

MCTA001 - Algoritmos e Estruturas de Dados I

Plano de ensino

Prof. Diogo S. Martins
Centro de Matemática, Computação e Cognição
Universidade Federal do ABC

Q1 2024
v.05/02

1 Informações básicas

- TPI: 2-2-4
- Horários oficiais:

Teoria	qui	10-12h	S204-0	semanal
Prática (DA1)	seg	08-10h	407-2	semanal
Prática (DA2)	seg	10-12h	407-2	semanal
- Ferramentas de comunicação:
 - Moodle
 - Email: santana.martins@ufabc.edu.br (informar disciplina e turma na mensagem)
 - Convite discord: <https://discord.gg/UC8JNbFRsk>
- Plantões:

qui	08-10h	S528-2
-----	--------	--------

O objetivo do plantão é esclarecer dúvidas e/ou reforçar temas vistos nas aulas ou outros materiais.
- Sala no Moodle:

<https://moodle.ufabc.edu.br/course/view.php?id=1444>

Todos já foram convidados para a sala, em caso de dificuldade de acesso, entrar em contato com o professor.

2 Objetivos

1. Apresentar as diversas estruturas de dados fundamentais, como estruturas lineares (listas encadeadas, pilhas, filas, etc.), estruturas não-lineares (árvores), os algoritmos básicos para a sua manipulação, assim como as suas aplicações;

2. Introduzir noções básicas de complexidade de algoritmos e técnicas básicas para comparação dos tempos de execução dos algoritmos estudados;
3. Apresentar a importância da escolha da estrutura de dados e algoritmos adequados para a resolução de problemas de maneira eficiente.

3 Ementa

- Breve introdução à linguagem C
- Estruturas lineares
- Noções básicas de análise de complexidade de tempo de algoritmos
- Árvores de busca. Árvores balanceadas.

4 Requisitos recomendados

Para participar dessa disciplina é recomendação oficial ter cursado e sido aprovado em:

- MCTA028 - Programação Estruturada

5 Bibliografia

1. CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. Algoritmos: teoria e prática. 2ª edição. Rio de Janeiro, RJ: Campus, 2002.
2. KNUTH, D. E. The art of computer programming. Upper Saddle River, USA: Addison-Wesley, 2005.
3. SEDGEWICK, R. Algorithms in C: parts 1-4 (fundamental algorithms, data structures, sorting, searching). Reading, USA: Addison-Wesley, 1998.

6 Metodologia de ensino-aprendizagem

- **Aulas teóricas.** Como não dispomos de computadores individuais, teremos, além da discussão dos conteúdos teóricos, também exercícios focados na análise de códigos prontos, questões mais conceituais, bem como a discussão de dúvidas levadas pelos alunos.
- **Aulas práticas.** Como neste espaço há disponibilidade de computadores individuais, teremos, além de complementação dos conceitos vistos em aula, também tutoriais e exercícios regulares de programação, com enunciados disponíveis no Moodle. O tempo em aula pode ser usado também para a discussão de dúvidas.

7 Avaliação

A avaliação consiste nos componentes dados pela Equação 1, onde:

$$N_F = 0.2 \cdot N_{atv} + 0.4 \cdot N_{P1} + 0.4 \cdot N_{P2} \quad (1)$$

- N_{atv} é a média das atividades com entrega no Moodle;
- N_{P1} e N_{P2} são as notas das provas;

O conceito final será obtido de acordo com a Equação 2.

$$C_F = \begin{cases} A, & \text{se } N_F \in [8.5, 10.0] \\ B, & \text{se } N_F \in [7.0, 8.5) \\ C, & \text{se } N_F \in [6.0, 7.0) \\ D, & \text{se } N_F \in [5.0, 6.0) \\ F, & \text{se } N_F \in [0.0, 5.0) \\ 0, & \text{se ausência exceder 25\%} \end{cases} \quad (2)$$

Sobre as Atividades

As atividades consistem em listas de exercícios que devem ser entregues no Moodle.

