
MCTA003-17 - Análise de Algoritmos - Noturno (2019 Q2)

Professora: Carla Negri Lintzmayer, Sala 508-2,
carla.negri@ufabc.edu.br

Avisos importantes (fique atento sempre!)

[24/05] Atenção: não haverá aula no dia 5/06. A primeira aula será no dia 7/06 (sexta-feira).

[24/05] Página da disciplina no ar.

Conteúdo dessa página



[Dias, horários e local das aulas](#)
[Dias, horários e local de atendimento](#)
[Ementa da disciplina](#)
[Recomendação](#)
[Bibliografia e outros materiais](#)
[Cronograma e notas de aula](#)
[Plágio](#)
[Listas de exercícios](#)
[Critérios de avaliação](#)
[Mecanismo de recuperação](#)
[Mecanismos de avaliação substitutivos](#)

Dias, horários e local das aulas ([voltar ao topo](#))

Quartas-feiras, das 21h às 23h, sala A-108-0.

Sextas-feiras, das 19h às 21h, sala A-108-0.

Dias, horários e local de atendimento ([voltar ao topo](#))

Quintas-feiras, das 13h15 às 15h15 e das 17h às 19h, na sala 508-2 do bloco A.

Ementa da disciplina ([voltar ao topo](#))

(Disponível na pg. 54 do [projeto pedagógico](#).)

Conceitos básicos: recorrências, medidas de complexidade: melhor caso, caso médio e pior caso. Técnicas gerais de projeto de algoritmos: divisão e conquista, método guloso e programação dinâmica. Classes de complexidade: P, NP e NP-completude.

Objetivos: (i) Apresentar noções e conceitos de complexidade de computação; (ii)

Apresentar métodos e conceitos que permitam ao aluno, de maneira confiável, avaliar a qualidade de um algoritmo. A essência destes métodos e conceitos estará focalizada no cálculo de complexidade e prova de corretude de algoritmos; (iii) Caracterizar técnicas

gerais de desenvolvimento de algoritmos que permitam ao aluno melhor projetá-los conforme sua natureza. As técnicas gerais escolhidas a serem estudadas são Divisão e Conquista, Método Guloso e Programação Dinâmica; (iv) Apresentar noções básicas de Classes de Complexidade, em particular as classes P, NP e NP-Completo.

Recomendação ([voltar ao topo](#))

Disciplinas: Matemática Discreta; Algoritmos e Estruturas de Dados I.

Para facilitar o acompanhamento do curso, é recomendado que você possua:

- conhecimentos de programação (em qualquer linguagem imperativa), com boas noções de algoritmos recursivos,
- familiaridade com estruturas de dados básicas (vetores, listas, pilhas, filas e árvores),
- capacidade para reconhecer argumentos lógicos em uma prova matemática (por indução, contradição, construção),
- familiaridade com linguagem matemática (como quantificadores lógicos, somatórios e manipulação de funções).

Materiais de apoio para esses tópicos:

- [O que é uma prova matemática](#), do prof. Paulo Feofiloff, da USP.
- [Matemática discreta para computação](#), dos profs. Anamaria Gomide e Jorge Stolfi, da Unicamp.
- [Indução matemática](#), do prof. Cid Carvalho de Souza, da Unicamp.
- [Portal da Matemática da OBMEP](#).
- [Indução e contagem](#), do prof. Rogério Steffenon e Felipe Guarnieri, da Unisinos.
- [Projeto de algoritmos \(em C\)](#), do prof. Paulo Feofiloff, da USP.
- [Estruturas de dados \(em C\)](#), do prof. Paulo Feofiloff, da USP.
- [Notas de aula](#), da disciplina de Programação Estruturada, da prof. Carla Lintzmayer (introdução à programação em C, recursão, vetores e listas).

Bibliografia e outros materiais ([voltar ao topo](#))

[CLRS2] Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Stein, C.. *Introduction to Algorithms*. 2nd ed. MIT Press. 2002.

[CLRS3] Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Stein, C.. *Introduction to Algorithms*. 3rd ed. MIT Press. 2009.

[LM] Lintzmayer, C. N.; Mota, G. O.. [Notas de aulas - Análise de algoritmos e estruturas de dados](#). (em constante atualização).

