**PLANO DE ENSINO**

**CURSO:** Ciência da Computação

**SÉRIE:** 7º semestre

**DISCIPLINA:** Análise de Algoritmos

**CARGA HORÁRIA SEMANAL:** 1,5 horas-aula

**CARGA HORÁRIA SEMESTRAL:** 30 horas-aula

### I – EMENTA

Conceito de algoritmo; como se analisa a eficiência de um algoritmo (uso de notação assintótica e relação de recorrência). Técnicas de projeto: força bruta, indução, divisão e conquista, programação dinâmica, método guloso. Algoritmos para cadeias; Algoritmos para matrizes. Análise amortizada de algoritmos. Estruturas avançadas: heaps de Fibonacci. Fluxos em redes: caminhos mínimos, fluxo máximo, fluxo de custo mínimo. Algoritmos de cadeias. Algoritmos aproximados e heurísticos para problemas NP-completos.

### II – OBJETIVOS GERAIS

A disciplina tem por objetivo apresentar a complexidade computacional de algoritmos, isto é, a quantidade de tempo de execução e consumo de espaço necessários para executá-los. Para tanto, serão apresentadas técnicas algorítmicas adequadas à resolução de problemas de diferentes áreas da Computação (grafos, redes, etc.), bem como a análise da eficiência das mesmas. Ainda, estudam-se certos paradigmas computacionais, que se mostram úteis na criação de algoritmos, a saber: divisão e conquista, programação dinâmica, gula, aproximação, etc.

### III – OBJETIVOS ESPECÍFICOS

* Determinar a complexidade de desempenho, no tempo de execução no espaço, dos algoritmos apresentados ao longo do curso.
* Explicar o que é um comportamento melhor, esperado e pior de um algoritmo
* Determinar a definição formal de O grande;
* Usar relações de recorrência para determinar a complexidade no tempo de algoritmos definidos recursivamente.
* Apresentar a propriedade das árvores binárias heaps e seu uso como filas de prioridades;
* Usar, implementar e avaliar desempenho de algoritmos de cadeias;
* Identificar exemplos práticos para as técnicas de programação dinâmica e métodos gulosos
* Apresentar algoritmos de aproximação para problemas NP-completos;
* Introduzir o conceito de algoritmos “multithreaded”
* Apresentar análise amortizada.

**IV – COMPETÊNCIAS**

Compreender a complexidade computacional de algoritmos utilizando técnicas adequadas à resolução de problemas. Compreender alguns paradigmas computacionais para desenvolver algoritmos eficientes.

**V – CONTEÚDO PROGRAMÁTICO**

Módulo1 - Complexidade Computacional e Assintótica; Notação O – Grande; Propriedades da Notação O – Grande; Notações Teta e Ômega. Notação o, Notação ômega minúsculo; O Melhor, o Médio e o Pior Caso;

Módulo 2 – Recorrências e algoritmos recursivos; Uso de notações, somatórios e funções.

Módulo 3 – Heaps: Heaps como filas de prioridades

Módulo 4– Organizando matrizes como heaps; Heap de Fibonacci

Módulo 5 – Algoritmos para cadeias: subsequência comum mais longa, transformando uma cadeia em outra, correspondências de cadeias.

Módulo 6 – Programação Dinâmica: Multiplicação de cadeia de matrizes

Módulo 7 – Algoritmos Gulosos: Códigos de Huffman

Módulo 8 – Redes: Fluxos Máximos

Módulo 9 – Redes: Fluxos Máximos de Custos Mínimos

Módulo 10 – Algoritmos aproximados para problemas NP completos

Módulo 11 – Análise Amortizada: tabelas dinâmicas

Módulo 12 – Algoritmos “Multithreaded”

### VI – ESTRATÉGIA DE TRABALHO

###### As disciplinas são ministradas preferencialmente por meio de aulas expositivas, metodologias ativas e diversificadas apoiadas nos planos de ensino. O desenvolvimento dos conceitos e conteúdos ocorre com apoio de propostas de leituras de livros e artigos científicos básicos e complementares, exercícios, discussões em fórum/chats ou presenciais - quando for o caso, sugestões de filmes, vídeos e demais recursos audiovisuais. Com o objetivo de aprofundar e enriquecer o domínio dos conhecimentos e incentivar a pesquisa, o docente pode propor trabalhos individuais ou em grupo, palestras, atividades complementares e práticas em diferentes cenários, que permitam aos alunos assimilarem os conhecimentos essenciais para sua formação.

###### **VII – AVALIAÇÃO**

###### A média do semestre será calculada de acordo com o Regimento da IES. As avaliações e o critério de aprovação seguem o determinado pela instituição, conforme divulgação feita no manual do aluno.

###### **VIII – BIBLIOGRAFIA**

Básica

Cormen, T. *Desimistificando Algoritmos* Elsevier, 2013.

[https://integrada.minhabiblioteca.com.br/reader/books/9788595153929/epubcfi/6/2[%3Bvnd.vst.idref%3Dhtml-cover-page]!/4/2/4%4051:2](https://integrada.minhabiblioteca.com.br/reader/books/9788595153929/epubcfi/6/2%5b%3Bvnd.vst.idref%3Dhtml-cover-page%5d!/4/2/4%4051:2) .Acesso em: 12.Nov.2022.

Dasgupta, S.; Papadimitrious, C. Varizani, U. *Algoritmos* McGraww Hill 2010. ISBN: 9788563308535

<https://integrada.minhabiblioteca.com.br/reader/books/9788563308535/pageid/0> .Acesso em: 12.Nov.2022.

Toscani, L. V. ; Veloso, P. A. S. *Complexidade de Algoritmos* Vol. 13 Série Livros Didáticos Informática da UFRGS. Bookman, LTC 2012. ISBN: 9788540701380.

<https://integrada.minhabiblioteca.com.br/reader/books/9788540701397/pageid/0> .Acesso em: 12.Nov.2022.

Complementar

Cormen, T. ;Leiserson, C.; Rivest, R. *Algoritmos* 3ª. Ed. Campus, 2012. ISBN: 9788535236996

[https://integrada.minhabiblioteca.com.br/reader/books/9788595158092/epubcfi/6/2[%3Bvnd.vst.idref%3Dcover]!/4/2/2%4051:2](https://integrada.minhabiblioteca.com.br/reader/books/9788595158092/epubcfi/6/2%5b%3Bvnd.vst.idref%3Dcover%5d!/4/2/2%4051:2) .Acesso em:12.Nov.2022.

Kelinberg, J.; Tardos, E. *Algorithm Design* Addison-Wesley, 2005.

Knuth, D. E. The Art of Computer Programming, Addison-Wesley, , 1998 ISBN-10: 0201485419.

Ziviani, N. Projeto de Algoritmos com Implementações em Java e C++. Cengage Learning, 2012. ISBN-10 8522105251.

<https://integrada.minhabiblioteca.com.br/reader/books/9788522108213/pageid/0> .Acesso em:12.Nov.2022.