Estrutura. Cada atividade será composta por um ou mais exercícios.

Frequência. Teremos uma atividade por semana, exceto nas semanas da prova.

Prazo. Cada atividade tem prazo de 15 dias para entrega. Não serão aceitas atividades entregues com atraso.

Verificação automática. As atividades contarão, sempre que possível, com recursos de verificação automática de corretude, com o objetivo de auxiliar o estudante a detectar, de modo automatizado, problemas recorrentes de programação. Na ausência de recursos de verificação automática, o estudante contará com uma rubrica para analisar a aderência do seu resultado ao que é esperado. Cabe ao aluno garantir que o programa passe por *todas* as verificações automáticas. Além disso, nas atividades em que haja rubrica, deve garantir que todos os critérios sejam atendidos. Submissões com falhas terão descontos substanciais na nota.

Avaliação somativa. Além dos testes automáticos, o professor analisará as submissões em atenção aos aspectos qualitativos e éticos detalhados neste plano de ensino. Ou seja, passar em todos os testes automáticos não implica necessariamente em nota máxima, visto que análise posterior pode implicar em desconto parcial ou total na nota da atividade.

Sobre as Provas

As provas terão as seguintes características:

- Presenciais;

- Com duração de 100 minutos;
- Efetuadas no computador do laboratório (não é permitido o uso de computador pessoal nem de dispositivos móveis);
- Sem consulta;
- Entregue no Moodle.

O prazo será controlado automaticamente pelo Moodle. Não serão aceitas, em hipótese alguma, provas entregues fora do prazo regulado pelo Moodle ou enviadas por meios de submissão alternativos.

Critérios de avaliação

A nota máxima de cada componente de avaliação (atividades e provas) será obtida apenas se o mesmo for entregue no prazo e executado correta e completamente.

Os programas solicitados em componentes de avaliação serão submetidos aos seguintes tipos de verificações:

- **Verificações automáticas.** Executadas pelo sistema de submissão.
 - **Sintaxe e semântica.** O programa deve compilar e executar sem erros ou avisos. **Programas que apresentem erros de compilação terão desconto de 100% na nota.**
 - **Corretude.** Os programas serão submetidos a testes unitários — falha em qualquer um dos testes resultará em desconto parcial ou total na nota, dependendo do peso do respectivo teste.
 - **Autenticidade.** Os programas serão submetidos a testes de autenticidade para detectar plágio (de outros alunos, de outros quadrimestres, da web, de chatbots, etc.). Cópias implicarão nas sanções previstas no código de honra.
- **Verificações manuais.** O professor inspecionará os programas para verificar os seguintes critérios gerais:
 - **Eficiência:** os programas desenvolvidos deverão ter bom desempenho, o que pode englobar o tratamento adequado dos seguintes fatores:
 - * Ler e escrever dados nas quantidades mínimas necessárias para resolver o problema;
 - * Não desperdiçar memória primária (RAM);
 - * Acessar memória secundária (disco) somente quando necessário e sem redundância;
 - * entre outros.
 - **Acurácia:** o programa deverá atender adequadamente a todos os requisitos enunciados para a atividade;
 - **Estrutura e organização do código:** atentar principalmente aos seguintes aspectos:
 - * **Modularização:** modelar as estruturas e algoritmos de modo a favorecer o reuso, isto é, que possam ser usados por diferentes programas principais, em diferentes contextos, sem a necessidade de “copiar e colar” código;

- * **Auto-documentação:** nomes intuitivos para variáveis, funções e módulos;
- * **Comentários:** documentação completa porém ao mesmo tempo concisa (sem poluição visual, apenas nos lugares adequados e necessários), sendo que os comentários explicam o que o código faz, e não como faz;
- **Autenticidade:** o código é original e não foi copiado de outras fontes (i.e. web, livros, terceiros, etc.), sob pena das sanções previstas no código de honra.

