Instituto de Computação/Unicamp – Complexidade de Algoritmos I – M0417A

Prof. Lehilton Lelis Chaves Pedrosa

Programa

Este é o plano de desenvolvimento da disciplina e um guia de estudos. Leia-o com atenção e consulte este documento durante todo o semestre. Também, sempre acompanhe os avisos na página da disciplina: https://www.ic.unicamp.br/~lehilton/mo417a/.

Objetivos

Ao final do curso, @ alun@ deverá ser capaz de:

- modelar problemas e subproblemas comuns: ordenação e problemas em grafos;
- projetar e desenvolver algoritmos utilizando técnicas clássicas: divisão e conquista, algoritmos gulosos e programação dinâmica;
- analisar a correção e a complexidade de algoritmos de maneira formal;
- descrever a teoria de NP-completude e suas implicações.

Pré-requisitos

Ao início do curso, @ alun@ deve ser capaz de:

- descrever e desenvolver algoritmos com comandos de repetição e recursão;
- aplicar conceitos matemáticos: conjuntos, relações, funções, somatórios, lógica (proposicional) e demonstrações.

Atividades

Aulas: Antes das aulas, @s alun@s devem ler os capítulos ou seções do livro-texto correspondes. Os assuntos e datas serão divulgados na página da disciplina com antecedência. Durante as aulas, @s alun@s devem participar levantando dúvidas, sugestões, ou possivelmente resolvendo os problemas solicitados. Após as aulas, serão propostos diversos exercícios de fixação (retirados do livro-texto e de outras fontes) para serem feitos em casa. O conteúdo das listas é considerado parte integrante do curso e @s alun@s devem resolvê-los para se prepararem para as avaliações. Recomenda-se tentar fazer as atividades individualmente e, só então, discutir em grupo.

Teste individual (avaliação): Serão realizados 11 testes de unidade **individuais e sem consulta** com duração de 15 a 30 min (especificado em cada teste), no horário das aulas, em datas a serem divulgadas na página da disciplina. Alguns dos testes conterão questões selecionadas dentre os exercícios de fixação, com conteúdo igual ou com pequenas modificações. A nota de cada teste valerá de 0 a 10.

Exercícios (avaliação): Serão divulgadas 11 listas de exercícios (retirados do livro-texto e de outras fontes), cada uma com prazo de entrega de uma semana, em datas a serem divulgadas na página da disciplina.

- A nota de cada lista valerá de 0 a 10 e corresponderá à correção de **uma ou duas questões**, que serão sorteadas **somente depois** do prazo de entrega. O número de questões a serem corrigidas será especificado em cada lista.
- Para cada lista, @ alun@ deverá elaborar individualmente um documento em PDF que contém as resoluções de todas as questões e submeter no endereço https://www.ic.unicamp.br/~lehilton/mo417a/submit/ usando a chave fornecida na primeira aula. Recomendação: para aumentar a legibilidade, crie o documento usando Latex, ou, se preferir escrever à mão, obtenha uma imagem do manuscrito; nesse caso, utilize um escâner de mesa ou um aplicativo de celular com a função scan para criar um arquivo PDF. Trabalhos copiados, não identificados, desorganizados, incompletos, ilegíveis, ou rasurados terão a nota zerada.

Avaliação

Serão calculadas as médias aritméticas dos testes individuais (T) e dos exercícios (E). Para se aprovar, o @ alun@ deve satisfizer todos os critérios abaixo:

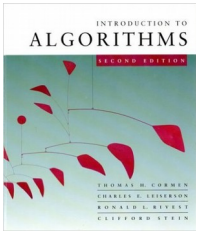
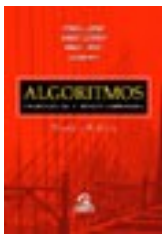
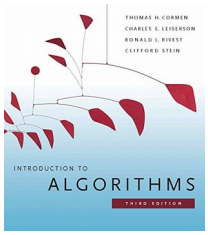
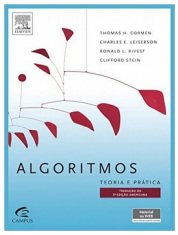
1. 75% de frequência;
2. T ≥ 5; E ≥ 7.