[R] Vídeo aulas do prof. Tim Roughgarden, de Stanford, em inglês (com legendas): [parte 1](#) e [parte 2](#).

[DPV] Dasgupta, S.; Papadimitriou, C.; Vazirani, U.. *Algorithms*. Boston: McGraw-Hill. 2008.

[KT] Kleinberg, J.; Tardos, E.. *Algorithm design*. Pearson/Addison-Wesley. 2006.

Site com execução passo a passo de vários [algoritmos em grafos](#).

[GFG] Site [Geeks for Geeks](#), muito bom, com implementações em várias linguagens.

Grupo de [Whatsapp](#) criado e mantido pelos alunos.

Cronograma ([voltar ao topo](#))

As referências utilizadas nas aulas serão atualizadas durante o curso.

Sobre qualquer material feito por mim, participe do [banco de informantes](#).

| Aula | Data | Tópicos |
|------|-------------|---|
| 1 | 05/6 | Não haverá aula presencial: revisar base matemática e recursão. Referências: material extra e outros |
| 2 | 07/6 | Introdução ao curso. Multiplicação e Fibonacci. Referências: [LM], notas de aula , Cap. 1 [CLRS2], Cap. 0.2 [DPV], 1.1, 1.2 e 1.3 [R]. |
| 3 | 12/6 | Insertion Sort com análise. Notação assintótica. |
| 4 | 14/6 | Notação assintótica. |
| 5 | 19/6 | Divisão e Conquista. Mergesort. Recorrências. |
| | 21/6 | ---- Feriado ---- |
| 6 | 26/6 | Solução de recorrências: substituição e iteração. |
| 7 | 28/6 | Solução de recorrências: árvore de recursão e Teorema Mestre. |
| 8 | 03/7 | Quicksort. |
| 9 | 05/7 | Selection sort. Heap. Heapsort. |
| 10 | 10/7 | Grafos. Busca em grafos. |
| 11 | 12/7 | Busca em grafos. Aplicações. |
| 12 | 17/7 | Ajuste. Solução de exercícios. |
| 13 | 19/7 | Prova 1 |
| 14 | 24/7 | Introdução a algoritmos gulosos. Escalonamento de tarefas compatíveis. Mochila fracionária. |
| 15 | 26/7 | Árvore geradora mínima. Algoritmo de Kruskal. Union-Find. |
| 16 | 31/7 | Compressão de dados. Algoritmo de Huffman. |
| 17 | 02/8 | Introdução a Programação Dinâmica. Corte da barra de ferro. Mochila inteira. |
| 18 | 07/8 | Alinhamento de seqüências. Algoritmo de Needleman-Wunsch. |
| 19 | 09/8 | Caminhos mínimos. Algoritmo de Floyd-Warshall. |
| 20 | 14/8 | Redução. Classes P, NP, NP-difícil e NP-completo. |
| 21 | 16/8 | Abordagens para problemas NP-difíceis. |
| 22 | 21/8 | Ajuste. Solução de exercícios. |
| 23 | 23/8 | Prova 2 |
| 24 | 28/8 | Prova de recuperação (reposição do feriado, portanto aula às 19h) |

Plágio ([voltar ao topo](#))

- Entre outros, o código de ética da UFABC estabelece em seu artigo 25 que é eticamente inaceitável que os discentes:
 1. fraudem avaliações,
 2. fabriquem ou falsifiquem dados,
 3. plagiem ou não creditem devidamente autoria,
 4. aceitem autoria de material acadêmico sem participação na produção,
 5. vendam ou cedam autoria de material acadêmico próprio a pessoas que não participaram da produção.
- Muitos ainda têm dúvidas sobre a interpretação das regras definidas pelo Código de Ética da UFABC.

