



PLANO DE ENSINO

CURSO	195 - Engenharia de Computação	MATRIZ	535
--------------	---------------------------------------	---------------	------------

FUNDAMENTAÇÃO LEGAL	Resoluções: N° 89/08-COEPP - N° 153/09-COEPP - N° 158/10-COEPP
----------------------------	--

DISCIPLINA/UNIDADE CURRICULAR	CÓDIGO	PERÍODO	CARGA HORÁRIA (aulas)					
			AT	AP	APS	AD	APCC	Total
Algoritmos e Estruturas de Dados 1	AE22CP	2º	72	30	06	00	00	108

AT: Atividades Teóricas, AP: Atividades Práticas, APS: Atividades Práticas Supervisionadas, AD: Atividades a Distância, APCC: Atividades Práticas como Componente Curricular.

PRÉ-REQUISITO	Fundamentos de Programação
EQUIVALÊNCIA	

OBJETIVOS

Apresentar ao aluno as bases de algoritmos e estruturas de dados e as técnicas de resolução de problemas. Ao final da disciplina o aluno será capaz de em face de um problema, escolher a melhor estrutura de dados e algoritmo para solucioná-lo em relação a tempo de desenvolvimento, complexidade computacional e espacial.

EMENTA

Definição de tipos abstratos de dados (TAD), implementação de tipos de dados: vetores, cadeias de caracteres, conjuntos e registros. Programação estruturada. Recursividade: aplicações, considerações sobre desempenho e uso de memória. Exemplos clássicos. Listas estáticas: simplesmente encadeadas, duplamente encadeadas e circulares. Alocação dinâmica de memória (uso de ponteiros tipados e genéricos). Listas dinâmicas: simplesmente encadeadas, duplamente encadeadas e circulares. Pilhas e Filas (estáticas e dinâmicas). Aplicações sobre pilhas: notação reversa polonesa. Listas, pilhas e filas generalizadas (utilizando TAD). Construção de bibliotecas para estruturas de dados. Algoritmos de ordenação e pesquisa. Matrizes esparsas: conceito, implementação com listas ligadas. Tabela HASH. Organização de arquivos. Seqüencial, direto, conceito de indexação de arquivos.

CONTEÚDO PROGRAMÁTICO

ITEM	EMENTA	CONTEÚDO
1	Definição de tipos abstratos de dados (TAD), implementação de tipos de dados: vetores, cadeias de caracteres, conjuntos e registros.	Tipos abstratos de dados e sua representação em C. Vantagem em comparação com tipos simples. Operações possíveis.
2	Programação estruturada. Recursividade: aplicações, considerações sobre desempenho e uso de memória. Exemplos clássicos.	Organização de programa na forma de funções simples, reuso de código, funções recursivas, vantagens da recursividade e suas implicações no tempo de processamento e no uso de memória RAM. Noções de passagem de parâmetros em C, empilhamento de parâmetros, endereço de retorno e informações relevantes para recursividade. Exemplos de algoritmos iterativos e recursivos.
3	Listas estáticas: simplesmente encadeadas, duplamente encadeadas e circulares.	Organização de informação sob a forma de uma lista, como acessá-la, localização do primeiro elemento, elemento seguinte, último. Inserções de elementos na lista, eliminação de elementos, listas simples, duplas e circulares. Trabalho sobre listas
4	Alocação dinâmica de memória (uso de ponteiros tipados e genéricos). Listas dinâmicas: simplesmente encadeadas,	Recapitulação de listas estáticas, problemas das listas estáticas. Alocação dinâmica de memória, diagrama de memória de um computador,

	duplamente encadeadas e circulares. Pilhas e Filas (estáticas e dinâmicas).	ponteiros para tipos simples de dados, ponteiros genéricos, liberação de memória, implementação de listas simples, duplas, circulares, pilhas, filas, exemplos e exercícios.
5	Aplicações sobre pilhas: notação reversa polonesa. Listas, pilhas e filas generalizadas (utilizando TAD).	Criação de conjunto de funções para realizar operações sob pilhas, listas e filas genéricas usando tipos abstratos de dados. Exemplos de problemas que utilizam estas estruturas de dados. Notação reversa polonesa e sua conversão para notação normal, interpretação de expressões.
6	Construção de bibliotecas para estruturas de dados.	Criação de bibliotecas em C utilizando as funções implementadas para trabalhar sobre as estruturas de dados listas, filas e pilhas (genéricas). Criação de arquivos objetos, linkagem estática, makefile, programação em C com vários módulos e possivelmente vários desenvolvedores.
7	Algoritmos de ordenação e pesquisa.	Algoritmos clássicos sobre ordenação e pesquisa de elementos simples (inteiros, palavras, números reais) e usando tipos abstratos de dados. Análise elementar sobre os algoritmos estudados em relação a sua complexidade de implementação e complexidade computacional. Comparação de desempenho utilizando simulação.
8	Matrizes esparsas: conceito, implementação com listas ligadas.	Definição de matrizes esparsas, localização de elementos e inserção de elementos. Implementação em C usando listas ligadas.
9	Tabela HASH.	Definição de uma tabela hash, função de hashing e tratamento de colisões. Questões sobre tempo de pesquisa e inserção no melhor caso/pior caso.
10	Organização de arquivos. Sequencial, direto, conceito de indexação de arquivos.	Funcionamento de um sistema de gerenciamento de arquivos. Implementação de arquivos das formas sequencial, direto. Questões de tempo de acesso e requisitos de hardware necessários. Conceito de indexação, chave, pesquisa e eliminação de elementos de um arquivo.