Mecanismos de avaliação substitutivos

É preciso apresentar justificativa prevista na Resolução ConsEPE 227. O documento de justificativa deve ser encaminhado por email ao professor, que cadastrará o aluno na prova. A prova substitutiva será realizada na data definida na seção 9.

Mecanismo de recuperação

A recuperação será aplicada apenas aos alunos que tiverem conceito final D ou F. Ocorrerá na data definida na seção 9, em formato similar ao estabelecido para as provas regulares.

A nota obtida na prova de recuperação (N_R) será usada para obter a nota final com recuperação (N_{FR}), que consiste na média estabelecida pela Equação 3.

$$N_{FR} = 0.5 \cdot N_F + 0.5 \cdot N_R \quad (3)$$

O conceito final obtido na recuperação (C_{FR}) é o conceito que entrará no histórico, obtido de acordo com os limiares para a nota final de recuperação (N_{FR}) dados pela Equação 4.

$$C_{FR} = \begin{cases} C, & \text{se } N_{FR} \geq 6.0 \\ D, & \text{se } N_{FR} \in [5.0, 6.0) \\ F, & \text{se } N_{FR} < 5.0 \end{cases} \quad (4)$$

No caso dos alunos que não participarem da recuperação, $C_{FR} = C_F$.

8 Código de honra

A aprovação na disciplina é baseada exclusivamente no esforço e trabalho pessoal do discente, ao qual cabe garantir que não ajudará ou receberá ajuda não-permitida em qualquer atividade avaliativa (e.g. provas, trabalhos, listas, etc.).

Exemplos de violação do código de honra incluem:

- Copiar atividades avaliativas (e.g. listas, trabalhos, provas, etc.) ou permitir que outros discentes copiem suas atividades avaliativas;
- Colaboração não-permitida entre indivíduos ou grupos (e.g. oferecer vantagens em troca de soluções prontas, doar trechos para o trabalho de outro grupo, etc.);
- Permitir que outros assumam sua identidade em atividades avaliativas (e.g. entregar trabalho que não fez ou permitir que outros façam provas por você);

- Plágio (i.e. aplicável a textos, programas de computador, etc.), o que envolve copiar porções significativas de textos ou programas de terceiros (de empresas, de pessoas ou de chatbots), sem atribuição de autoria ou, mesmo que haja atribuição de autoria, demonstre-se que não houve trabalho original significativo;
- Receber ou conceder ajuda em atividades avaliativas quando o contexto mostra que não é sensato receber tal ajuda.

Como consequências de violação do código de honra tem-se:

- Reprovação automática na disciplina, com conceito F, sem direito ao mecanismo de recuperação;
- Denúncia na Comissão Disciplinar Discente da Graduação (CDDG), a qual decidirá sobre a punição adequada à violação, o que pode levar a advertência, suspensão ou desligamento, de acordo com os arts. 78-82 do Regimento Geral da UFABC.

9 Cronograma de aulas

O cronograma a seguir pode variar de acordo com o aproveitamento aferido nas turmas durante o quadrimestre.

Aula #	Data	T/P	Tema
1	05/02	P	Revisão C
2	08/02	T	Revisão C
	12/02		Feriado
3	15/02	T	Tipos abstratos de dados
4	19/02	P	Tipos abstratos de dados
5	22/02	T	Estruturas lineares
6	26/02	P	Estruturas lineares
7	29/02	T	Estruturas lineares
8	04/03	P	Estruturas lineares
9	07/03	T	Análise de complexidade
10	11/03	P	Prova 1
11	14/03	T	Árvores binárias
12	18/03	P	Árvores binárias
13	21/03	T	Árvores binárias
14	25/03	P	Árvores binárias
15	28/03	T	Ordenação
16	01/04	P	Ordenação
17	04/04	T	Ordenação
	08/04		Feriado
18	11/04	T	Ordenação
19	15/04	P	Ordenação
20	18/04	T	Ordenação
21	22/04	P	Prova 2
22	25/04	T	Revisão de provas
23	30/04	P	Sub (reposição 12/02)
24	07/05	P	Rec (reposição 08/04)