



Universidade Federal
de São João del-Rei

COORDENADORIA DO CURSO DE ENGENHARIA MECATRÔNICA

PLANO DE ENSINO

Disciplina: Programação Competitiva I			Período: 10º		Currículo: 2010
Docente Responsável: Alex Vidigal Bastos, Samuel			Unidade Acadêmica: DTECH		
Pré-requisito: Algoritmos e Estrutura de Dados II			Co-requisito: não se aplica		
C.H. Total: 72h	C.H. Prática: 36h	C.H. Teórica: 36h	Grau: Bacharelado	Ano: 2023	Semestre: 2º

EMENTA

Conceitos básicos de aritmética e geometria. Estudo de técnicas de projeto de algoritmos e programação de computadores. Conceitos básicos de matemática discreta, algoritmos em grafos, estruturas de dados e geometria computacional. Estudo e implementação de algoritmos para tratar problemas reais.

OBJETIVOS

Objetivos: Esta disciplina tem como objetivo preparar os alunos para trabalhar com diferentes tipos de algoritmos, com foco em maratonas de programação e a Olimpíada Brasileira de Informática (OBI). Propõe-se introduzir noções da teoria da complexidade computacional, estratégias computacionais, e, paralelamente, apurar as soft skills dos participantes, como a capacidade de trabalhar em equipe, de concentração e de trabalho sob pressão. Para tal, este curso terá um propósito prático, voltado para a resolução ótima de problemas computacionais reais. O foco será na resolução de problemas nos moldes daqueles utilizados em competições de programação, utilizando técnicas como: Programação Dinâmica, Algoritmos Gulosos, Algoritmos em Grafos, Backtracking e Heurísticas e Algoritmos Geométricos. Ademais, esta disciplina busca a integração de disciplinas do núcleo de matemática e programação de computadores, por meio do desenvolvimento de problemas práticos. Para cursar essa disciplina, é extremamente necessário o domínio de técnicas de programação básicas.

CONTEÚDO PROGRAMÁTICO

1. Unidade de nivelamento, revisão e introdução:
 - Noções de complexidade e análise de algoritmos;
 - Classes de problemas computacionais: problemas de decisão, de busca e de otimização;
 - Estratégias computacionais: gulosa, heurística, divisão e conquista, programação dinâmica e força bruta;
 - Algoritmos exatos e aproximados.
2. Conceitos básicos de Aritmética e Geometria:
 - Inteiros, operações e comparações;
 - Propriedades básicas dos inteiros (sinal, paridade, divisibilidade etc.);
 - Frações;
 - Linha, segmento de linha, ângulo, triângulo, retângulo, quadrado, circunferência;
 - Distância Euclidiana;
 - Teorema de Pitágoras;
 - Números primos;
 - Ponto, vetor, coordenadas no plano;
 - Aritmética modular básica: adição, subtração e multiplicação;
 - Polígono (vértice, aresta, convexo, área);
 - Operações com matrizes (adição, multiplicação e exponenciação).
3. Conceitos básicos de Matemática Discreta:
 - Conceitos de grafos e árvores;
 - Árvores e suas propriedades básicas, árvores enraizadas;
 - Grafos direcionados e não direcionados;

- Grau, caminho, ciclo, conectividade;
- Grafos com pesos, cores ou classificações nas arestas ou vértices;
- Operações simples em inteiros de tamanho arbitrário;
- Algoritmos de força bruta e programação dinâmica com auxílio de máscaras de bits;
- Exponenciação de matrizes para resolver problemas de programação dinâmica;
- Quickselect para achar o *k-ésimo* menor elemento.

4. Algoritmos em grafos:

- Percorrer grafos com busca em largura e busca em profundidade;
- Algoritmos de caminho mínimo (Dijkstra, Bellman-Ford, Floyd-Warshall);
- Encontrar componentes conexas;
- Ordenação topológica;
- Árvores geradoras mínimas;
- Encontrar um caminho/ciclo de Euler;
- Conjunto de arestas independentes em grafo bipartido (bipartite matching) em $O(VE)$.

5. Estruturas de dados:

- Não é necessário, mas é muito útil conhecer a STL usando C++;
- Pilhas e filas;
- Listas ligadas;
- Representação de grafos;
- Árvore de busca binária estática;
- Heap binário;
- Conjuntos disjuntos: Union-find;
- Árvore de Fenwick (binary indexed tree) 1D;
- Menor ancestral comum: algoritmo para responder perguntas em $O(\log(N))$;
- Árvore de Fenwick (binary indexed tree) 2D;
- Árvore de segmentos (Segment tree);
- Estruturas de dados persistentes;
- Divisão em buckets de tamanho \sqrt{N} (square root decomposition);
- Árvores de busca binária balanceadas (Treaps, splay trees, etc);
- Árvore de segmentos 2D;
- Tries.

