

# ALGORITMOS E ESTRUTURAS DE DADOS

## CES-11

Prof. Paulo André Castro

[pauloac@ita.br](mailto:pauloac@ita.br)

Sala 110 – Prédio da Computação

[www.comp.ita.br/~pauloac](http://www.comp.ita.br/~pauloac)

IECE - ITA

## OBJETIVOS GERAIS

- Compreensão da necessidade de uma boa estruturação das informações processadas no computador
- Capacidade de escolher a estrutura de dados mais adequada para uma determinada aplicação
- Capacidade de programar métodos que criam e manipulam essas estruturas
- “Estar alfabetizado” *versus* “saber redigir”



# ALGORITMO

- Informalmente, um algoritmo é qualquer procedimento computacional bem definido que recebe um conjunto de valores como *entrada* e produz outro conjunto de valores como *saída*
- Algoritmo correto
  - Para cada instância de entrada, termina e produz uma saída correta
- Algoritmo incorreto
  - Quando não para (permanece em *loop*)
  - Quando para com uma resposta diferente da desejada



# EXEMPLOS DE ALGORITMOS

## ○ Projeto Genoma Humano

- O DNA humano possui cerca de 100 mil genes e 3 bilhões de pares de bases químicas
  - Necessidade de armazenar os dados e analisá-los
  - Inserção, eliminação e atualização desses dados
  - Ordenação, classificação e pesquisa de informações
  - Correlação entre esses dados
  - Cálculos científicos envolvendo grandes matrizes multidimensionais

## ○ Internet

- Emprega algoritmos para gerenciar e manipular um enorme volume de dados
  - Localização de rotas por onde os dados trafegam
  - Mecanismos de pesquisa de páginas



# EXEMPLOS DE ALGORITMOS

## ○ Comércio eletrônico

- Necessidade de manter informações privadas (número de cartões, senhas, etc.)
  - Criptografia de chave pública e assinaturas digitais
  - Compactação e descompactação

## ○ Indústria e comércio

- Alocação adequada de recursos escassos
  - Onde localizar os poços de petróleo para maximizar o lucro?
  - Onde investir em publicidade? Público? Região?
  - Como designar tripulações de voo de um modo menos dispendioso?
  - Qual melhor investimento: ações, opções, títulos públicos?



# DESENVOLVIMENTO DE UM ALGORITMO

## ○ Características dos algoritmos

- Considere um problema prático
- Existem muitas resoluções candidatas, mas provavelmente não serão bem aquilo que desejamos...

Convém conhecer os algoritmos existentes para então desenvolver o seu

## ■ Qual o melhor algoritmo?

- Basta que forneça a resposta correta?
- Também é importante a sua *eficiência*
  - Tempo de computação
  - Memória utilizada



## ESTRUTURAS DE DADOS

- O consumo de tempo e de memória torna-se extremamente crítico quando o universo de informações é muito grande...
- A eficiente utilização dos recursos computacionais e a redução do tempo de resposta dependem de dois fatores:
  - Boa estruturação das informações
  - Bons algoritmos que manipulem essas estruturas
- Principais modelos para visualizar, interpretar e armazenar dados:
  - Listas lineares
  - Árvores
  - Grafos



## LISTAS LINEARES

- Seus elementos formam uma sequência linear:



- Cada elemento possui um antecessor e um sucessor (exceto o primeiro e o último)
- Tabelas, em geral, se enquadram neste modelo:

