

Prática Avançada de Programação A (MCZA038-17)

Cláudio Nogueira de Meneses

7 de fevereiro de 2025

Conteúdo

Formatação de entrada/saída; estruturas de dados (pilha, fila, lista, árvore, *heap*); ordenação eficiente (*quicksort*, *mergesort*); enumeração de permutações e subconjuntos; técnica de busca *backtracking* e busca (largura/profundidade) em grafos.

Créditos (T-P-E-I): 0-4-0-4

T - horas semanais de aulas teóricas presenciais;

P - horas semanais de trabalho de laboratório, aulas práticas ou aulas de exercícios, realizadas na universidade;

E - horas semanais de atividades extensionistas;

I - horas semanais adicionais de trabalho extraclasse necessárias para o bom aproveitamento da disciplina.

Recomendação: Algoritmos e Estruturas de Dados I

Detalhes sobre a disciplina

Código da turma: TDA1MCZA038-17SA;

Quadrimestre e ano: primeiro de 2025;

Horário das aulas: Segunda-feira das 10h às 12h e quinta-feira das 8h às 10h;

Local: laboratório 407-2 no campus de Santo André.

Planejamento da disciplina

Objetivos gerais: ensinar as estruturas de dados básicas, algoritmos de ordenação eficientes, algoritmos de enumeração e de geração de subconjuntos de um conjunto finito, técnica de *backtracking* e algoritmos básicos de buscas em grafos.

Objetivos específicos: fazer com que os alunos sejam capazes de resolver problemas utilizando os conceitos aprendidos.

Conteúdo programático:

SEMANA 1: Apresentação da ementa, forma de avaliação, definições de problemas que podem ser resolvidos utilizando os assuntos cobertos na disciplina, definições das operações básicas na estrutura de dados *pilha*, algoritmos, complexidades de tempo e espaço, implementações e aplicações.

SEMANA 2: Definições da estrutura de dados *fila* e *lista*, operações básicas, algoritmos, complexidades de tempo e espaço, implementações e aplicações.

SEMANA 3: Definições das estruturas de dados *árvore*, *árvore binária*, operações básicas, algoritmos, complexidades de tempo e espaço, implementações e aplicações.

SEMANA 4: Definição da estrutura de dados *heap*, operações básicas, algoritmos, complexidades de tempo e espaço, implementações e aplicações.

SEMANAS 5-7: Algoritmos *bucket sort*, *quicksort*, *heapsort* e *mergesort*, complexidades de tempo e espaço, implementações e aplicações.

SEMANAS 8-9: Enumeração de permutações e subconjuntos, algoritmos, complexidades de tempo e espaço, implementações e aplicações.

SEMANA 10: Buscas em largura e profundidade em grafos, algoritmos, complexidades de tempo e espaço, implementações e aplicações (p.ex., identificar componentes conexos em grafos).

SEMANAS 11-12: Técnica de *backtracking*, algoritmos, implementações e aplicações (p.ex., resolver problemas como: *eight queens puzzle*, *sudoku* e *0-1 knapsack*).

Estratégias didáticas

As aulas serão expositivas e com resolução de problemas. Além disso, será aplicada uma prova ao final de cada aula.

O material da disciplina estará disponível em um site do Moodle-UFABC, onde os alunos serão inscritos pelo professor ainda na primeira semana de aula.

Forma de avaliação

Serão aplicadas listas de exercícios semanalmente e provas curtas em todas as aulas. A média final antes da recuperação (MF) será calculada da seguinte maneira:

$$MF = 0.5 \times (\text{soma das 16 maiores notas das provas})/16 + 0.5 \times (\text{soma das 11 notas das listas de exercícios})/11 \quad (1)$$

e o conceito na disciplina será definido de acordo com as regras descritas na Tabela 1.

Conceito	Média final antes da recuperação (MF)
A	$9 \leq MF$
B	$7 \leq MF < 9$
C	$6 \leq MF < 7$
D	$5 \leq MF < 6$
F	$0 \leq MF < 5$

Tabela 1: Tabela de conversões de notas antes da recuperação.

Após determinar o conceito, de acordo com a Tabela 1, todos os(as) alunos(as) que ficarem com conceito D ou F poderão fazer uma prova de recuperação. Esta prova poderá conter problemas sobre quaisquer assuntos discutidos nas aulas e a média final após a recuperação (MFR) será calculada da seguinte maneira:

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

onde NR é a nota do(a) aluno(a) na prova de recuperação. O conceito obtido a partir da fórmula (1) será substituído por aquele obtido após a prova de recuperação. Este novo conceito será determinado usando as regras descritas na Tabela 2.

Conceito	Média final após a recuperação (MFR)
C	$6 \leq MFR$
D	$5 \leq MFR < 6$
F	$0 \leq MFR < 5$

Tabela 2: Tabela de conversões de notas após a prova de recuperação.

Bibliografia

- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. Algoritmos: teoria e prática. 2. ed. Rio de Janeiro, RJ: Campus, 2002.
- KNUTH D. E. The art of computer programming. Upper Saddle River, USA: Addison- Wesley, 2005.
- FEOFILOFF, P. Algoritmos em Linguagem C. Rio de Janeiro, RJ: Elsevier, 2009.

Bibliografia complementar

- AHO, A. V.; HOPCROFT, J. E.; ULLMAN, J. D. Data structures and algorithms. Reading, Mass: Addison-Wesley, 1983.
- DROZDEK, A. Estrutura de dados e algoritmos em C++. São Paulo, SP: Thomson Learning, 2002.
- RODRIGUES, P.; PEREIRA, P.; SOUSA, M. Programação em C++: conceitos básicos e algoritmos. Lisboa, PRT: FCA de Informática, 2000.
- SEDGEWICK, R. Algorithms in C++: parts 1-4: fundamentals, data structures, sorting, searching. Reading, USA: Addison-Wesley, 1998.
- TENENBAUM, A. M.; LANGSAM Y.; AUGENSTEIN M. J. Estruturas de dados usando C. São Paulo, SP: Pearson Makron Books, 1995.