

Campus: São José dos Campos		
Curso (s): Ciência da Computação, Engenharia de Computação, Matemática Computacional		
Unidade Curricular (UC): Projeto e Análise de Algoritmos		
Unidade Curricular (UC): [nome da UC em inglês] Design and Analysis of Algorithms		
Unidade Curricular (UC): [nome da UC em espanhol - opcional]		
Código da UC: 3579		
Docente Responsável/Departamento: Reginaldo Massanobu Kuroshu / Departamento de Ciência e Tecnologia		Contato (e-mail): [opcional] rmkuroshu@unifesp.br
Docente (s) Colaborador/a (es/as)/Departamento (s):		Contato (e-mail): [opcional]
Ano letivo: 2022	Termo: 4º	Turno/Turma: Integral/Noturno
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: 2833 -Algoritmos e Estruturas de Dados II e 2201 - Matemática Discreta		
Carga horária total (em horas): 72 hrs		
Carga horária teórica (em horas): 52 hrs	Carga horária prática (em horas): 20 hrs	Carga horária de extensão (em horas, se houver):
Se houver atividades de extensão, indicar código e nome do projeto ou programa vinculado na Pró-Reitoria de Extensão e Cultura (ProEC):		
Ementa: 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		
Conteúdo programático: Introdução à análise de algoritmos, Notação assintótica: definições e aplicação do conceito para análise da complexidade de algoritmos. Corretude de algoritmos: invariante de laço, indução. Divisão e conquista: conceitos e exemplos de aplicação. Relações de recorrência: árvore de recorrência, método da substituição, teorema mestre. Projeto e análise de algoritmos de ordenação: heaps, heapsort, quicksort. Projeto por indução: exemplos de aplicação. Backtracking: exemplos de aplicação. Programação dinâmica: conceitos, propriedades e exemplos de aplicação. Algoritmos gulosos: conceitos, propriedades e exemplos de aplicação. Redução: conceitos e exemplos. NP-completude: conceitos e demonstração de NP-completude.		
Objetivos: Geral: Nesta unidade curricular o aluno aprenderá técnicas de construção de algoritmos, prova de corretude, cálculo de complexidade. Também entrará em contato com classes de problemas intratáveis na exatidão. Específicos: Ao final do curso é esperado que os alunos projetem algoritmos com um maior formalismo e utilizando de técnicas que otimizem a complexidade para o seu melhor desempenho. Além disso, o aluno deverá reconhecer problemas de fácil e de difícil solução por meio do estudo de sua complexidade.		
Metodologia de ensino: Aulas expositivas; Laboratório de programação; Listas de exercícios; Atividades complementares a distância.		

Avaliação:

- Provas (P)

- Avaliação bimestral parcial 1 (P1);

- Avaliação bimestral parcial 2 (P2);

- $P = (P1 + P2)/2$

- Atividades e exercícios práticos de programação, os quais poderão ser realizados individualmente ou em grupos (T);

- A nota final será dada por: $\text{Nota final} = 0,7xP + 0,3xT$

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 NPCompleteness. 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

Básica:

[igual consta no Projeto Pedagógico do Curso]

Catálogo de UCs vigentes dos PPCs: <https://www.unifesp.br/campus/sjc/catalogo-de-disciplinas/ucs-vigentes.html>

Complementar:

Cronograma: [opcional]