

MINISTÉRIO DA EDUCAÇÃO

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA CAMPUS PETRÓPOLIS

CURSO SUPERIOR DE ENGENHARIA DE COMPUTAÇÃO

DEPARTAMENTO		PLANO DE CURSO DA DISCIPLINA			
ENGENHARIA DE COMPUTAÇÃO		ALGORITMOS E ESTRUTURAS DE DADOS II			
CÓDIGO	PERÍODO	ANO	SEMESTRE	PRÉ-REQUISITOS	
GCOM5033PE	4	2014	1		
CRÉDITOS	AULAS/SEMANA			TOTAL DE AULAS NO SEMESTRE	Algoritmos e Estrutura de Dados II
6	TEÓRICA	PRÁTICA	ESTÁGIO	108	
	4	2	0		

EMENTA

1. Árvores binárias de busca balanceadas: AVL, Árvore rubro-negra.
2. Árvores de busca geral: árvores B e suas variações.
3. Teoria dos grafos: introdução, histórico, definição formal, teorema do aperto de mãos
 - a. Tipos de grafos: grafo completo, regular, conexo e desconexo, isomorfismo, complemento, bipartido, grafos Eulerianos e Hamiltonianos.
 - b. Representação computacional
 - c. Isomorfismo, subgrafos, métricas/medidas em grafos, passeio, cadeia, caminho, ciclo.
 - d. Alcançabilidade
 - e. Conexidade
 - f. Planaridade
4. Algoritmos em grafos
 - a. Buscas em grafos, ciclos Eulerianos e Hamiltonianos.
 - b. Árvore geradora, árvore geradora de custo mínimo, algoritmos de Prim, Kruskal.
 - c. Caminho mínimo: Bellman-Ford, Dijkstra, Floyd-Warshall.
5. Divisão e Conquista
6. Backtracking

BIBLIOGRAFIA

Básica:

1. CORMEN, T.H. et al. Algoritmos: teoria e prática. Rio de Janeiro: Campus. Tradução da 3ª edição americana. 2012.
2. SZWARCFITER, J.L.; MARKENZON, L. Estrutura de dados e seus algoritmos. 3ª edição. LTC, 2014.
3. NETTO BOAVENTURA, P.O. Grafos: teoria, modelos, algoritmos. 5ª edição. Blucher, 2011.

Complementar:

1. GOODRICH, M.T.; TAMASSIA, R. Estruturas de Dados & Algoritmos em Java. 4ª edição. Bookman, 2007.
2. GOLDBARG, M.C.; Goldbarg E. Grafos: Conceitos, algoritmos e aplicações. Campus, 2012.
3. TENENBAUM, A.M.; LANGSAM, Y.; AUGENSTEIN, M.J. Estruturas de dados usando C. São Paulo: Makron Books, 2005.

4. ROCHA, A. A. Estruturas de dados e algoritmos em C. 3ª edição revista e aumentada. FCA, 2014.
5. UNGNICKEL, D. Graphs, Networks and Algorithms. 4ª edição. Springer, 2013.

OBJETIVOS GERAIS

O objetivo da disciplina é apresentar ao aluno a teoria e o funcionamento das principais estruturas de dados avançadas (árvores balanceadas, grafos), capacitando o aluno nos conhecimentos das diversas estruturas de dados e principais algoritmos para suas implementações. Aprender os elementos de complexidade em algoritmos, manipulação de dados em disco, de forma a capacitá-los na criação de programas corretos e eficientes, com o uso das estruturas e algoritmos adequados. Compreender e planejar soluções otimizadas no uso das estruturas de dados.

METODOLOGIA

Cada aula consistirá na combinação adequada de:

- Exposição detalhada de cada estrutura de dados.
- Construção de estruturas de dados adequadas a cada problema proposto.
- Exercícios, atividades e estudos de casos, de forma individual ou em grupo.
- Trabalhos práticos de implementação das técnicas ensinadas.
- Provas individuais.