- Por esta razão, diversos professores elaboraram um documento ([disponível aqui](#)) com vários exemplos e esclarecendo a interpretação das regras acima.
- Abaixo uma versão resumida, que não substitui de modo algum sua leitura.
- Sempre consulte o documento completo ou **converse com o seu professor em caso de dúvidas!**
 - *Regra 1:* Você não pode enviar para avaliação um trabalho que não seja de sua própria autoria ou que seja derivado/baseado em soluções elaboradas por outros.
 - *Regra 2:* Você não pode compartilhar a sua solução com outros alunos nem pedir aos seus colegas que compartilhem as soluções deles com você.
 - *Regra 3:* Nos trabalhos enviados para avaliação você deve indicar eventuais assistências que você tenha recebido.
 - Nós encorajamos fortemente que você procure outras pessoas quando houver a necessidade. Discuta o problema e possíveis ideias para soluções, mas elabore sua própria solução, por conta própria.
 - Qualquer violação às regras descritas acima implicará em descarte dos conceitos atribuídos a TODAS as tarefas avaliativas regulares de TODOS os envolvidos, causando assim suas reprovações automáticas com conceito F.
 - Possível denúncia à Comissão de Transgressões Disciplinares Discentes da Graduação, a qual decidirá sobre a punição adequada à violação que pode resultar em advertência, suspensão ou desligamento, de acordo com os artigos 78-82 do Regimento Geral da UFABC.

Listas de exercícios ([voltar ao topo](#))

- Ao todo teremos 4.5 listas, cujos enunciados serão disponibilizados aqui, nas datas indicadas abaixo.
- As soluções das listas de exercícios deverão ser feitas **à mão** (capriche na letra!), escaneadas e **um único arquivo formato PDF** deve ser entregue (sugestão de aplicativo: CamScanner).
- A entrega deverá ser feita **apenas** pelo [tidia](#) (a menos que haja problemas no tidia nas datas em questão, caso em que elas podem ser entregues por e-mail).
- Todos os prazos de entrega já estão definidos (marque na sua agenda!).
- Soluções entregues fora do prazo valerão 0.

Lista 1: notação assintótica e divisão e conquista (disponível em 12/06 - entrega até 25/06)

Lista 2: recorrências e ordenação (disponível em 19/06 - entrega até 09/07)

Lista 2.5: grafos (disponível em 10/07 - entrega até 18/07)

Lista 3: gulosos (disponível em 24/07 - entrega até 04/08)

Lista 4: programação dinâmica e complexidade (disponível em 02/08 - entrega até 20/08)

Critérios de avaliação ([voltar ao topo](#))

- A avaliação da disciplina constituirá da nota de duas provas, respectivamente denotadas por P1 e P2, e da média simples das notas das listas de exercícios, denotada por L.
- A prova 1 vale 35% da nota.
- A prova 2 vale 45% da nota.
- As listas valem 20% da nota.
- Sua média final (MF) antes da recuperação, portanto, será

$$MF = 0.35 \times P1 + 0.45 \times P2 + 0.2 \times L$$

- Seu conceito final será

- A, se $MF \geq 8.5$
- B, se $7.0 \leq MF < 8.5$
- C, se $6.0 \leq MF < 7.0$
- D, se $2.0 \leq MF < 6.0$
- F, se $MF < 5.0$

Mecanismo de recuperação ([voltar ao topo](#))

- A recuperação será aplicada apenas aos alunos que tiverem conceito final D ou F e cujas ausências não excederem 25% da quantidade de aulas.
- Consistirá numa prova, em formato similar às aplicadas ao longo do curso.
- O conteúdo da prova englobará todos os temas vistos durante o quadrimestre.
- A nota obtida na prova de recuperação (NR) será usada obter a nota final com recuperação (NFR), que consiste na média a seguir:

$$NFR = \max \{MF, (MF + NR) / 2\}$$

- O conceito final obtido na recuperação substituirá o conceito original e será
 - C, se $NFR \geq 6.0$
 - D, se $5.0 \leq NFR < 6.0$
 - F, se $0.0 \leq NFR < 5.0$

Mecanismos de avaliação substitutivos ([voltar ao topo](#))

A prova substitutiva será aplicada ao aluno que possuir justificativa de ausência em uma das provas. A listagem dos documentos aceitos como justificativa consta na resolução [ConsEPE n° 181](#).

A nota obtida na prova substitutiva necessariamente substituirá a prova para a qual o aluno tem justificativa.

A data para realização da prova substitutiva deverá ser combinada com a professora por e-mail o quanto antes, assim que o aluno estiver em condições de realizá-la.