

**CENTRO UNIVERSITÁRIO FAESA****PLANO DE ENSINO****1. IDENTIFICAÇÃO**

INSTITUIÇÃO:	CENTRO UNIVERSITÁRIO FAESA		
CURSO:	CIÊNCIA DA COMPUTAÇÃO E ENGENHARIA DA COMPUTAÇÃO	ANO/SEMESTRE:	2020/1
DISCIPLINA:	COMPLEXIDADE DE ALGORITMOS	CARGA HORÁRIA:	80 H/A

**2. EMENTA**

Complexidade de Algoritmo: Medida de tempo de execução de um algoritmo. Comportamento assintótico de funções: As notações big oh, ômega e theta. Relações de recorrências. Método da Divisão e Conquista. Programação dinâmica; Algoritmos Gulosos. NP-Completeness: Codificação; Problemas de decisão, localização e otimização; Classes P e NP; Análise da complexidade de algoritmos clássicos na área da computação

**3. OBJETIVOS GERAIS**

- Introduzir os conceitos da teoria da NP-Completeness de problemas de modo a que o aluno possa avaliar o processo de modelagem e solução computacional dos problemas algorítmicos, entendendo seus limites e possibilidades.
- Apresentar modelos de análise de eficiência e eficácia de algoritmos.
- Aplicar em vários contextos os conceitos teóricos desenvolvidos.
- O aluno deverá ser capaz de prever o comportamento de um algoritmo antes de implementá-lo.
- Apresentar diversas técnicas de programação;

**4. CONTEÚDOS****Unidade 1 - Medidas de complexidade e análise assintótica****1.1 Introdução****1.2 Análise de complexidade algoritmos não recursivos no melhor caso, pior caso e caso médio.****1.3 Comportamento assintótico de funções; A notação O, ômega e theta****Unidade 2 – Análise de algoritmos recursivos e relações de recorrência****2.1 Análise de algoritmos de seleção e de outros algoritmos recursivos.****2.2 Métodos de resolução de relação de recorrência****2.2.1 Desdobramento****2.2.2 Árvore****2.2.3 Teorema mestre****2.4 Modelagem de problemas usando relações de recorrência****Unidade 3 – Problemas P e NP****3.1 Introdução****3.2 Classes P, NP e NPC****3.3 Problemas de decisão, localização e otimização**

3.4 Caráter NP-completo e redutibilidade

3.5 Visão geral das técnicas para mostrar que um problema é NPC

#### **Unidade 4 – Técnicas de programação**

4.1 Método da Divisão e Conquista: problema da multiplicação de inteiros e da multiplicação de matrizes, dentre outros.

4.2 Programação dinâmica: Elementos da Programação Dinâmica; PD x Divisão e conquista; Multiplicação de matrizes; subsequência comum mais longa, problema da mochila, dentre outros.

4.3 Algoritmos Gulosos: Elementos da estratégia gulosa; Algoritmos gulosos x PD; Fundamentos teóricos para métodos gulosos; Problema de escalonamento de tarefas;

### **5. AVALIAÇÃO DE APRENDIZAGEM**

O sistema de Avaliação será composto por três notas: C1, C2 e C3. Sendo a média parcial  $MP = (C1 + C2 + C3) / 3$ . O aluno que obtiver nota maior ou igual a 7,0 na MP está aprovado.

O aluno que tiver nota menor que 7,0 e com percentual de presença igual ou superior a 75% deverá fazer a avaliação final (AF). A média final, do aluno que não obteve nota maior que 7,0 na MP, será  $MF = MP * 0,6 + AF * 0,4$ .

Será considerado aprovado, com Avaliação Final, aquele que tiver Média Final (MF), igual ou superior a 5 e 75% ou mais de presenças às aulas letivas.

#### **Avaliações de C1 (Unidade 2):**

- A1 - Avaliação Grupo. Peso 2.
- P1 - Avaliação Individual. Peso 8.

#### **Avaliações de C2 (Unidade 3):**

- A2 - Avaliação Grupo. Peso 2.
- P2 - Avaliação Individual. Peso 8.

#### **Avaliações de C3 (Unidade 4 e 5):**

- A3 - Avaliação Grupo. Peso 4.
- P3 - Avaliação Individual. Peso 6.

#### **Previsão das Avaliações:**

13/3/20	A1
20/3/20	P1
24/4/20	A2
06/5/20	P2
05/6/20	P3
17/6/20	A3

- De 22 a 27 de junho – Prova Substitutiva (a ser marcado pela coordenação)
- De 29 de junho a 4 de julho – Avaliação Final (a ser marcado pela coordenação)

#### **Avaliação Substitutiva:**

- No caso de perder uma avaliação presencial, o aluno poderá fazer a avaliação substitutiva.
- Itens de nota não presenciais que forem perdidos não podem ser substituídos.

## **6. BIBLIOGRAFIA BÁSICA**

- Cormen T. H., Leiserson C. E., Rivest R. L. Algoritmos: Teoria e Prática. Trad. 2ª Ed. americana. Editora Campus 2002.
- U. Manber, Introduction to Algorithms - A creative Approach, Addison-Wesley, 1989.
- Toscani, Laira e VELOSO, Paulo. Complexidade de Algoritmos: análise, projeto e métodos. Sagra-Luzzatto, 2005.

## **7. BIBLIOGRAFIA COMPLEMENTAR**

- Knuth, Donald Ervin. The Art of computer programming. Addison-Wesley, 1998
- Ronald Graham, Donald Knuth, Oren Patashnik. Matemática Concreta - Fundamentos para a Ciência da Computação. Livros Técnicos e Científicos Editora. 1995. (Tradução de Concrete Mathematics - A Foundation for Computer Science, Addison-Wesley, 1994.)
- Terada, R. Desenvolvimento de Algoritmos e Estruturas de Dados. São Paulo, McGraw- Hill do Brasil, 1991.
- Terada, R. & Setzer, V. W. Introdução à Computação e à Construção de Algoritmos. São Paulo, Departamento de Ciência da Computação - IME-USP, 1986.
- ZIVIANI, Nivio. Projeto de algoritmos : Complexidade de algoritmos: com implementações em Java e C++. São Paulo. Thomson, 2007.
- Hazzan, Samuel. Fundamentos de Matemática Elementar, 5 : combinatória, probabilidade. Atual. 7a ed. 2004.