



UNIVERSIDADE FEDERAL DE UBERLÂNDIA



FICHA DE COMPONENTE CURRICULAR

CÓDIGO:	COMPONENTE CURRICULAR: ESTRUTURAS DE DADOS II	
UNIDADE ACADÊMICA OFERTANTE: FACULDADE DE COMPUTAÇÃO		SIGLA: FACOM
CH TOTAL TEÓRICA: 30 horas	CH TOTAL PRÁTICA: 30 horas	CH TOTAL: 60 horas

1. **OBJETIVOS**

Capacitar o aluno a desenvolver soluções computacionais eficientes para problemas que necessitam de algoritmos e/ou estruturas de dados avançadas. Discutir o custo computacional de algoritmos. Utilizar adequadamente solução iterativa e recursiva. Utilizar árvores, grafos e tabelas *hash*.

2. **EMENTA**

Árvores: Tipos, Árvore de Busca Binária (ABB), Balanceamento de ABB. Grafos. *Hashing*.

3. **PROGRAMA****1. Árvores**

1.1. Tipos de árvores e representação.

1.2. Árvore binária.

1.2.1. Tipo Abstrato de Dados.

1.2.2. Árvore de sintaxe abstrata: travessia em-ordem, pré-ordem e pós-ordem.

1.2.3. Atributos: Nível, Altura, Comprimento de Caminho.

1.2.4. Algoritmos (cálculo da altura e comprimento de caminho) e operações básicas.

1.2.5. Implementação de TAD Árvore Binária em Linguagem Python.

1.3. Árvore Binária de Busca.

1.3.1. Árvore AVL.

1.3.2. Fator de Balanceamento.

1.3.3. Operações: Inserção/Remoção de nós e implementação em Linguagem Python.

1.3.4. Análise e custo dos algoritmos.

2. Grafo

2.1. Definição e aplicações.

2.2. Grau de vértice, adjacência, incidência, grau de um vértice, caminho, ciclo, densidade, grafo conectado, grafo completo.

2.3. Influência da densidade do grafo em algoritmos.

2.4. Tipo Abstrato de Dados.

2.4.1. Operações básicas: criação/destruição de um grafo, inserção/remoção de vértices e arestas, apresentação de vértices e arestas, apresentação do número de vértices, cópia de um grafo.

2.5. Representações.

2.5.1. Matriz de Adjacência.

2.5.2. Lista de Adjacência.

2.5.3. Comparação entre Matriz X Lista: Espaço e Tempo.

2.5.4. Implementação de TAD Grafo em Python.

2.6. Caminhamentos em grafos: largura e profundidade.

2.7. Menor caminho entre dois vértices.

3. Hash

3.1. O que é Hash.

3.2. Funções de Hashing.

3.2.1. Chaves: número real, número inteiro, string.

3.3. Tratamento de Colisão.

3.3.1. Lista de colisões.

3.3.2. Sondagem Linear.

3.3.3. Duplo Hash.

4. BIBLIOGRAFIA BÁSICA

1. CORMEN, T. H. **Algoritmos: Teoria e Prática**. Rio de Janeiro: Campus, 2002.

2. GOODRICH, M.T.; TAMASSIA, R.; GOLDWASSER, M.H. **Data structures and algorithms in Python**. John Wiley & Sons Ltd, 2013.

3. SEDGEWICK, R. **Algorithms**. 4th ed. Upper Saddle River: Addison-Wesley, c2011.

5. BIBLIOGRAFIA COMPLEMENTAR

1. BOAVENTURA NETTO, P. O. **Grafos: teoria, modelos, algoritmos**. 4. ed. rev. ampl São Paulo: E. Blucher, 2006.

2. DROZDEK, A. **Estruturas de dados e algoritmos em C++**. São Paulo: Cengage Learning, 2002.

3. SEDGEWICK, R. **Algorithms in C**. 3rd. ed. Boston: Addison-Wesley, 2002.

4. TENENBAUM, A.M.; LANGSAM, Y.; AUGENSTEIN, M.J. **Estrutura de Dados Usando C**. São Paulo: Makron Books, 1995.

5. ZIVIANI, N. **Projeto de Algoritmos**. 2.ed. São Paulo: Pioneira , 2004.

6. APROVAÇÃO

Prof. Dr. Jean Carlos Domingos
Coordenador do Curso de Graduação em Gestão
da Informação

Prof. Dr. Mauricio Cunha Escarpinati
Diretor da Faculdade de Computação



Documento assinado eletronicamente por **Jean Carlos Domingos, Coordenador(a)**, em 18/11/2019, às 15:53, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Mauricio Cunha Escarpinati, Diretor(a)**, em 18/11/2019, às 17:06, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1660380** e o código CRC **781A541C**.

Referência: Processo nº 23117.094672/2019-84

SEI nº 1660380