

## PLANO DE ENSINO

### ANÁLISE DE ALGORITMOS

#### 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 delas. Ainda se estudam 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**

A disciplina é ministrada por meio de aulas expositivas, metodologias ativas e diversificadas apoiadas no plano de ensino. O desenvolvimento dos conceitos e conteúdos ocorre com o apoio de propostas de leituras de livros e artigos científicos básicos e complementares, exercícios, discussões em fórum e/ou *chats*, 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 a sua formação.

## **VII – Avaliação**

A avaliação é um processo desenvolvido durante o período letivo e leva em conta todo o percurso acadêmico do aluno, como segue:

- Acompanhamento de frequência.
- Acompanhamento de nota.
- Desenvolvimento de exercícios e atividades.
- Trabalhos individuais ou em grupo.
- Projeto Integrado Multidisciplinar.
- Estudos Disciplinares.
- Atividades complementares.

A avaliação presencial completa esse processo. Ela é feita no polo de apoio presencial no qual o aluno está matriculado, seguindo o calendário acadêmico. Estimula-se a autoavaliação, por meio da autocorreção dos exercícios, questionários e atividades, de modo que o aluno possa acompanhar sua evolução e rendimento escolar, possibilitando ainda a oportunidade de melhoria contínua por meio de revisão e *feedback*.

## **VIII – Bibliografia**

## Básica

CORMEN, T. *Desmistificando algoritmos*. Rio de Janeiro: Elsevier, 2013.

DASGUPTA, S.; PAPADIMITRIOU, C.; VARIZANI, U.; *Algoritmos*. São Paulo: McGraw-Hill, 2013.

TOSCANI, L. V.; VELOSO, P. A. S. *Complexidade de algoritmos*. Vol. 13. Série Livros Didáticos de Informática da UFRGS. Rio de Janeiro: LTC, 2012.

## Complementar

CORMEN, T.; LEISERSON, C.; RIVEST, R. *Algoritmos*. Rio de Janeiro: Campus, 2012.

KLEINBERG, J.; TARDOS, E. *Algorithm design*. Boston: Addison-Wesley, 2005.

KNUTH, D. E. *The art of computer programming*. Boston: Addison-Wesley, 1998.

ZIVIANI, N. Projeto de algoritmos com implementações em Java e C++. São Paulo: Cengage Learning, 2006.