Nome	Endereço	Outros

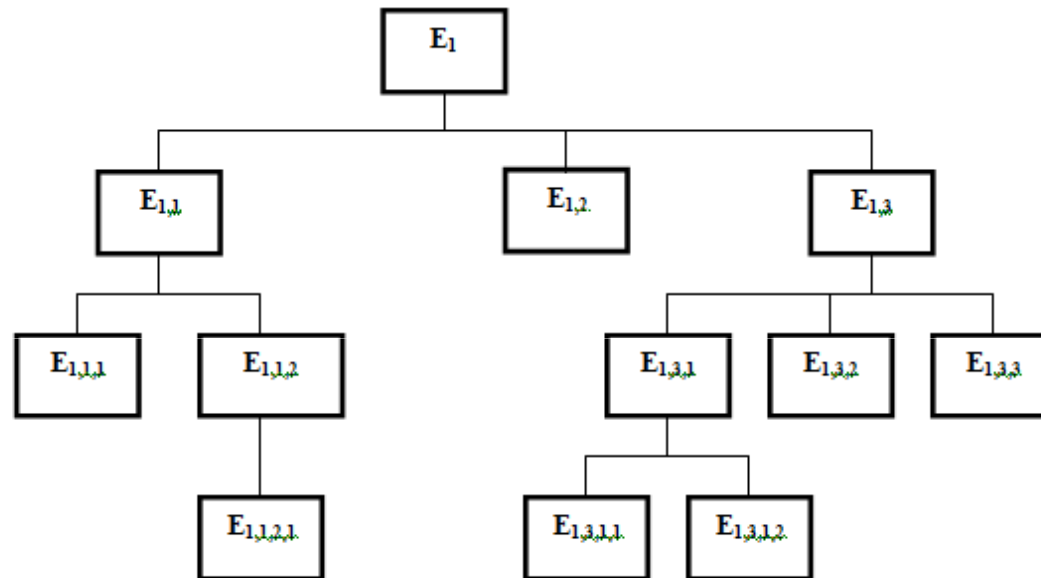
- Listas telefônicas
- Folhas de pagamento
- Livros de uma biblioteca
- Tabelas de um banco de dados relacional





# ÁRVORES

- Há uma hierarquia entre os elementos:



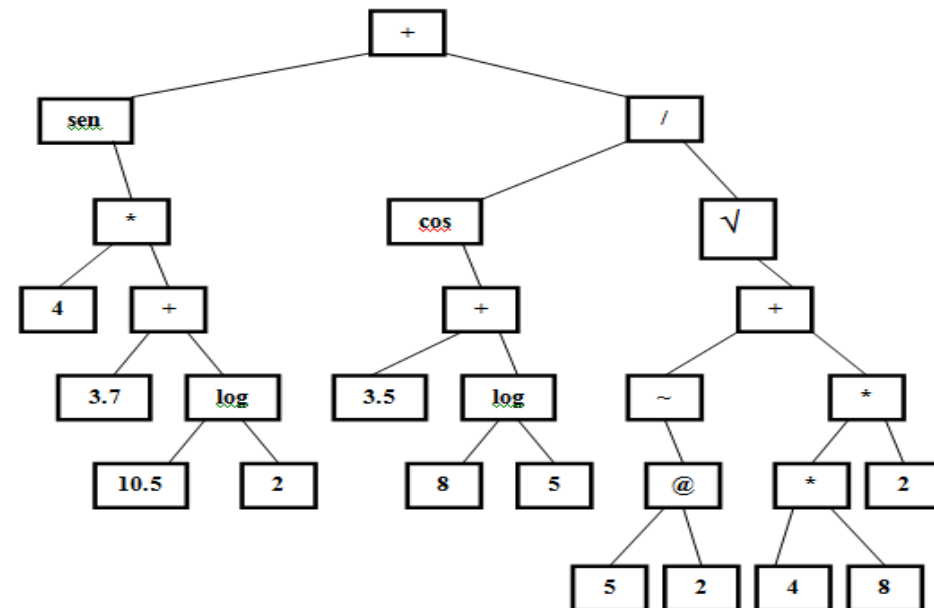
- Cada elemento:
  - Tem um único pai (exceto a raiz)
  - Pode ter vários filhos
  - Não pode ser pai de nenhum ancestral



