

Campus: São José dos Campos		
Curso (s): Engenharia de Computação e Ciência da Computação		
Unidade Curricular (UC): Projeto e Análise de Algoritmos		
Unidade Curricular (UC): <i>Design and Analysis of Algorithms</i>		
Código da UC: 3579		
Docente Responsável: Sanderson L. Gonzaga de Oliveira		Contato (e-mail): [opcional]
Ano letivo: 2024	Termo: 4º	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 checked="" 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: Matemática Discreta (código 2201) e Algoritmos e Estruturas de Dados II (código 2833).		
Carga horária total (em horas): 72		
Carga horária teórica (em horas): 52	Carga horária prática (em horas): 20	Carga horária de extensão (em horas, se houver): 0
Se houver atividades de extensão, indicar código e nome do projeto ou programa vinculado na Pró-Reitoria de Extensão e Cultura (ProEC): não se aplica		
Ementa: <i>Análise assintótica. Relações de recorrência. Técnicas de prova de corretude de algoritmos. Construção de algoritmos por indução. Análise de Algoritmos: gulosos, ordenação e pesquisa. Programação dinâmica. Redutibilidade de problemas. Introdução à NP-Compleitude.</i>		
Conteúdo programático: <i>Análise de assintótica: Comportamento Assintótico; Notação Assintótica; Notações Padrões e funções comuns. Relações de recorrência: Definições Recorrentes; Resolvendo relações de recorrência; Análise de Algoritmos usando relação de recorrência. Técnicas de prova de corretude: Demonstração de Correção; Técnicas de demonstração: construção, contradição e indução; Algoritmo de Euclides. Construção de algoritmos por indução: Exemplos de problemas cujos algoritmos são construídos por indução: subgrafo induzido maximal, designação; Algoritmo de divisão e conquista; Programação dinâmica: o problema da mochila. Análise de Algoritmos: Estudo da complexidade de algoritmos gulosos, de divisão e conquista e de programação dinâmica; Análise de complexidade de métodos de ordenação: Quicksort, árvore de busca binária ótima. Redutibilidade de problemas: Exemplos de Redução; Reduções envolvendo programação linear. NP-completude: Introdução; Reduções em tempo polinomial; Teorema de Cook e Exemplos de provas NP-completude.</i>		
Objetivos:		
<p><b>Gerais:</b> Esta unidade curricular (UC) está relacionada à área básica de ensino de algoritmos. Os alunos e as alunas aperfeiçoarão e consolidarão seus conhecimentos relacionados a algoritmos, aprendidos em disciplinas anteriores. Nesta unidade curricular, os alunos e as alunas aprenderão a analisar algoritmos, principalmente em relação a custo tempo de execução. Também aprenderão a resolver recorrências, as técnicas básicas para construção de algoritmos e introdução à teoria sobre NP-Compleitude. Com a assimilação dos conteúdos apresentados nesta unidade curricular, o(a)s estudantes terão bases teóricas para se aperfeiçoarem em diversas áreas da Computação relacionadas a problemas computacionais complexos, como Otimização Combinatória, processos estocásticos, programação linear, inteligência artificial, aprendizagem de máquina etc.</p> <p><b>Específicos:</b> Ao final desta unidade curricular é esperado dos alunos e das alunas um entendimento razoável sobre: análise de algoritmos, resolução de recorrências, técnicas básicas para construção de algoritmos e NP-Compleitude.</p>		

Metodologia de ensino: aulas expositivas, aulas de resolução de exercícios, aulas para estudantes tirarem dúvidas. Atividades monitoradas. Atividades complementares à distância.
Avaliação: Avaliação dividada em quatro provas, com peso de 25% para cada uma das 1ª, 2ª e 3ª provas e 20% para a 4ª prova, e um trabalho, com peso de 5%. O(a) estudante será aprovado ou reprovado de acordo com os critérios definidos na Seção IV

(da Avaliação Acadêmica) do Capítulo IV (do Currículo) do Título II (da Graduação) do Regimento Interno da Pró-Reitoria de Graduação da Universidade Federal de São Paulo. A frequência é definida na Seção II (da Frequência) do Capítulo IV (do Currículo) do Título II (da Graduação) do Regimento Interno da Pró-Reitoria de Graduação da Universidade Federal de São Paulo. Especificamente nesta UC, poderá haver chamada em cada uma das duas aulas de cada encontro. Poderá ser atribuída ausência ao(à) discente que se ausentar por mais de 15 minutos em cada uma das duas aulas de cada encontro.

**Bibliografia:**

Básica:

1. CORMEN, Thomas H et al. *Algoritmos: teoria e prática*. Rio de Janeiro: Campus, 2002. 916 p. ISBN 978-85-352-0926-6. tradução de "Introduction to algorithms" 2.ed.
2. VELOSO, Paulo; TOSCANI, Laira Vieira. *Complexidade de algoritmos*. 2.ed. Porto Alegre: Bookman, 2008. 261 p. ISBN 978-85-7780-350-7.
3. MANBER, Udi. *Introduction to algorithms: a creative approach*. Reading, Massachussets: Addison-Wesley, 1989. 478 p p. ISBN 978-0-201-12037-0.
4. Gersting, Judith L; Iorio, Valéria de M. *Fundamentos matemáticos para a ciência da computação: um tratamento moderno de matemática discreta*. 5.ed. Rio de Janeiro: LTC, 2008. 597 p. ISBN 978-85-216-1422-7.

Complementar:

1. Garey, Michael R; Johnson, David S. *Computers and intractability: a guide to the theory of NP-Completeness*. New York: W.H.Freeman and Company, 1979. 338 p. ISBN 978-0-7167-1045-5.
2. ZIVIANI, N. *Projeto de algoritmos: com implementações em PASCAL e C*. 2 ed. rev. e ampl. São Paulo: Thomson, 2004. 552 p. ISBN 978-85-221-0390-4.
3. ZIVIANI, Nivio; BOTELHO, Fabiano C. *Projeto de algoritmos: com implementações em JAVA e C++*. São Paulo: Thomson, 2007. 621 p. ISBN 978-85-221-0525-0.
4. Lewis, Harry R; Papadimitriou, Christos H. *Elementos de teoria da computação*. 2 ed. Porto Alegre: Bookman, 2004. 344 p. ISBN 978-85-7307-534-2.
5. Sipser, Michael. *Introdução à teoria da computação*. [Introduction to the theory of computation]. Tradução: Ruy J. G. B. Queiroz. : Cengage, 2012. 459 p. ISBN 9788522104994.