

PLANO DE ENSINO

DEPARTAMENTO: Ciência da Computação

DISCIPLINA: Complexidade de Algoritmos

SIGLA: CAL0001

CARGA HORÁRIA TOTAL: 72

TEORIA: 36

PRÁTICA: 36

CURSO(S): Bacharelado em Ciência da Computação

SEMESTRE/ANO: 1/2017

PRÉ-REQUISITOS: TEG0001

OBJETIVO GERAL DA DISCIPLINA: Analisar a complexidade de tempo e espaço de algoritmos. Identificar o melhor caso, o pior caso e o caso médio de execução de algoritmos. Identificar problemas tratáveis e intratáveis.

EMENTA: Estudo de complexidade via métodos de desenvolvimento de algoritmos. Modelos de computação e ferramentas para notação para análise de algoritmos. Algoritmos iterativos e recursivos. Solubilidade de problemas. Intratabilidade de problemas. Análise da complexidade de algoritmos clássicos na área da computação.

OBJETIVOS ESPECÍFICOS:

Capacitar o aluno a analisar a complexidade de tempo e espaço de algoritmos e ser capaz de identificar problemas considerados intratáveis.

C.H.	CONTEÚDOS PROGRAMATICOS	AValiação
14 h/a	Apresentação do planejamento e da ementa da disciplina. Introdução à disciplina. Conceitos Básicos de Complexidade: - Notação O grande - Ordens de complexidade - Análise de complexidade com uma variável - Análise de complexidade de tempo de algoritmos recursivos - Somatórios - Complexidade de Espaço - Notação Assintótica - Teorema Mestre	Prova 1 Exercícios
8 h/a	Análise de Algoritmos de Ordenação: Merge Sort, Quick Sort, Heap Sort, Counting Sort e Bucket Sort	Prova 1 Trabalho
14 h/a	Análise de Complexidade de Estruturas de Dados Elementares: pilha, fila, lista encadeada e árvore Tabelas Hash Análise de Complexidade com múltiplas variáveis	Prova 1 Trabalho
14 h/a	Análise de Complexidade de operações elementares com inteiros de 'n' bits Números primos, aritmética modular e algoritmo de criptografia RSA	Prova 2 Trabalho
10 h/a	Abordagens para Resolução de Problemas: - Indução matemática - Divisão e conquista	Prova 2

	<ul style="list-style-type: none"> - Algoritmos gulosos - Algoritmos de tentativa e erro - Programação dinâmica - Algoritmos de aproximação 	Exercício
12 h/a	Conceitos de Teoria da Computação vinculados com complexidade: <ul style="list-style-type: none"> - Problemas tratáveis e intratáveis; - Classes de problemas: P, NP, NP-Completo e NP-Difícil; - Redução de problemas; - Problemas NP-Completos: SAT, 3-CNF-SAT, Clique, Cobertura de Vértices, Ciclo Hamiltoniano, Caixeiro Viajante, Subset-Sum - Algoritmos pseudo-polinomiais 	Prova 2 Exercícios

METODOLOGIA PROPOSTA:

Aulas expositivas acompanhadas de trabalhos práticos relacionados aos conteúdos apresentados na aula. Aulas práticas em laboratório que objetivam a implementação ou pesquisa dos conceitos apresentados nas aulas teóricas. Listas de exercícios para auxiliar na fixação do conteúdo apresentado. Seminários para proporcionar a busca de forma autônoma pelo conhecimento. Provas teóricas para avaliar o conteúdo conceitual aprendido. Trabalhos para avaliar a capacidade do uso dos conceitos aprendidos. Até 20% do conteúdo programático poderá ser ministrado na forma de ensino a distância.

AValiação:

Do desempenho do aluno:

Os alunos serão avaliados com base no seu desempenho nas provas, trabalhos e listas de exercícios, sendo que ao final do semestre o aluno deverá ter comparecimento mínimo de 75% às aulas e desempenho mínimo de 70% nas avaliações. O grau de desempenho do aluno será avaliado com base no desenvolvimento das seguintes atividades e com os seguintes critérios:

- a) Provas: 40%
 - a.1) Prova 1 (20%)
 - a.2) Prova 2 (20%)
- b) Trabalhos: 45%
 - b.1) Trabalho/Pesquisa algoritmos de ordenação (7%)
 - b.2) Apresentação trabalho complexidade alg. Grafos: (8%)
 - b.3) Trabalho com relatório comparativo entre busca binária, árvore e hash (12%)
 - b.4) Implementação e análise do algoritmo de criptografia RSA (18%)
- c) Exercícios: 15%
 - c.1) Análise de algoritmos simples I (1,5%)
 - c.2) Somatórios e algoritmos recursivos (3%)
 - c.3) Análise de algoritmos II – teorema mestre e complexidade de espaço (3%)
 - c.4) Implementação e análise de algoritmos - guloso, *backtracking* e programação dinâmica (5%)
 - c.5) Exercícios de classes de problemas (2,5%)

BIBLIOGRAFIA (GERAL) OU DE USO DA DISCIPLINA:

Básica:

Algoritmos. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Campus.
 Projeto de Algoritmos com Implementações em Pascal e C. Nívio Ziviani. Cengage Learning.
 Data structures and Algorithms. Data structures and algorithms. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman. Addison Wesley, 1987.

Complementar:

The Status of the P Versus NP Problem. Lance Fortnow. Communications of the ACM, Vol. 52 No. 9, Pages 78-86.
 Algorithms. Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani. McGraw Hill.