CRITÉRIO DE AVALIAÇÃO

A avaliação será feita através de duas provas teórica, correspondendo a 60% da nota final e dois trabalhos práticos, juntamente com seu relatório correspondendo a 40% dos pontos da disciplina.

CHEFE DO DEPARTAMENTO

NOME	ASSINATURA
Laura Silva de Assis	

PROFESSOR RESPONSÁVEL PELA DISCIPLINA

NOME	ASSINATURA
Laura Silva de Assis	

APROVADO PELO CONSELHO DEPARTAMENTAL EM:

___/___/___

CONTEÚDO PROGRAMÁTICO

1. Apresentação da disciplina:
 - 1.1. Programa;
 - 1.2. Avaliação;
 - 1.3. Datas importantes;
 - 1.4. Onde encontrar informação, prazos, carga horária;
 - 1.5. Bibliografia.
2. Árvores binárias de busca balanceadas:
 - 2.1. Árvores AVL: apresentação, características, motivação, operações de busca, inserção e remoção, análise de complexidade, exercícios.
 - 2.2. Árvores Rubro-Negra: apresentação, características, motivação, operações de busca, inserção e

remoção, análise de complexidade, exercícios.

3. Árvores de busca balanceadas:
 - 3.1. Árvores B: apresentação, características, motivação, operações de busca, inserção e remoção, análise de complexidade, exercícios;
 - 3.2. Variações de árvores B: árvores B* e árvores B+. Características, motivação, operações de busca, inserção e remoção, exercícios.
4. Introdução a teoria de grafos:
 - 4.1. Conceitos básicos, vértices, arestas, definição formal, exemplos, histórico;
 - 4.2. Representação:
 - 4.2.1. Geométrica;
 - 4.2.2. Matemática;
 - 4.2.3. Computacional: matriz de adjacência, matriz de incidência e lista de adjacência;
 - 4.2.4. Exercícios.
 - 4.3. Tipos de grafos, características:
 - 4.3.1. Trivial, simples, completo, regular, ponderado, com atributos;
 - 4.3.2. Subgrafos, complementar, multigrafo, conexo;
 - 4.3.3. Maximal, minimal, planares;
 - 4.3.4. Grafos bipartidos, grafos eulerianos e algoritmos de Fleury e Hierholzer, grafos hamiltonianos.
 - 4.3.5. Exercícios.
 - 4.4. Operações e métricas:
 - 4.4.1. Adição e remoção de vértices e arestas, contração de vértices e arestas;
 - 4.4.2. Desdobramento, inversão, união, soma, produto cartesiano;
 - 4.4.3. Cintura, excentricidade, raio, diâmetro, centro, vizinhança, fechos transitivos.
 - 4.4.4. Exercícios.
5. Algoritmos em grafos:
 - 5.1. Busca em grafos: busca em largura (BFS) e busca em profundidade (DFS), exercícios;
 - 5.2. Árvores: árvores em grafos e seus teoremas, árvore geradora mínima, algoritmos de Prim e Kruskal, exercícios;
 - 5.3. Caminhos mínimos: problemas de caminhos mínimos, algoritmos de Dijkstra, Bellman-Ford e Floyd-Warshall, exercícios;
6. Técnicas de programação:
 - 6.1. Backtracking:
 - 6.1.1. Ideia da força bruta e melhoria com o backtracking, exemplos de problemas comparando as duas técnicas;
 - 6.1.2. Construção da árvore de estados;
 - 6.1.3. Ideia geral do algoritmo de backtracking;
 - 6.1.4. Exemplo de problemas e proposta de solução utilizando backtracking, mostrando pseudo-código e a árvore de estados;
 - 6.1.5. Problemas: ciclo hamiltoniano, mochila, N rainhas, passeio do cavalo, labirinto;
 - 6.1.6. Exercícios
 - 6.2. Divisão e conquista:
 - 6.2.1. Conceitos, definições, etapas;
 - 6.2.2. Algoritmo genérico, vantagens e desvantagens;
 - 6.2.3. Exemplos de problemas mostrando o pseudo-código de como resolvê-los através de divisão e conquista.
 - 6.2.4. Exercícios