## EXEMPLOS DE ÁRVORES

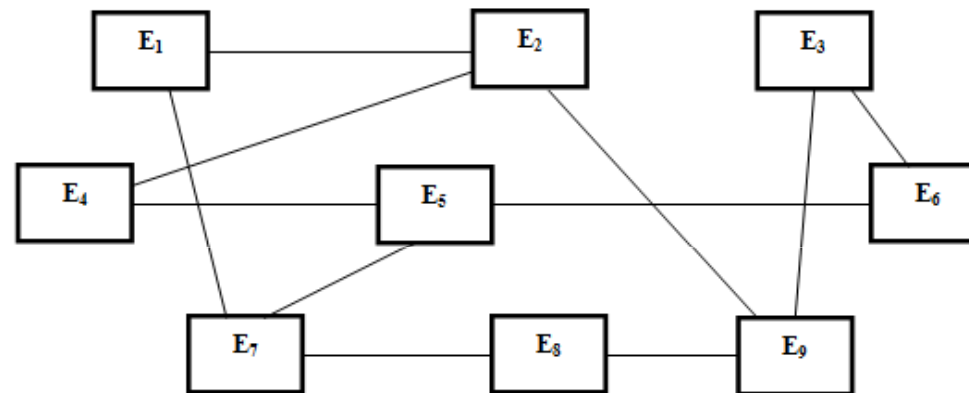
- Organograma de empresas
- Organização de livros e cursos
- Jogos eliminatórios de um campeonato
- Expressões aritméticas

$$\text{sen}(4 * (3.7 + \log_2 10.5)) + \frac{\text{cos}(3.5 + \log_5 8)}{\sqrt{-(5)^2 + 4 * 8 * 2}}$$



# GRAFOS

- Há uma interligação geral entre os elementos, sem formar sequências ou hierarquias:



- Exemplos:
  - Tarefas de um projeto
  - Sistema rodoviário
  - Redes de interconexão
  - Fornecimento de produtos entre fábricas
  - Máquinas de estados finitos
  - etc.



## IMPORTÂNCIA DA DISCIPLINA

- Na Ciência da Computação:
  - Processo de compilação
  - Gerenciamento de programas e de memória
  - Construção de bancos de dados
  - etc.
- Mesmo sem ser um especialista na área, um engenheiro nos dias de hoje deve ser capaz de elaborar algoritmos não triviais.
- Alguns exemplos de problemas em grafos:
  - Caminhos mais curtos
  - Validação de grafos acíclicos
  - Componentes fortemente conexos
  - Pontos de articulação
  - Árvore geradora de custo mínimo

