



UNIVERSIDADE FEDERAL DE OURO PRETO
PRÓ-REITORIA DE GRADUAÇÃO
PLANO DE ENSINO



Nome do Componente Curricular em português: Projeto e Análise de Algoritmos		Código:
Nome do Componente Curricular em inglês: Design and Analysis of Algorithms		BCC241
Nome e sigla do departamento: Departamento de Computação (DECOM)		Unidade acadêmica: ICEB
Nome do docente: Anderson Almeida Ferreira		
Carga horária semestral 60 horas	Carga horária semanal teórica 8 horas/aula	Carga horária semanal prática 0 horas/aula
Data de aprovação na assembleia departamental: / /		
Ementa: Medidas de complexidade, análise assintótica de complexidade e notação Big O, Little o, Omega e Theta; análise de algoritmos iterativos e recursivos; medidas empíricas de performance; estratégias de projeto de algoritmos: divisão e conquista, método guloso, programação dinâmica, backtracking, branch and bound, probabilístico, aproximado; classes de complexidade: P, NP, NP-Completo e NP-Difícil.		
Conteúdo programático: <ul style="list-style-type: none">• Medidas de complexidade, análise assintótica de complexidade e notação Big O, Little o, Omega e Theta.<ul style="list-style-type: none">• Panorama e Conceitos Básicos.• Medidas de Complexidade (tempo, espaço)• Análise Assintótica• Análise de Algoritmos Iterativos e Recursivos<ul style="list-style-type: none">• Teorema Mestre• Medidas Empíricas de Performance• Estratégias de projeto de algoritmos<ul style="list-style-type: none">• Divisão e conquista: MergeSort, Medianas, QuickSort e Exponencial• Método guloso: Conceitos, Árvores Geradoras Mínimas - Prim & Kruskal, Código de Huffman, Cláusula de Horn, Problema da Mochila e Seleção de atividades• Programação dinâmica: Conceitos, Maior Sequência Crescente, Distância de Edição, Problema da Mochila e Multiplicação de Cadeia de Matrizes• Backtracking• Branch and bound• Probabilístico• Aproximado• Classes de complexidade<ul style="list-style-type: none">• Problemas de Busca - Decisão e Otimização,		

- Classe P
- Classe NP
- Classe NP-Completo
- NP-Difícil
- Redução de problemas

Objetivos:

Ao final do curso espera-se que os alunos possuam os seguintes conhecimentos e habilidades: Capacidade para fazer análise de pior e melhor caso do tempo de execução de algoritmos típicos, como algoritmos de ordenação e pesquisa, algoritmos clássicos sobre grafos, etc; Compreensão sobre o significado da notação usada para expressar ordem de complexidade de algoritmos; Compreensão do princípio básico de cada uma das técnicas de programação estudadas e dos casos aos quais elas se aplicam; Habilidade para implementação dos algoritmos estudados e para aplicação dos mesmos na solução de problemas práticos.

Metodologia:

Aulas expositivas sobre o conteúdo programático: uso de ferramentas síncronas (webconferências) e assíncronas (com recursos de vídeos, textos e áudios).

Estudos Dirigidos: atividades individuais práticas contendo exercícios e implementações dos métodos estudados. Serão avaliados assincronamente.

Trabalhos práticos: Implementação e análise de problemas implementados usando técnicas de programação. Serão avaliados assincronamente.

Leituras recomendadas: leitura de textos técnicos com a finalidade de proporcionar ao discente a oportunidade de consulta e desenvolvimento de sua capacidade de análise, síntese e crítica de uma bibliografia específica. Serão utilizadas ferramentas assíncronas.

Frequência: será computada a frequência mediante a entrega de atividades.

Provas: serão online e no horário regular da disciplina. Os alunos receberão uma prova com questões referentes aos conteúdos estudados e terão um tempo determinado para enviar a folha de respostas.

Atividades avaliativas:

- 04 estudos dirigidos – Valor: 10,0 pontos – Peso: 0,5 cada
- 02 provas teóricas:
 - Prova I – Data: 17/09/20 – Valor: 10,0 pontos – Peso: 2
 - Prova II – Data: 15/10/20 – Valor: 10,0 pontos – Peso: 2
- 02 trabalhos práticos:
 - Trabalho Prático I – Data: 16/09/20 – Valor: 10,0 pontos – Peso: 2
 - Trabalho Prático II – Data: 14/10/20 – Valor: 10,0 pontos – Peso: 2

• Exame Especial: Os alunos que tiverem pelo menos 75% de frequência (mínimo para aprovação) e média inferior a seis poderão fazer o Exame Especial. O Exame Especial será uma prova única, oral e individual, contendo toda a matéria estudada ao longo do semestre. Será agendado um horário para cada aluno.

