



Plano de Ensino

1 Código e nome da disciplina

WYF1509 ANÁLISE E PROJETO DE ALGORÍTIMOS

2 Carga horária semestral

3 Carga horária semanal

4 Perfil docente

5 Ementa

Ao final desta disciplina, o aluno estará capacitado a projetar algoritmos eficientes para resolução de problemas; avaliar e classificar algoritmos de acordo com o seu comportamento e desempenho quando em execução; implementar software capaz de classificar informações; identificar as principais características das informações a serem armazenadas em um programa; e analisar a complexidade inerente a um determinado problema para apresentar solução. Para tanto serão abordados conteúdos como estruturas de dados, notação assintótica, algoritmos de ordenação, recorrências, grafos e estudo das classes de problemas. O processo da aprendizagem será desenvolvido mediante aulas expositivas dialogadas, aulas práticas de programação, que poderão ser realizadas tanto em sala quanto no laboratório de informática. A avaliação da aprendizagem será processual, realizada por meio de aplicação de provas, resolução de exercícios individuais, culminando com desenvolvimento de um trabalho, que será apresentado e discutido em sala, e acompanhamento da efetiva participação do aluno nas atividades programadas.

6 Objetivos

1. Projetar algoritmos eficientes para resolução de problemas, utilizando a análise dos elementos necessários para a sua solução. 2. Avaliar e classificar algoritmos de acordo com o seu comportamento e desempenho quando em execução, visando escolher a solução mais adequada para tratar um determinado problema. 3. Implementar software capaz de classificar informações, utilizando os algoritmos clássicos de ordenação de dados. 4. Identificar as principais características das informações a serem armazenadas em um programa, permitindo a escolha da estrutura de dados mais adequada para solução do problema. 5. Analisar a complexidade inerente a um determinado problema, permitindo avaliar a dificuldade da sua resolução.

7 Procedimentos de ensino-aprendizagem

8 Temas de aprendizagem

1. Algoritmos: definição, elementos fundamentais e propriedades. Situações problema que podem ser resolvidos empregando-se algoritmos. Algoritmos e estruturas de dados. Função dos algoritmos na computação. 10. Paradigma da divisão e conquista. Processo de construção de uma solução computacional: método da divisão e conquista. Análise de exemplos de algoritmos que fazem uso da técnica da divisão e conquista. 11. Grafos: definição e propriedades. Representações de grafos. Método da busca em largura. Método da busca em profundidade. Análise de algoritmos que utilizam o método da busca em largura e busca em profundidade. 12. Classes de problemas. Problemas NP-Completo. Tempo polinomial. Verificação de tempo polinomial. Caráter NP-Completo e redutibilidade. Provas do caráter NP-Completo. Exemplos de problemas NP-Completo. 13. Algoritmos de aproximação: fundamentos. Problemas NP-difíceis e algoritmos de aproximação. Processo de construção de uma solução computacional utilizando algoritmos de aproximação para problemas de otimização. 14. Estado da arte da pesquisa e desenvolvimento tecnológico em algoritmos. Estrutura e etapas para elaboração de um projeto de software. Análise de soluções computacionais alternativas para problemas clássicos da computação. 2. Tipos abstratos de dados. Estruturas de dados fundamentais: pilhas; filas; listas encadeadas; deque; e árvores. Tabelas de dispersão (hash). Análise comparativa entre as estruturas de dados clássicas. 3. Principais algoritmos de ordenação de dados: ordenação por troca ("BubbleSort" e "QuickSort"); ordenação por inserção ("InsertionSort"); ordenação por seleção ("SelectionSort" e "HeapSort"); e outros métodos (MergeSort). 4. Fundamentos sobre a análise de algoritmos. Notação assintótica. Ordem O. Ordem ômega. Ordem theta. Análise de algoritmos clássicos de ordenação de dados (BubbleSort, InsertionSort, MergeSort e QuickSort). 5. Estruturas de dados avançadas. Árvores B: propriedades; operações básicas; e eliminação de uma chave. Árvores "Red-Black": propriedades. Utilização de árvores B e de árvores "Red-Black" em computação. 6. Equações de recorrência: definição. Método de substituição. Método de árvore de recursão. Método mestre. Demonstração do teorema mestre. Análise comparativa da adequação e aplicabilidade dos vários métodos. 7. Método da programação dinâmica. Otimização combinatória: fundamentos. Problemas de otimização combinatória com uso de programação dinâmica. Análise de exemplos de algoritmos envolvendo programação dinâmica. 8. Características necessárias para um problema possuir solução utilizando o método da programação dinâmica. Etapas para criação de um algoritmo de programação dinâmica. Projeto de software utilizando programação dinâmica. 9. Método dos algoritmos gulosos. Elementos necessários a uma estratégia gulosa. Algoritmos que fazem uso de métodos gulosos: fundamentos teóricos. Análise de exemplos de algoritmos que utilizam uma solução gulosa.

9 Procedimentos de avaliação

10 Bibliografia básica

11 Bibliografia complementar