PROCEDIMENTOS DE ENSINO

AULAS TEÓRICAS

Aulas ministradas em sala de aula, nas quais a ênfase está em explicações conceituais.

AULAS PRÁTICAS

Aulas centradas na realização de atividades práticas pelos alunos com supervisão, orientação e auxílio do professor; aulas em que o professor realiza a resolução tutorada de exercícios (o professor conduz a resolução que é acompanhada pelos alunos); aulas em que o professor exemplifica a resolução de exercícios. As aulas práticas incluem aulas de laboratório que são realizadas em ambientes específicos em que há uso de equipamentos e materiais que permitem a experimentação.

ATIVIDADES PRÁTICAS SUPERVISIONADAS

Atividades acadêmicas desenvolvidas sob a orientação, supervisão e avaliação de docentes e realizadas pelos discentes em horários diferentes daqueles destinados às atividades presenciais (aulas teóricas e aulas práticas). Estas atividades incluem: estudos dirigidos, trabalhos individuais, trabalhos em grupo, desenvolvimento de projetos, atividades em laboratório, atividades de campo, oficinas, pesquisas, estudos de casos, seminários, desenvolvimento de trabalhos acadêmicos, dentre outras. Deverá ser dada ênfase à realização de atividades em grupo que envolva pesquisa e seja interdisciplinar.

PROCEDIMENTOS DE AVALIAÇÃO

Considerar-se-á aprovado na disciplina, o aluno que tiver frequência igual ou superior a 75% (setenta e cinco por cento) e Nota Final igual ou superior a 6,0 (seis), consideradas todas as avaliações previstas no início do semestre.

No caso do aluno perder alguma avaliação presencial e escrita, por motivo de doença ou força maior, poderá requerer uma única segunda chamada por avaliação, no período letivo. O requerimento deve ser protocolado no Departamento de Registros Acadêmicos dentro do prazo estabelecido pelo regulamento da UTFPR, a prova será aplicada após o deferimento. Para a prova de segunda chamada o professor definirá os conteúdos e a data da avaliação.

REFERÊNCIAS

Referências Básicas:

- SZWARCFITER, Jayme Luiz, **Estruturas de dados e seus algoritmos**, 3. ed., 2010.
- SEBESTA, Robert W. **Conceitos de linguagens de programação**. 4 ed. Porto Alegre: Bokman, 2000.
- SEBESTA, Robert W. **Conceitos de linguagens de programação**. 5 ed. Porto Alegre: Bokman, 2003.
- MIZRAHI, Victorine Viviane. **Treinamento em Linguagem C: Curso Completo - Módulo 2**. São Paulo, Makron: 2001.

Referências Complementares:

- VELOSO, Paulo A. S. **Estruturas de Dados**. Rio de Janeiro, Campus: 1985.
- VILLAS, Marcos Vianna. **Estrutura de Dados: Conceitos e Técnicas de Implementação**. Rio de Janeiro, Campus: 1993.
- PEREIRA, Silvio do Lago. **Estrutura de Dados Fundamentais: Conceitos e Aplicações**. 12. ed. São Paulo, Érica: 2008.
- SILVA, Osmar Quirino. **Estrutura de Dados e Algoritmos Usando C: Fundamentos e Aplicações**. Rio de Janeiro, Ciência Moderna: 2007.
- LAUREANO, Marcos. **Estrutura de Dados com Algoritmos e C**. Rio de Janeiro, Brasport: 2008.
- MIZRAHI, Victorine Viviane. **Treinamento em linguagem C: curso completo - módulo 1**. São Paulo: McGraw-Hill, 2001.
- IBPI. **Estruturas de dados com Pascal**. Rio de Janeiro, IBPI Press: 1993.

ORIENTAÇÕES GERAIS

As datas das avaliações, exceto as de segunda chamada, serão estabelecidas em sala de aula no início do semestre.

O uso de aparelhos celulares deve ser feito somente fora de sala de aula. A utilização de notebook apenas em caso de necessidade em atividades da disciplina.

Professor da disciplina

Coordenador do Curso