

Algoritmos e Estruturas de Dados II



PUC Minas

Instituto de Ciências Exatas e Informática
Departamento de Ciência da Computação

Contextualização

- **Algoritmos e Estruturas de Dados I:** Representação e armazenamento de dados.

Manipulação e movimentação de dados em memória principal e secundária. Abstração de dados. Estruturas e abstração de controle. Modularização, Encapsulamento e Herança. Recursividade. Documentação e testes. Implementação em linguagem de programação.

- **Introdução à Computação:** Pesquisa, Ensino, Extensão, Inovação e Mercado. História da computação. Fundamentos de Computabilidade. Linguagens de programação. Principais áreas da Computação. Computação Verde. Práticas de extensão.

- **Cálculo I:** Funções: polinomiais, racionais, algébricas, exponenciais, logarítmicas, trigonométricas. Limites. Continuidade. Derivada: definição e interpretações. Regras de derivação. Derivação implícita. Aplicações da derivada: taxas relacionadas, regra de L'Hospital, estudo do comportamento de funções, esboço de gráficos e otimização. Aplicações práticas.

• Somatórios. Fundamentos de análise de algoritmos. Ordenação e pesquisa em memória principal. Tipos abstratos de dados lineares e flexíveis. Árvores. Balanceamento de árvores. Tabelas. Dicionários.

Objetivos

- Fazer com que o aluno desenvolva habilidade de construir programas eficientes por meio da estruturação de dados e da aplicação de algoritmos de ordenação e pesquisa em memória principal
- Propiciar um ambiente no qual o aluno avance no desenvolvimento das habilidades de construção, teste e documentação de programas
- Dar condições para que o aluno desenvolva competências para comparar criticamente diversas soluções para problemas computacionais
- Levar o aluno a compreender os aspectos mais importantes da manipulação de dados em memória dinâmica
- Possibilitar a integração das disciplinas do núcleo de matemática e programação de computadores, por meio do desenvolvimento de problemas práticos

Métodos Didáticos

- Aulas expositivas com apresentação de conteúdo e discussão de problemas e aplicações
- Desenvolvimento de algoritmos de forma dinâmica durante as aulas
- Revisões de exemplos e atividades práticas que possam estimular o desenvolvimento de uma análise crítica das diversas técnicas estudadas
- Exercícios extraclasse, provas e trabalhos práticos individuais e em grupos
- Aprendizagem por meio de solução de problemas
- Estudos-de-casos que realcem a importância da disciplina e sua aplicação em problemas reais
- Trabalhos práticos e exercícios extraclasse para aprendizado aprofundado dos conceitos e técnicas estudadas

Unidades de Ensino

- Unidade 0: Nivelamento (EAD)
- Unidade I: Fundamentos de análise de algoritmos
- Unidade II: Ordenação em memória principal
- Unidade III: Estruturas de dados básicas lineares
- Unidade IV: Estruturas de dados básicas flexíveis
- Unidade V: Árvores binárias
- Unidade VI: Balanceamento de árvores
- Unidade VII: Tabelas e dicionários
- Unidade VIII: Árvores TRIE

Unidades de Ensino

- **Unidade 0: Nivelamento (EAD)**
 - Unidade I: Fundamentos de análise
 - Unidade II: Ordenação em memória
 - Unidade III: Estruturas de dados
 - Unidade IV: Estruturas de dados
 - Unidade V: Árvores binárias
 - Unidade VI: Balanceamento de árvores
 - Unidade VII: Tabelas e dicionários
 - Unidade VIII: Árvores TRIE
- Revisão de AEDs I
 - **Introdução ao Linux (vídeo)**
 - Introdução às linguagens C, C++ e Java
 - Arquivos em C, C++ e Java
 - Introdução à Orientação por Objetos
 - Recursividade
 - Tratamento de exceção
 - Ponteiros e referência
 - Argumento do Método Main
 - *Encoding*
 - Redirecionamento de entrada e saída
 - **Processo AEDs II de Exercícios (vídeo)**

Unidades de Ensino

- Unidade 0: Nivelamento (EAD)
- **Unidade I: Fundamentos de análise de algoritmos**
- Unidade II: Ordenação em memória
- Unidade III: Estruturas de dados
- Unidade IV: Estruturas de dados
- Unidade V: Árvores binárias
- Unidade VI: Balanceamento de árvores
- Unidade VII: Tabelas e dicionários
- Unidade VIII: Árvores TRIE

