recuperação: JUN(13, 27);

Bibliografia



Ementa

- Conceitos de algoritmos
- Conceitos de linguagens de programação
- Constantes e Variáveis
- Tipos de Dados
- Operadores
- Expressões Aritméticas, lógicas e literais
- Comandos básicos
- Estruturas condicionais e de repetição
- Modularização
- Recursividade
- Variáveis compostas homogêneas e variáveis compostas heterogêneas
- Estruturas de dados básicas
- Algoritmos e Meio Ambiente

Introdução

Conceitos básicos

Definição de algoritmo (CORMEN, 2009)

Procedimento computacional bem definido que toma algum valor ou conunto de valores como entrada e produz algum valor ou conjunto de valores como saída.

Definição de algoritmo (CORMEN, 2009)

Sequência de etapas computacionais que transformam entrada em saída

Conceitos básicos

Corretude

- Um algoritmo é correto se para toda instância ou entrada, ele parar com a saída correta
- Um algoritmo incorreto pode não parar ou parar com uma resposta incorreta

Existem problemas para os quais não se conhece nenhuma solução eficiente (Não determinísticos polinomiais, NP), caso em que um algoritmo incorreto pode ser útil.

Conceitos básicos

Exemplo: problema da ordenação

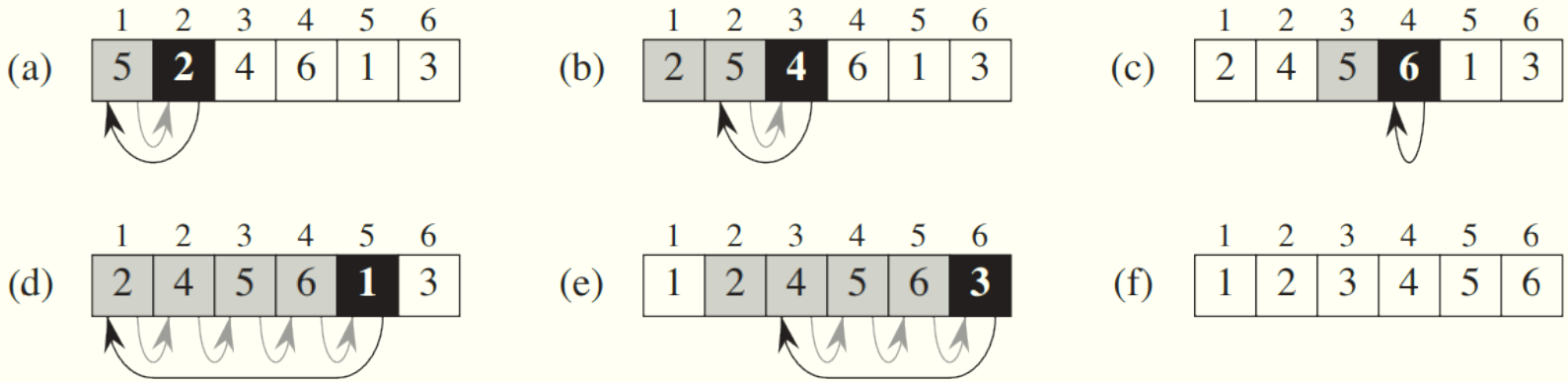
- Entrada: sequência de números $(a_1, a_2, \ldots, a_n)$
- Saída: uma permutação da sequência de entrada $(a'_1, a'_2, \ldots, a'_n) \mid (a_1 \leq a_2 \leq \cdots \leq a_n)$

Seja a instância ou entrada do problema a sequência
(31, 41, 59, 26, 41, 58).
Um algoritmo de ordenação devolve a saída
(26, 31, 41, 41, 58, 59)

Conceitos básicos

INSERTION-SORT(A)

```
1: for j = 2 to |A| do
2:     key = A[j]
3:     //Insert A[j] into the sorted sequence A[1..j − 1]
4:     i = j − 1
5:     while i > 0 and A[i] > key do
6:         A[i + 1] = A[i]
7:         i = i − 1
8:     A[i + 1] = key
```



Conceitos básicos

Análise de algoritmos

Analisar um algoritmo significa prever quando recurso computacional (tempo e espaço) ele irá consumir.

O tempo de execução do algoritmo é a soma dos tempos de execução de cada instrução executada.
Exemplo, uma instrução que demanda c_i passos para executar, contibuirá com $c_i n$ para o tempo de execução total.

Conceitos básicos

Custo de tempo por operação

INSERTION-SORT(A)	*cost*	*times*
1 **for** $j = 2$ **to** $A.length$	c_1	n
2 $key = A[j]$	c_2	$n-1$
3 // Insert $A[j]$ into the sorted sequence $A[1..j-1]$.	0	$n-1$
4 $i = j-1$	c_4	$n-1$
5 **while** $i > 0$ and $A[i] > key$	c_5	$\sum_{j=2}^{n} t_j$
6 $A[i+1] = A[i]$	c_6	$\sum_{j=2}^{n}(t_j-1)$
7 $i = i-1$	c_7	$\sum_{j=2}^{n}(t_j-1)$
8 $A[i+1] = key$	c_8	$n-1$

Custo total de tempo

$$T(n) = c_1 n + c - a(n-1) + c_4(n-1) + c_5 \sum_{j=2}^{n} t_j + c_6 \sum_{j=2}^{n} t_j - 1 + c_7 \sum_{j=2}^{n} t_j - 1 + c_8(n-1)$$

Conceitos básicos

Linguagens de programação

- **Linguagens de programação** são conjuntos de comandos, instruções e sintaxe, com os quais é possível criar um programa de computador
- "*High-level languages*" podem ser compiladas para "*low-level languages*" as quais são reconhecidas diretamente pelo hardware

Conceitos básicos

Linguagem de programação C

- Criada em 1972, por Dennis Ritchie, nos *Bell Telephone Laboratories* para permitir a escrita do Unix
- Chama-se "**C**" porque foi adaptada da linguagem "**B**" dos *Bell Telephone Laboratories*
- American National Standards Institute (ANSI) padronizou-a em 1983

Conceitos básicos

Programando com C

- Criação/Edição do código fonte
- Compilação do programa:
  - Verificação de sintaxe: compilador para ao encontrar um erro
  - Outras verificações: compilador continua e emite um *warnning*
- *Linkagem*: criação de um arquivo objeto (executável)

Conceitos básicos

Tipos de dados

- **Variável** é um nome que damos a uma posição de memória que deverá conter um valor de determinado tipo
- **Tipos básicos** são tipos de valores que pertencem a um domínio. Normalmente, linguagens de programação os definem
- Exemplo: int
  - negativos inteiros
  - positivos inteiros
  - zero

Conceitos básicos

Tipos de dados primitivos em C

- int
- float
- double
- char