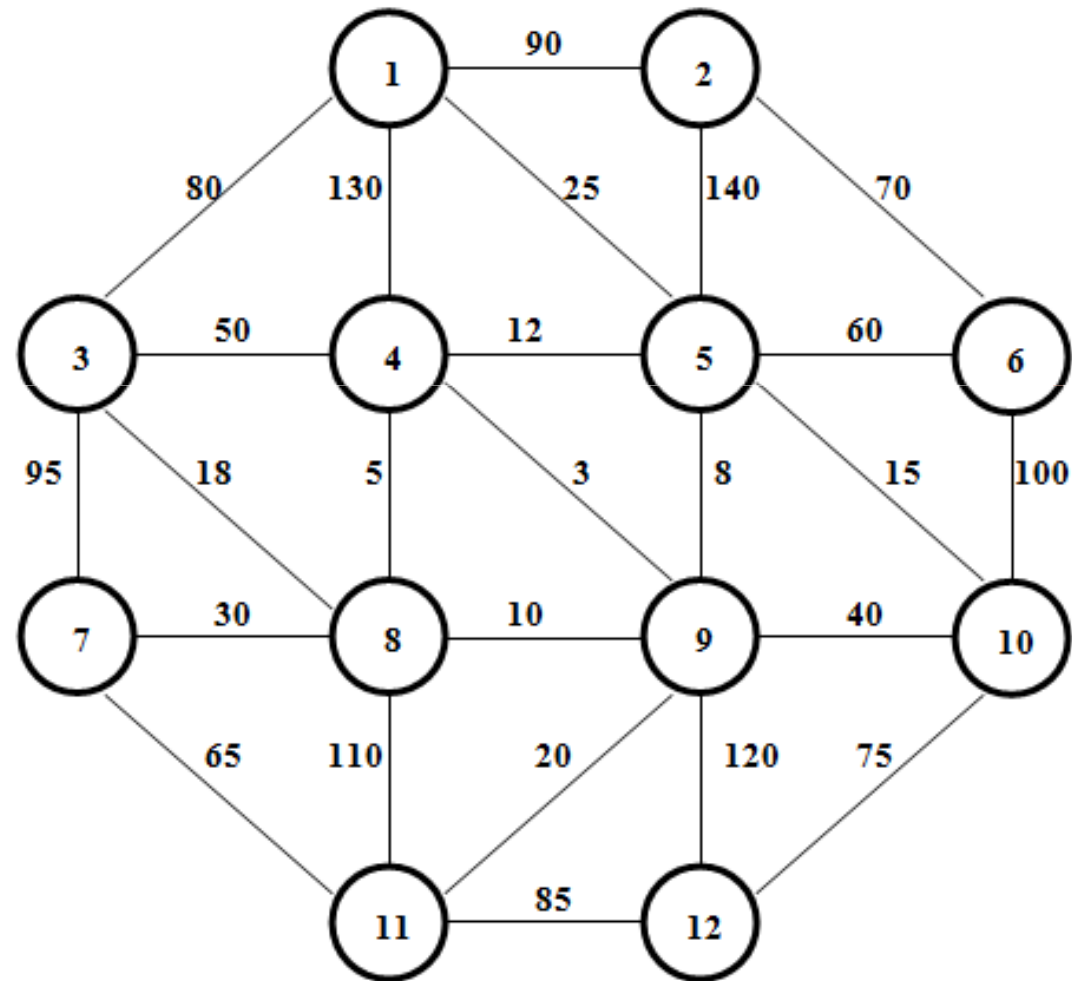


# CAMINHOS MAIS CURTOS

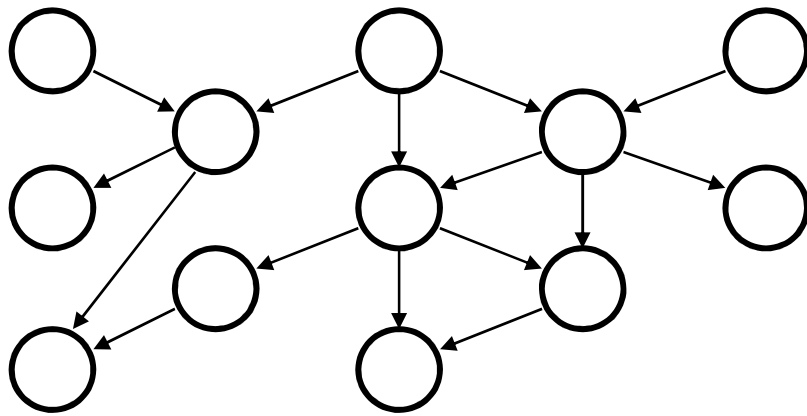
- Distâncias a partir de um dado vértice

- Aplicações:

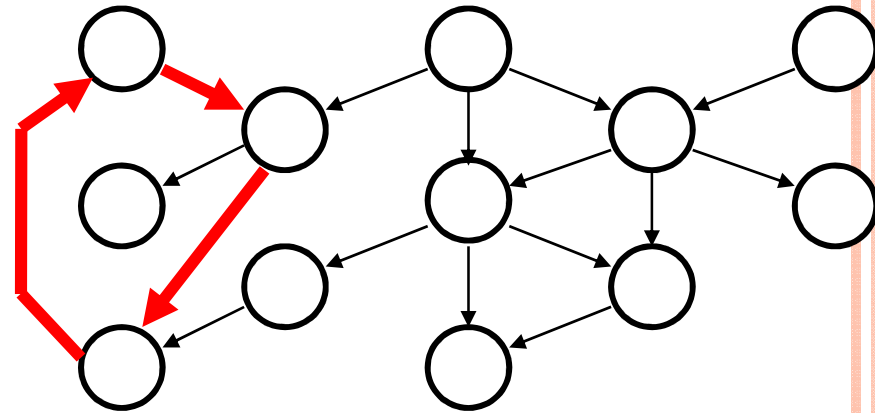
- Transporte terrestre ou aéreo
- Robótica
- Projetos de VLSI



# VALIDAÇÃO DE GRAFOS ACÍCLICOS



*Grafo acíclico*



*Grafo cíclico*

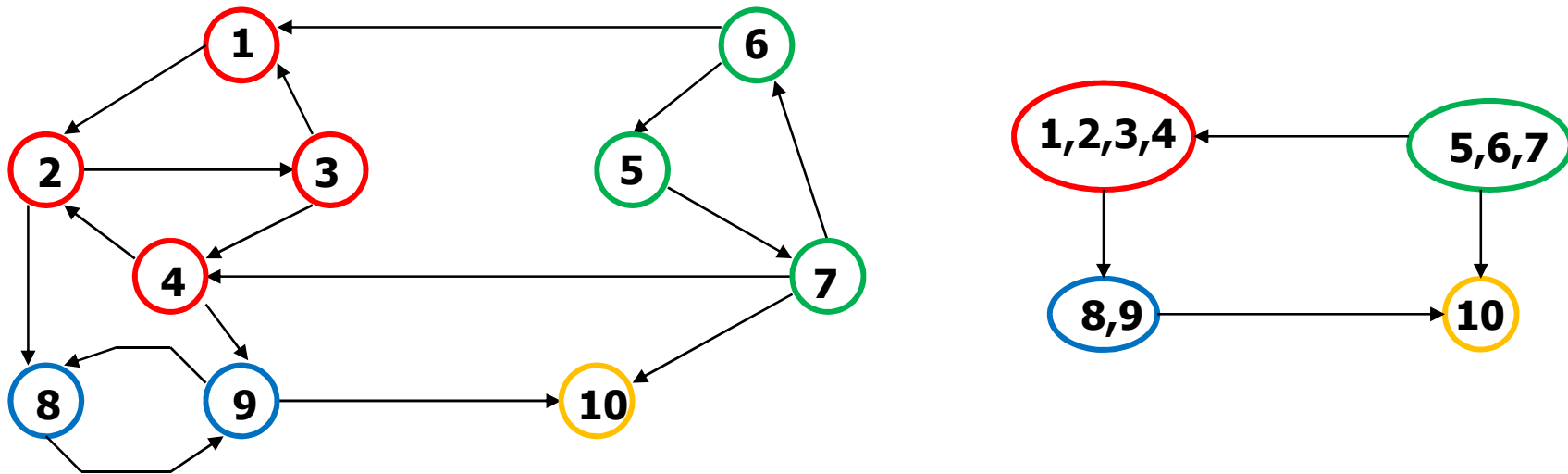
## ○ Aplicações:

- Gerenciamento de projetos (redes PERT-CPM: *Program Evaluation and Review Technique, Critical Path Method*)
- Simulação de circuitos combinacionais



# COMPONENTES FORTEMENTE CONEXOS

- Subconjuntos maximais de vértices onde há caminhos de qualquer vértice a todos os outros.