- Plano de Ensino
- **Noções de complexidade (híbrido)**
- **Pesquisa sequencial e binária (vídeo)**
- Algoritmo de ordenação por seleção
- **Somatórios (híbrido)**
- **Contagem de operações (híbrido)**
- **Aspectos da análise de algoritmos (híbrido)**
- **Função de complexidade (híbrido)**
- **Notações O , Ω e Θ (híbrido)**

Unidades de Ensino

- Unidade 0: Nivelamento (EAD)
- Unidade I: Fundamentos de análise de algoritmos
- **Unidade II: Ordenação em memória principal**
- Unidade III: Estruturas de dados básicas lineares
- Unidade IV: Estruturas de dados básicas flexíveis
- Unidade V: Árvores binárias
- Unidade VI: Balanceamento de árvores
- Unidade VII: Tabelas e dicionários
- Unidade VIII: Árvores TRIE

- **Método da bolha (vídeo)**
- Método de Inserção
- **Shellsort (vídeo)**
- Quicksort
- Heapsort
- Mergesort
- **Countingsort (vídeo)**
- **Radixsort (vídeo)**
- Comparação entre os métodos
- **Ordenação Parcial (vídeo)**
- **Algoritmos Paralelos de Ordenação (vídeo)**

Unidades de Ensino

- Unidade 0: Nivelamento (EAD)
- Unidade I: Fundamentos de análise de algoritmos
- Unidade II: Ordenação em memória principal
- **Unidade III: Estruturas de dados básicas lineares**
- Unidade IV: Estruturas de dados básicas flexíveis
- Unidade V: Árvores binárias
- Unidade VI: Balanceamento de árvores
- Unidade VII: Tabelas e dicionários
- Unidade VIII: Árvores TRIE

- Lista
- Pilha
- Fila circular

Unidades de Ensino

- Unidade 0: Nivelamento (EAD)
 - Unidade I: Fundamentos de análise de algoritmos
 - Unidade II: Ordenação em memória principal
 - Unidade III: Estruturas de dados básicas lineares
 - **Unidade IV: Estruturas de dados básicas flexíveis**
 - Unidade V: Árvores binárias
 - Unidade VI: Balanceamento de árvores
 - Unidade VII: Tabelas e dicionários
 - Unidade VIII: Árvores TRIE
- Pilha em Java
 - Fila em Java
 - Lista simples e dupla em Java
 - Matriz
 - Pilha, fila e lista em C
 - Coleta de lixo em Java
 - TADs Nativos em Java

Unidades de Ensino

- Unidade 0: Nivelamento (EAD)
 - Unidade I: Fundamentos de ar
 - Unidade II: Ordenação em me
 - Unidade III: Estruturas de dado
 - Unidade IV: Estruturas de dado
 - **Unidade V: Árvores binárias**
 - Unidade VI: Balanceamento de
 - Unidade VII: Tabelas e dicioná
 - Unidade VIII: Árvores TRIE
- Definições e conceitos
 - Tipo Nó em Java
 - Inserção em Java com retorno de referência
 - Pesquisa
 - Remoção
 - Caminhamento
 - Inserção em Java com passagem de pai
 - Inserção em C com ponteiro
 - Inserção em C++ com passagem por referência
 - Estruturas híbridas

Unidades de Ensino

- Unidade 0: Nivelamento (EAD)
 - Unidade I: Fundamentos de análise de algoritmos
 - Unidade II: Ordenação em memória principal
 - Unidade III: Estruturas de dados básicas lineares
 - Unidade IV: Estruturas de dados básicas flexíveis
 - Unidade V: Árvores binárias
 - **Unidade VI: Balanceamento de árvores**
 - Unidade VII: Tabelas e dicionários
 - Unidade VIII: Árvores TRIE
- Tipos de rotação
 - Árvore AVL
 - Árvore 2-3-4
 - Árvore Bicolor