Para aprovad@s, a média será M := (T+E)/2, do contrário, M := 4. O conceito será:

$$\begin{cases} \text{A} & \text{se} \quad M \geq 8{,}5\,; \\ \text{B} & \text{se} \quad 7 \leq M < 8{,}5\,; \\ \text{C} & \text{se} \quad 6 \leq M < 7\,; \\ \text{D} & \text{se} \quad M < 6\ . \end{cases}$$

Bibliografia

Será adotado como livro-texto: “Algoritmos – Teoria e Prática”, de T. Cormen, C. Leiserson, R. Rivest, C. Stein (CLRS). Poderá ser usada a segunda ou a terceira edição, tanto na versão em inglês quanto na versão traduzida. Por causa das diferentes numerações entre as edições, os exercícios solicitados do livro poderão ser transcritos.

	inglês	português
2ª edição	[1] 	[2] 
3ª edição	[3] 	[4] 

[1] T. Cormen, C. Leiserson, R. Rivest, C. Stein. Introduction to Algorithms (2ª edição), 2001.
[2] T. Cormen, C. Leiserson, R. Rivest, C. Stein. Algoritmos – Teoria e Prática (2ª edição), 2002.
[3] T. Cormen, C. Leiserson, R. Rivest, C. Stein. Introduction to Algorithms (3ª edição), 2009.
[4] T. Cormen, C. Leiserson, R. Rivest, C. Stein. Algoritmos – Teoria e Prática (3ª edição), 2012.

Além do livro de CLRS, em algumas unidades, também será utilizado o livro de Udi Manber, particularmente nas unidades referentes a indução e projeto de algoritmos baseado em indução.

[5] U. Manber, Algorithms. A Creative Approach, 1989.

Além desses, poderão ser consultados outros livros, disponíveis na biblioteca, como o livro clássico de Donald Knuth, que é considerado o pai da análise de algoritmo.

[6] D. E. Knuth. The Art of Computer Programming, 1974.

[5] 

[6] 

Há alguns livros pouco mais recentes, como o livro do Jon Kleinberg e da Éva Tardos, que inclui, além dos tópicos estudados, alguns capítulos para tratamento de problemas NP-difíceis,
[7] J. Kleinberg, E. Tardos. Algorithm Design, 2005.

e o livro de Sanjoy Dasgupta, Christos Papadimitriou e Umesh Vazirani, que é um pouco mais informal e com um ritmo diferente do CLRS.
[8] S. Dasgupta, C. Papadimitriou, U. Vazirani. Algorithms, 2006.

Livros de autores brasileiros importantes também estão disponíveis.
[9] J. L. Szwarcfiter. Grafos e Algoritmos Computacionais, 1984.
[10] N. Ziviani. Projeto de Algoritmos (2ª edição), 2004.

A teoria de NP-completude e os modelos de computação serão tratados apenas superficialmente nesse curso e utilizando a abordagem de CLRS. Um tratamento mais aprofundado de Teoria da Computação e com a abordagem mais tradicional é assunto da disciplina Linguagens Formais e Autômatos, que tem entre referências clássicas o livro de Sipser.
[11] M. Sipser. Introduction to the Theory of Computation (3ª edição), 2012.

Já um aprofundamento em complexidade pode ser encontrado no livro de Arora e Barak.
[12] S. Arora and B. Barak. Computational Complexity: A Modern Approach, 2009.

Uma leitura mais gentil sobre modelos computacionais e reduções e recomendada para alun@s desse curso é o primeiro capítulo do livro dos professores Rezende e Stolfi.
[13] P. J. de Rezende, J. Stolfi. Fundamentos de geometria computacional, 1994. Disponível em: https://www.ic.unicamp.br/~rezende/rez-sto-94-fgc.pdf.

Não menos importante, na internet há diversos e excelentes recursos que podem servir para pesquisa. Em particular, veja o curso de análise de algoritmos do prof. Paulo Feofiloff.
[14] P. Feofiloff. Análise de Algoritmos. https://www.ime.usp.br/~pf/analise_de_algoritmos/.

Um bom conjunto de notas de aulas está livremente disponível.
[15] J. Erickson. Algorithms, Etc. http://jeffe.cs.illinois.edu/teaching/algorithms/.

E, lendo com cuidado, a Wikipédia é uma excelente fonte de consulta.
[16] Wikipédia. https://en.wikipedia.org/wiki/Analysis_of_algorithms.