6. Geometria computacional:

- Pontos, vetores, linhas e segmentos de linhas;
- Pontos colineares, vetores paralelos e ortogonais;
- Interseção de duas linhas;
- Compressão de coordenadas;
- Envoltória convexa (convex hull) em $O(N\log(N))$;
- Line sweep;
- Calcular área de um polígono;
- Checar se um polígono contém um ponto.

METODOLOGIA DE ENSINO

- Aulas expositivas com diálogos e atividades iterativas;
- Aprendizagem através de solução de problemas, de forma que o aluno aprenda com a prática;
- Desenvolvimento de exercícios teóricos, ou com o uso de uma linguagem de programação em laboratório, assim que a matéria é lecionada;
- Para cada item do conteúdo programático será dada uma atividade;
- Exercícios extraclasse, provas e trabalhos práticos individuais e em grupos, para aprendizado aprofundado dos conceitos e técnicas estudadas.

- Desenvolvimento de trabalho prático, preferencialmente relativo à área de interesse do curso.	
CONTROLE DE FREQUÊNCIA E CRITÉRIOS DE AVALIAÇÃO	
<p>Dez pontos distribuídos ao longo do semestre da seguinte maneira:</p> <ul style="list-style-type: none"> ○ 3 avaliações práticas, cada uma valendo 2.5 pontos (total 7.5 pontos) ○ <i>n</i> exercícios práticos e/ou teóricos, totalizando 2.5 pontos <p>Prova Substitutiva: o aluno que ficar abaixo da média de 60% ao final do curso, ou vier a perder alguma aplicação de prova, poderá submeter-se à uma prova de substituição no valor de 2.5 pontos, substituindo a nota de uma das avaliações teóricas ou práticas. Essa atividade substitutiva abordará todo o conteúdo da disciplina. Ao final do semestre, o aluno que não atingir 6 pontos totais, não será aprovado.</p> <p>Será feito o controle de presença em todas as aulas. Por tratar-se de um curso presencial, o comparecimento do corpo discente às aulas é obrigatório. Em nenhuma hipótese será concedido abono de falta, exceto nos casos previstos na legislação em vigor e no estatuto da universidade. O discente que não comparecer a 75% das aulas será reprovado por infrequência.</p>	
BIBLIOGRAFIA BÁSICA	
<ol style="list-style-type: none"> 1. ZIVIANI, <u>Nivio</u>. Projeto de Algoritmos Com Implementações em Pascal e C, Pioneira, 2ª Edição, 2004. 2. CORMEN, T. H. , LEISERSON C.E., RIVEST R.L, STEIN C. Algoritmos - Teoria e Prática, Elsevier, 2a edição, 2002. 	
BIBLIOGRAFIA COMPLEMENTAR	
<ol style="list-style-type: none"> 1. SOUZA, Marco, et al., Algoritmos e Lógica de Programação, 2005. 2. FORBELLONE, A. L. V.; EBERSPACHER, H. F. Lógica de Programação. São Paulo, Makron Books, 2000. 3. EVARISTO, Jaime. Aprendendo a programar: Programando em Linguagem C. Rio de Janeiro: BookExpress, 2001. 4. Kernighan, B.W., Ritchie, D.M. C: a linguagem de programação (Padrão ANSI). 16ª Ed. Rio de Janeiro: Campus, 3ª Edição, 2003. 5. LOPES, Anita; GARCIA, Guto. Introdução à programação: 500 algoritmos resolvidos. Rio de Janeiro: Elsevier, 2002. 469 p. il. 5ªtiragem. ISBN 85-352-1019-9. 	
<hr/> Docente Responsável	Aprovado pelo Colegiado em / / <hr/> Coordenador do Curso de Engenharia Mecatrônica



Emitido em 11/08/2023

PLANO DE ENSINO Nº PE PC I 2023/2/2023 - CEMEC (12.56)

(Nº do Documento: 2967)

(Nº do Protocolo: 23122.031258/2023-71)

(Assinado digitalmente em 14/08/2023 11:02)

ALEX VIDIGAL BASTOS

PROFESSOR DO MAGISTERIO SUPERIOR

DTECH (12.27)

Matrícula: ###921#4

(Assinado digitalmente em 14/08/2023 15:31)

EDGAR CAMPOS FURTADO

COORDENADOR DE CURSO

CEMEC (12.56)

Matrícula: ###424#4

(Assinado digitalmente em 11/08/2023 16:22)

SAMUEL MOREIRA ABREU ARAUJO

PROFESSOR DO MAGISTERIO SUPERIOR

DTECH (12.27)

Matrícula: ###282#8

Visualize o documento original em <https://sipac.ufsj.edu.br/public/documentos/> informando seu número: **2967**, ano: **2023**, tipo: **PLANO DE ENSINO**, data de emissão: **11/08/2023** e o código de verificação: **45dc6d1635**