

Professor: Maycon Sambinelli

E-mail: m.sambinelli@ufabc.edu.br📧 [Caixa de Sugestões](#)

Avisos importantes

Objetivos

- Apresentar noções e conceitos de complexidade de computação;
- 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 correte de algoritmos;
- 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;
- Apresentar noções básicas de Classes de Complexidade, em particular as classes P, NP e NP-Completo.

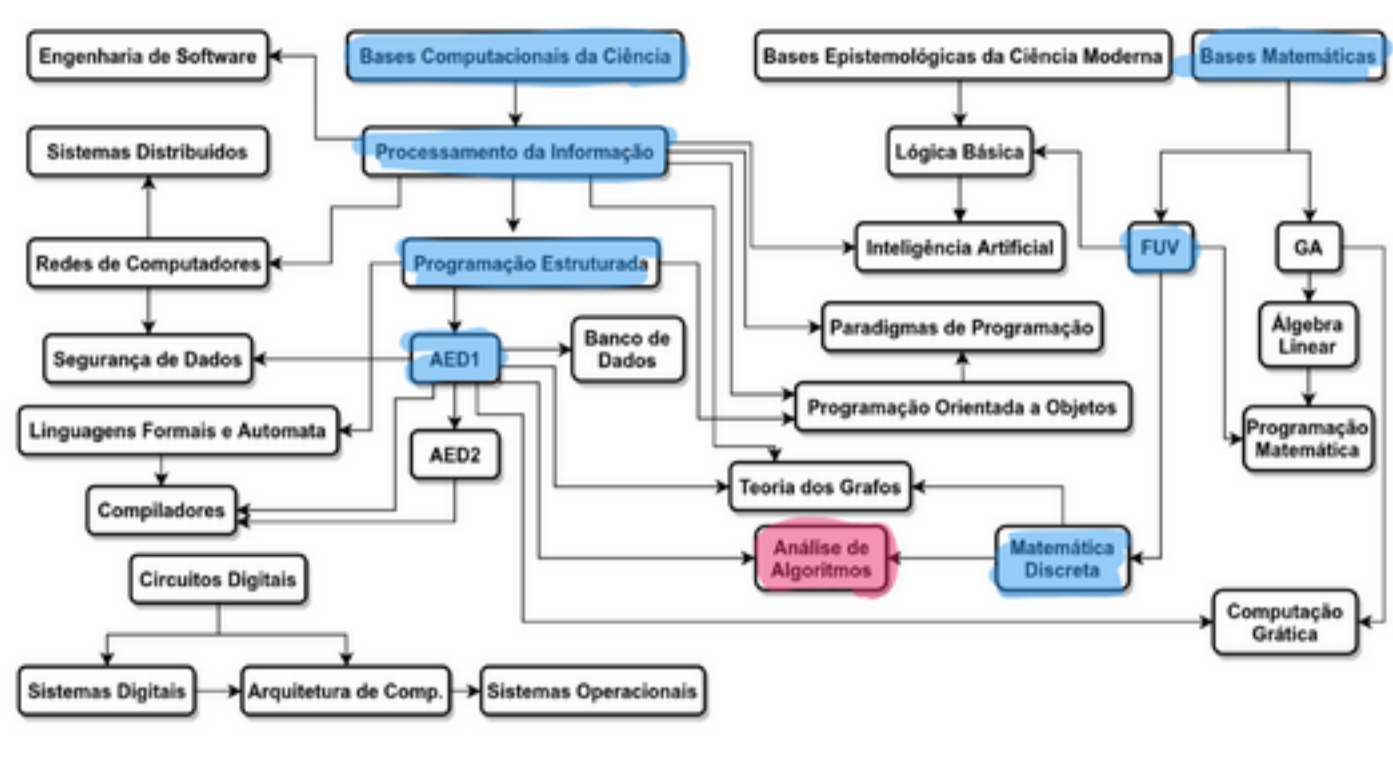
Ementa da disciplina

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.

Recomendações

Disciplinas: Matemática Discreta, Algoritmos e Estruturas de Dados I

ANEXO G: FLUXOGRAMA DAS RECOMENDAÇÕES ENTRE DISCIPLINAS OBRIGATORIAS DO BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO



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).