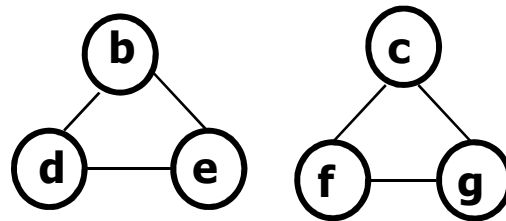
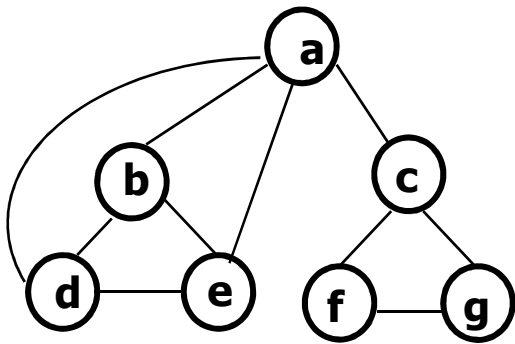
- Aplicações:

- Redução do tamanho de determinados problemas
- Classes de equivalência em circuitos digitais
- Privacidade em sistemas de comunicação

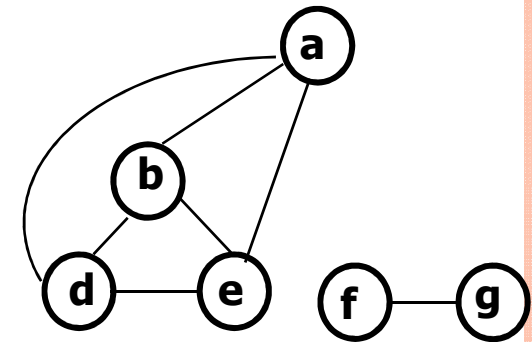


# PONTOS DE ARTICULAÇÃO

- Um vértice é ponto de articulação se, ao ser removido, desconecta o grafo.



**a** é ponto de articulação



**c** é ponto de articulação

## ○ Aplicações:

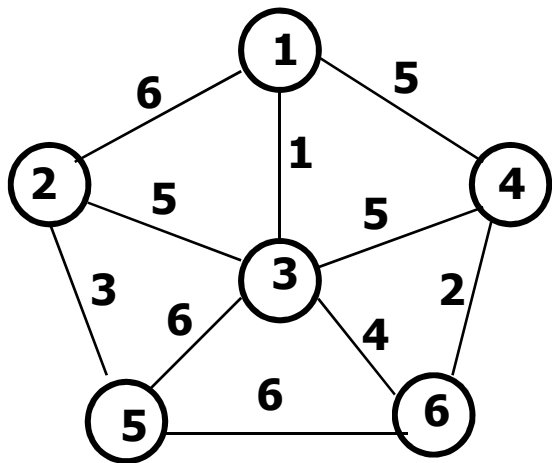
- Redes de interconexão em geral
- Sistemas de transmissão de energia elétrica
- Sistemas de distribuição hidráulica