Sugestão: Não tente ler várias referências de uma vez; na maior parte do conteúdo, atenha-se ao livro-texto ([1]-[4]) e consulte o livro [5] nos capítulos em que ele for usado. Consulte a bibliografia complementar sempre que solicitado, para se aprofundar, ou se tiver dificuldade com algum assunto do livro-texto e quiser uma apresentação diferente. Ler os exercícios de outras fontes (e tentar resolver os que achar mais interessantes) também é uma boa maneira de estudar. Não leia assuntos mais aprofundados (sobre Teoria da Computação e Complexidade Computacional) antes de entender o conteúdo básico correspondente tratado por CRLS. Finalmente, antes de começar a ler e fazer exercícios, certifique-se de que entendeu o espírito, a forma e o porquê de uma demonstração matemática. Se não, então o link O que é uma prova matemática? da referência [14] é um bom lugar para começar a estudar.

Material didático

Os slides usados serão disponibilizados. A maioria dos slides e exercícios adicionais foram ou criados e gentilmente cedidos pelo prof. Cid Carvalho de Souza e pela profa. Cândida Nunes da Silva (particularmente com modificações do prof. Orlando Lee); ou criados e gentilmente cedidos pelo prof. Flávio Keidi Miyazawa. Eu adaptei o material disponibilizado e possivelmente introduzi erros, que podem ser reportados a mim.

Horário e local

As aulas serão ministradas na sala 352, das 14 h às 16 h, segundas e quartas-feiras.

Atendimento

Poderá ser combinado um horário para atendimento com o professor via e-mail por um@ ou mais alun@s, desde que solicitado com, pelo menos, três dias de antecedência.

Rotina de estudo

Importante: Note que não há provas final e de meio de semestre. Portanto é fundamental criar uma rotina de estudos contínua. Planeje-se e separe algumas horas e alguns dias por semana para ler o livro e resolver as listas de exercícios. @s alun@s também são encorajados a se reunir e estudar em grupo (sempre depois de tentar resolver os exercícios individualmente). Cada um@ tem sua própria maneira de estudar. Não obstante, algumas sugestões são úteis para o bom desenvolvimento da disciplina:

1. Leia o capítulo ou seção do livro correspondente ao conteúdo antes da aula correspondente (veja a ordem dos conteúdos abaixo); utilize a aula principalmente para tirar dúvidas e confirmar o seu entendimento.
2. Faça ou tente fazer os exercícios correspondentes a uma aula no próximo horário que tiver reservado para estudar a disciplina; anote as principais dificuldades e discuta com colegas e, persistindo, com o professor no início das próximas aulas.
3. Veja o resultado das avaliações! Elas servem para que você identifique os problemas (erro de lógica, incorreção, conceitos incorretos, dificuldade com formalismo, incompletude, erros de português e escrita matemática, etc.). Tente refazer as tarefas e, se não puder identificar o que está errado ou não conseguir corrigir o problema, anote a dúvida e leve-a ao professor (mas evite pedir aumento de nota, ou comparar notas de colegas).

Organização do curso

O curso tem 11 unidades. A ordem de unidades e de conteúdos abaixo serve para que @ alun@ se planeje e estude para as aulas. A ordem pode variar dependendo do andamento da disciplina e as datas serão divulgadas na página da disciplina.

i - Introdução
- Introdução à análise de algoritmos
- Notação assintótica e crescimento de funções

ii - Indução
- Princípio da indução
- Invariantes de laço e demonstração de correção
- Projeto de algoritmos por indução

iii - Divisão e conquista
- Projeto de algoritmos baseados em divisão e conquista
- Recorrências para algoritmos de divisão e conquista

iv - Ordenação
- Visão geral de algoritmos de ordenação
- Fila de prioridade e Heapsort
- Ordenação por particionamento
- Ordenação em tempo linear
- Estatísticas de ordem

v - Técnicas de projeto de algoritmos avançadas
- Programação Dinâmica
- Algoritmos Gulosos

vi - Algoritmos e conceitos fundamentais de grafos
- Conceitos de grafos
- Fatos básicos de grafos
- Representação de grafos

vii - Buscas em grafos

- Busca em largura
- Busca em profundidade
- Ordenação topológica
- Componentes fortemente conexas

viii - Caminhos mínimos

- Caminhos mínimos com uma origem
- Algoritmo de Dijkstra
- Algoritmo de Bellman-Ford
- Caminhos mínimos entre todos os pares de vértices

ix - Árvore geradora mínima

- Arvore geradora mínima e algoritmo de Prim
- Algoritmo de Kruskal e conjuntos disjuntos
- Conjuntos disjuntos com florestas disjuntas

x - Redução entre problemas

- Conceitos de redução entre problemas
- Exemplos de reduções

xi - NP-completude

- Classes de problemas e problemas polinomiais
- Problemas verificáveis em tempo polinomial
- Demonstrações de NP-completude