Unidades de Ensino

- Unidade 0: Nivelamento (EAD)
 - Unidade I: Fundamentos de análise de algoritmos
 - Unidade II: Ordenação em memória principal
 - Unidade III: Estruturas de dados básicas lineares
 - Unidade IV: Estruturas de dados básicas flexíveis
 - Unidade V: Árvores binárias
 - Unidade VI: Balanceamento de árvores
 - **Unidade VII: Tabelas e dicionários**
 - Unidade VIII: Árvores TRIE
- Tabela *Hash* Direta com Reserva
 - Tabela *Hash* Direta com *Rehash*
 - Tabela *Hash* Indireta com Lista
 - Dicionários, conjuntos e mapas

Unidades de Ensino

- Unidade 0: Nivelamento (EAD)
- Unidade I: Fundamentos de análise de algoritmos
- Unidade II: Ordenação em memória principal
- Unidade III: Estruturas de dados básicas lineares
- Unidade IV: Estruturas de dados básicas flexíveis
- Unidade V: Árvores binárias
- Unidade VI: Balanceamento de árvores
- Unidade VII: Tabelas e dicionários
- **Unidade VIII: Árvores TRIE**
 - Tipo Nó
 - Lista flexível
 - Árvore Balanceada
 - Tabela *Hash*
 - Árvore TRIE
 - Árvore TRIE PATRICIA

Métodos de Avaliação

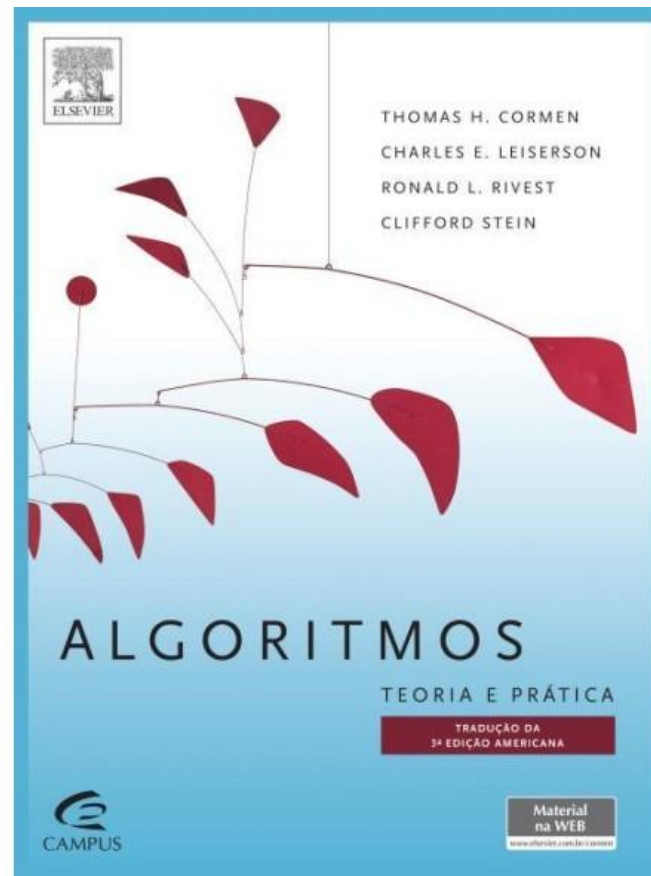
- Prova P1, P2 e P3 – 20 pontos cada
- ADA – 5 pontos
- Trabalhos – 20 pontos onde $NOTA_{[Trabalhos\ Práticos]} \times NOTA_{[Trabalhos\ Teóricos]}$
- Fator Quiz (F_Q) – 1.4 de quizzes a serem usados no desempenho
- Desempenho – 15 pontos

$$\text{Desempenho} = \frac{(P_{\text{MAIOR}} + P_{\text{SEGUNDAMAIOR}})}{40} \times \frac{\text{Trabalhos}}{20} \times 15 \times F_Q$$

- Reavaliação – 20 pontos para substituir a nota da menor prova sendo que o aluno aprovado na reavaliação terá nota igual a 60