Outros materiais de apoio:

- [Livro de Bases Matemáticas](#), dos profs. Armando Caputi e Daniel Miranda, da UFABC.
- [Fundamentos da matemática para computação](#), (vídeoaulas) do prof. Cláudio Possani, da USP.
- Minhas aulas de [revisão de Matemática Discreta](#).
- Aulas de revisão de indução da profa. Carla Lintzmayer: [definição e exemplos](#).
- [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 Estruturas de Dados do prof. Rafael Schouery, da Unicamp (introdução à programação em C, recursão, listas, pilhas e filas, árvores).
- [Notas de aula](#) da disciplina de Programação Estruturada da profa. Carla Lintzmayer (introdução à programação em C, recursão, vetores e listas).

Bibliografia e outros materiais

1. [LM] Lintzmayer, C. N.; Mota, G. O. [Notas de aulas - Análise de algoritmos e estruturas de dados](#) (o conteúdo dessa disciplina está completo no livro, mas sempre existem atualizações - verifique sempre sua versão).
2. [CLRS3] Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Stein, C.. *Introduction to Algorithms*. 3rd ed. MIT Press. 2009.
3. [CLRS2] Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Stein, C.. *Introduction to Algorithms*. 2nd ed. MIT Press. 2002.

- [colinha](#)

Critérios de avaliação regular

A média final antes da REC (MF) será calculada da seguinte forma:

$$MF = \sqrt{P1 \times P2},$$

onde,

- P1 e P2 são as notas da primeira e segunda avaliações, respectivamente.

$$CF = \begin{cases} \text{A,} & \text{se } MF \in [8,5; 10,0] \\ \text{B,} & \text{se } MF \in [7,0; 8,5) \\ \text{C,} & \text{se } MF \in [6,0; 7,0) \\ \text{D,} & \text{se } MF \in [5,0; 6,0) \\ \text{F,} & \text{caso contrário} \end{cases}$$

🚫 Caso seja verificado ocorrência de fraude acadêmica, o aluno será automaticamente reprovado com F.

Mecanismo de recuperação

- A recuperação será aplicada apenas aos alunos que tiverem conceito final **D** ou **F**.
- Consistirá numa avaliação, cujo conteúdo englobará todos os temas vistos durante o quadrimestre.
- A nota obtida na avaliação de recuperação (**NR**) será usada para obter a nota final com recuperação (**NFR**), que consiste na média:

$$NFR = \frac{MF + NR}{2}$$

- O conceito final com recuperação (**CFR**) será calculado da seguinte maneira:

$$CFR = \begin{cases} C, & \text{se } CF = D \text{ e } NFR \geq 6; \\ D, & \text{se } CF = D \text{ e } NFR < 6; \\ D, & \text{se } CF = F \text{ e } NFR \geq 5; \\ F, & \text{caso contrário.} \end{cases}$$

Plágio

🚫 **Leitura obrigatória** 🚫

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 acadêmica 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 Gradação, 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.

Datas importantes

- Avaliação 1: 13/07 (Qua)
- Avaliação 2: 24/08 (Qua)
- Recuperação: 24/09 (Sáb) às 10h:00 (sala a definir)

Notas

em breve

Aulas

06/06 – 1 Introdução ao Curso

08/06 – 2 Introdução à análise de algoritmos & Corretude de Algoritmos (iterativos)

13/06 – 3 Tempo de Execução

15/06 – 4 Tempo com notação assintótica e Insertion Sort

20/06 – 5 Corretude e tempo de execução de algoritmos recursivos & Recorrências

22/06 – 6 Divisão e Conquista & Mergesort

27/06 – 7 Recorrências

29/06 – 8 Recorrências

04/07 – 9 SelectionSort, Heap & Heapsort

06/07 – 10 Grafos

11/07 – 11 Busca em Grafos I

13/07 – 12 Avaliação 1 (P1)

18/07 – 13 Busca em Grafos II

20/07 – 14 Algoritmos Gulosos

25/07 – 15 Árvore Geradora Mínima

27/07 – 16 Programação Dinâmica

01/08 – 17 Programação Dinâmica II

03/08 – 18 Caminhos mínimos (Dijkstra)

08/08 – 19 Caminhos mínimos (Floyd-Warshall)

10/08 – 20 Reduções

15/08 – 21 Classes P, NP, NP-Completo, NP-Difícil

17/08 – 22

22/08 – 23

24/08 – 24 Avaliação 2 (P2)