Cronograma:

Data	Aulas	Tópicos
24/ago	1-2	Apresentação do Curso, Panorama e Conceitos Básicos.
25/ago	3-4	Medidas de Complexidade (tempo, espaço). Análise Assintótica.

26/ago	5-6	Medidas de Complexidade (tempo, espaço). Análise Assintótica.
27/ago	7-8	Algoritmos Probabilísticos.
31/ago	9-10	Algoritmos Probabilísticos.
01/set	11-12	Estudo dirigido
02/set	13-14	Divisão e Conquista (D&C) e Análise: Teorema Mestre.
03/set	15-16	D&C e Cota Inferior : MergeSort
07/set	17-18	Feriado
08/set	19-20	D&C: Medianas, QuickSort, exponencial
09/set	21-22	Estudo dirigido
10/set	23-24	Algoritmos Gulosos: Conceitos. Seleção de Atividades.
14/set	25-26	Algoritmos Gulosos: Árvores Geradoras Mínimas (Kruskal & Prim).
15/set	27-28	Gulosos: Huffman, Horn, Mochila.
16/set	29-30	Trabalho prático 1
17/set	31-32	Prova 1 – atividade síncrona
21/set	33-34	Programação Dinâmica (PD): Conceitos. DAG. Maior Sequência Crescente.
22/set	35-36	PD: Distância de Edição.
23/set	37-38	PD: Mochila.
24/set	39-40	PD: Multiplicação de Cadeia de Matrizes.
28/set	41-42	Estudo dirigido
29/set	43-44	Busca Exaustiva e Busca Exaustiva Inteligente (BEI)
30/set	45-46	BEI: Backtracking e Branch-and-bound
01/out	47-48	BEI: Branch-and-bound
05/out	49-50	Redução de Problemas
06/out	51-52	Algoritmos Aproximados
07/out	53-54	Estudo dirigido
08/out	55-56	Problemas de Busca: Decisão e Otimização.
12/out	57-58	Feriado
13/out	59-60	Classes de Complexidade P, NP, NP-Difíceis e NP-Completo.
14/out	61-62	Trabalho prático 2
15/out	63-64	Prova 2 – atividade síncrona
19/out	65-66	Exame especial
20/out	67-68	Exame especial
21/out	69-70	Exame especial
22/out	71-72	Finalização das atividades da disciplina

Bibliografia básica:

ZIVIANI, Nivio. Projeto de algoritmos: com implementações em Pascal e C. 3. ed. rev. e ampl. Cengage Learning, 2010. 660 p. Disponível em <https://integrada.minhabiblioteca.com.br/#/books/9788522126590/cfi/0!/4/2@100:0.00>

ASCÊNCIO, Ana Fernanda Gomes; ARAÚJO, Graziela Santos de. Estrutura de dados: algoritmos, análise da complexidade e implementações em Java e C/C++. 1. ed. Pearson,

2010. 450 p.

Disponível em <https://plataforma.bvirtual.com.br/Leitor/Publicacao/1995/pdf/0>

DASGUPTA, Sanjoy; PAPADIMITRIOU, Christos H.; VAZIRANI, Umesh Virkumar. Algoritmos. São Paulo: McGraw-Hill, 2009. Disponível em:

<http://algorithmics.lsi.upc.edu/docs/Dasgupta-Papadimitriou-Vazirani.pdf>

Bibliografia complementar:

SIMÕES-PEREIRA, J. M. S. Grafos e redes: teoria e algoritmos básicos. **Rio de Janeiro: Interciência**, 2013.

Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Publicacao/42049/pdf/0>

FORTNOW, Lance. The status of the P versus NP problem. **Communications of the ACM**, v. 52, n. 9, p. 78-86, 2009.

Disponível em:

https://dl.acm.org/doi/fullHtml/10.1145/1562164.1562186?casa_token=jnbLmyHmaIsAAAAA:8fAZSIa-8Lq59aRq1ndDwHZuA8qHkSA3ADHoaNkvpPWkq34L7OzpP13UDa1XVIM2h9jXaZbjf-NgQ

PEREIRA, Silvio do Lago. Estruturas de dados em C: uma abordagem didática. 1. ed. Érica, 2018. 264 p.

Disponível em:

<https://integrada.minhabiblioteca.com.br/#/books/9788536517254/cfi/0!/4/2@100:0.00>

STEIN, Clifford; DRYSDALE, Robert L.; BOGART, Kenneth. Matemática discreta para ciência da computação. Pearson Education do Brasil. 2013.

<https://plataforma.bvirtual.com.br/Acervo/Publicacao/3824>

ROSEN, Kenneth H. Matematica discreta e suas aplicações. 6. ed. São Paulo: McGraw-Hill, c2009. xxi, 982 p. ISBN 9788577260362.

<https://integrada.minhabiblioteca.com.br/#/books/9788563308399/cfi/0!/4/2@100:0.00>