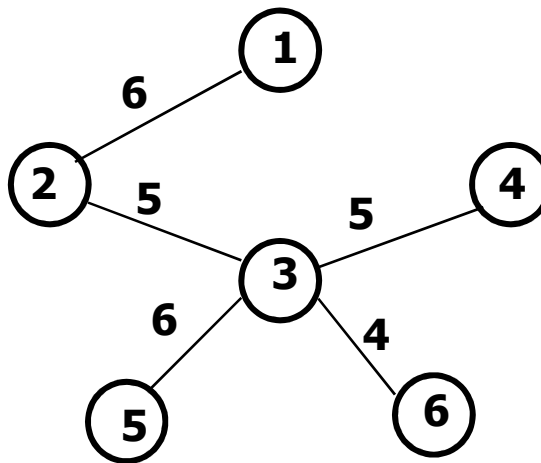


# ÁRVORE GERADORA DE CUSTO MÍNIMO

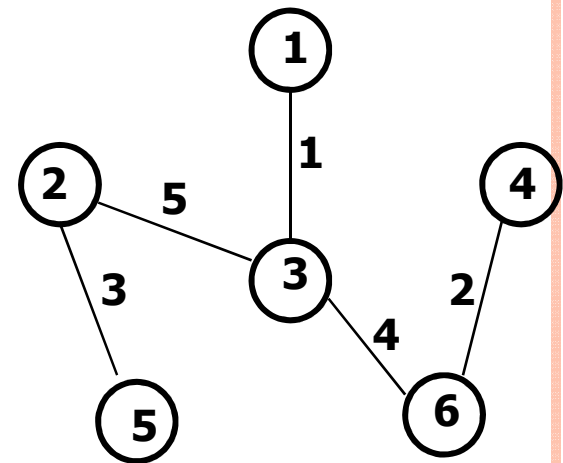
- Às vezes, é necessário encontrar um subgrafo gerador (todos os vértices e um subconjunto das arestas) que seja árvore e que tenha custo mínimo.



*Grafo não orientado*



*Árvore de custo 26*



*Árvore de custo 15  
(mínimo)*

## ○ Aplicações:

- Redes de interconexão em geral



# PLANO DO CURSO

## ○ Primeiro Bimestre:

- Breve revisão de ponteiros
- Noções de complexidade de algoritmos
- Listas lineares
- Pilhas, filas e *deques*
- Árvores
- Árvores binárias

## ○ Segundo Bimestre:

- Algoritmos de ordenação
- Paradigma da Divisão-e-Conquista
- Grafos
  - Representações
  - Soluções de alguns problemas clássicos
- Noções de Programação Orientada a Objetos



## CRITÉRIOS DE AVALIAÇÃO

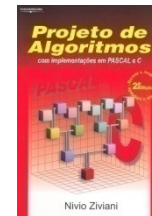
- Por bimestre: 2 provas e de 2 a 5 laboratórios
- **Nota bimestre** = (média provas + média labs) / 2
- Um exame final
- As provas, os laboratórios e o exame são **individuais**.

# BIBLIOGRAFIA

- P. Feofiloff  
*Algoritmos em Linguagem C*



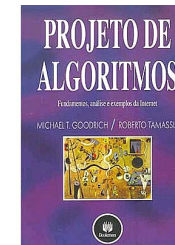
- N. Ziviani  
*Projeto de Algoritmos*



- A.M. Tanenbaum, Y. Langsam, M.J. Augenstein  
*Estruturas de Dados usando C*



- M.T. Goodrich, R. Tamassia  
*Projeto de Algoritmos*

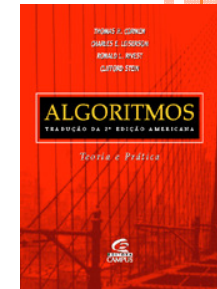
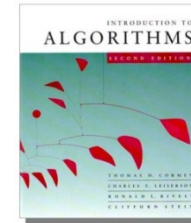


- B.R. Preiss  
*Estruturas de Dados e Algoritmos*



# BIBLIOGRAFIA COMPLEMENTAR

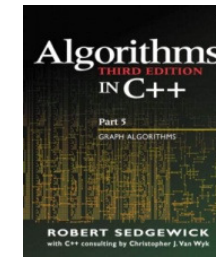
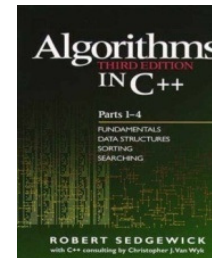
- **T.H. Cormen, C.E. Leiserson, R.L. Rivest**  
*Algoritmos : Teoria e Prática*



- **A. Drozdek**  
*Estrutura de Dados e Algoritmos em C++*



- **R. Sedgwick**  
*Algorithms in [C, C++, Java]*



- **A. V. Aho, J. E. Hopcroft, J. D. Ullman**  
*Data Structures and Algorithms*

