

Campus: São José dos Campos		
Curso (s): Engenharia de Computação e Ciência da Computação		
Unidade Curricular (UC): Algoritmos e Estrutura de Dados II		
Unidade Curricular (UC): <i>Algorithms and Data Structure II</i>		
Código da UC: 2833		
Docente Responsável: Lilian Berton		Contato (e-mail): lberton@unifesp.br
Docente (s) Colaborador/a (es/as):		Contato (e-mail):
Ano letivo: 2022	Termo: terceiro	Turma (s): Integral
Nome do Grupo/Módulo/Eixo da UC (se houver):		Idioma predominante em que a UC será oferecida: <input checked="" type="checkbox"/> Português <input type="checkbox"/> English <input type="checkbox"/> Español <input type="checkbox"/> Français <input type="checkbox"/> Libras <input type="checkbox"/> Outro:
UC: <input checked="" type="checkbox"/> Fixa <input type="checkbox"/> Eletiva <input type="checkbox"/> Optativa	Oferecida como: <input checked="" type="checkbox"/> Disciplina <input type="checkbox"/> Módulo <input type="checkbox"/> Estágio <input type="checkbox"/> Outro:	Oferta da UC: <input checked="" type="checkbox"/> Semestral <input type="checkbox"/> Anual
Ambiente Virtual de Aprendizagem: <input type="checkbox"/> Moodle <input checked="" type="checkbox"/> Classroom <input type="checkbox"/> Outro: <input type="checkbox"/> Não se aplica		
Pré-Requisito (s) - Indicar Código e Nome (s) da (s) UC: 2832 - Algoritmos e Estruturas de Dados I		
Carga horária total (em horas): 72h		
Carga horária teórica (em horas): 46	Carga horária prática (em horas): 26	Carga horária de extensão (em horas, se houver): 0h
Se houver atividades de extensão, indicar código e nome do projeto ou programa vinculado na Pró-Reitoria de Extensão e Cultura (ProEC):		
Ementa: Métodos de ordenação interna: quadrático, $n \log n$, linear e outros. Métodos de pesquisa interna: sequencial, busca binária, árvores de pesquisa, balanceamento de árvores. Algoritmos em grafos: busca em largura, profundidade e menor caminho. Tabelas de espalhamento: Hash. Memória externa: modelos, ordenação e pesquisa.		
Conteúdo programático: Análise de complexidade, Métodos de ordenação quadráticos, Métodos de ordenação $n \log n$, Métodos de ordenação linear, Métodos de ordenação externos, Métodos de pesquisa, Árvores AVL, Árvores vermelho-preto, Árvores B, Tabelas Hash, Grafos busca em largura e profundidade, Grafos caminho mínimo.		
Objetivos: <u> Gerais:</u> Introduzir algoritmos e estruturas de dados avançadas. Ao final do curso o aluno deve estar familiarizado com os algoritmos e estruturas de dados avançadas, devendo ser capaz de definir qual algoritmo e/ou estrutura de dados é mais apropriada para cada tipo de problema. <u> Objetivos específicos:</u> Implementar algoritmos e estruturas de dados sofisticadas, para aplicações em memória primária e secundária em alguma linguagem de programação tais como: C, C++ ou Java.		

Metodologia de ensino:

Para que os objetivos dessa disciplina possam ser atendidos e, conseqüentemente contribua com os objetivos do curso, as seguintes estratégias de ensino-aprendizagem serão utilizadas:

Aula teórica e prática para apresentação do conteúdo e acompanhamento das dúvidas dos alunos.

Laboratório de programação para prática do conteúdo.

Aprendizagem baseada em problemas (cada aluno deve propor soluções usando técnicas vistas na UC para problemas apresentados pelo professor).

Prática de exercícios aplicando os conteúdos trabalhados.

Desenvolvimento de pesquisas extraclases sobre os assuntos abordados em aula.

Avaliação:

(E) Lista de exercícios teóricos e judge online semanais com feedback e correções (0 <= Nota <= 10)

(T) 3 trabalhos práticos (aplicação de técnicas de AED2 em problemas reais, os resultados devem ser apresentados por meio de relatório e vídeo) (0 <= Nota <= 10)

(P) 3 provas (0 <= Nota <= 10)

Se média das provas (MP) e média dos trabalhos (MT) forem maior que 5.0:

- Nota Final = $(0,6*MP + 0,3*MT) + 0,1*E$

senão

- Nota Final = $\min (MP, MT) + 0,1*E$

Se Nota final >= 6 será aprovado

Se Nota final < 6 será reprovado

Frequência:

Caso a frequência for menor que 75% o aluno será reprovado por frequência.

Bibliografia:

Básica:

1. CORMEN, Thomas H; SOUZA, Vanderberg D; STEIN, Clifford; RIVEST, Ronald L; LEISERSON, Charles E. Algoritmos: teoria e prática. Rio de Janeiro: Campus, 2002. 916 p. ISBN 9788535209266.

2. ZIVIANI, Nivio. Projeto de algoritmos: com implementações em PASCAL e C. 3 ed. rev. e ampl. São Paulo: Heinle Cengage Learning, 2011. 639 p. ISBN 9788522110506.

3. KNUTH, Donald E. The art of computer programming vol. 1: fundamental algorithms. 3 ed. Upper Saddle River, NJ: Addison-Wesley, 1997. 650 p ISBN 9780201896831.

Complementar:

1. ZIVIANI, Nivio. Projeto de algoritmos : com implementações em JAVA e C++. São Paulo Cengage Learning 2012 1 recurso online ISBN 9788522108213.

2. SEDGEWICK, Robert. Algorithms in C: part 5 - graph algorithms. New Jersey: Addison-Wesley, 2007. 482 p. ISBN 9780201316636.

3. SKIENA, Steven S. The algorithm design manual. 2nd.ed. New York: Springer, c2008. 730 p. ISBN 9781848000698.

4. TENENBAUM, Aaron M; MAYER, Roberto C; SOUZA, Teresa C.f; AUGENSTEIN, Moshe J; LANGSAM, Yedidyah. Estruturas de dados usando C. São Paulo: Pearson, 2008. 884 p. ISBN 9788534603485.

5. GOODRICH, Michael T. Estruturas de dados & algoritmos em Java. 5. Porto Alegre Bookman 2013 1 recurso online ISBN 9788582600191.

6. SHEN, Alexander. Algorithms and programming: problems and solutions. 2nd. ed. New York: Springer, 2010. 272 p. (Springer Undergraduate Texts in Mathematics and Technology). ISBN 9781441917478.