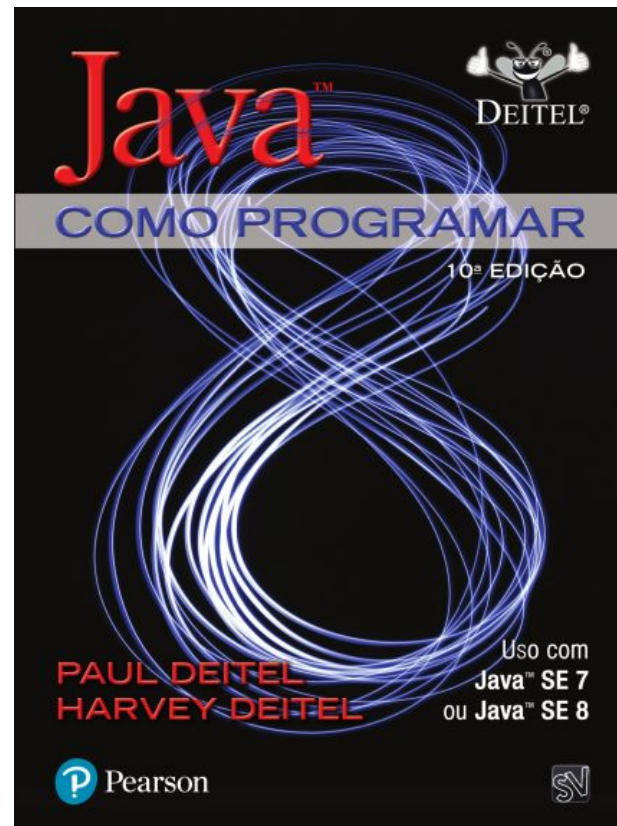
Bibliografia Básica

- CORMEN, T.H., LEISERSON, C.E., RIVEST, R.L, STEIN, C.;
Algoritmos: Teoria e Prática; Editora Campus; 3ª Edição; 2012



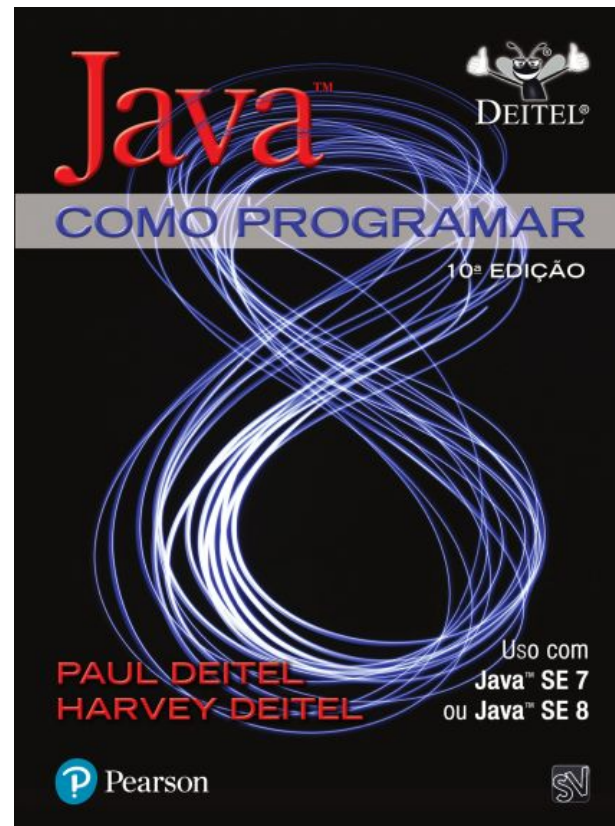
Bibliografia Básica

- DEITEL, H. M.; DEITEL, P. J. Java: Como programar. 10^a edição. Pearson Prentice Hall, 2016



Bibliografia Básica

- DEITEL, H. M.; DEITEL, P. J. Java: Como programar. 10ª edição. Pearson Prentice Hall, 2016



Bibliografia Básica

- ZIVIANI, N. Projeto de algoritmos: Com implementações em Java e C++. Pioneira Thomson Learning, 2006



Bibliografia Básica

- GRAHAN, J., KNUTH, D., PATASHNIK, O.; Matemática Concreta: Fundamentos para a Ciência da Computação; LTC; 2a edição; 1995



Bibliografia Básica

- SCHILDT, H. C Completo e Total. 3ª edição. Pearson, 1997



Trabalho Teórico I

- Todos os exercícios de Nivelamento (unidade 0), entregar um arquivo zip