

# Plano de desenvolvimento da disciplina MC202 Estruturas de Dados

Turmas G e H - 2º Semestre de 2019

Página da disciplina: <http://www.ic.unicamp.br/~rdahab/cursos/mc202/>

## Professor e auxiliares didáticos

- Professor: Ricardo Dahab - Sala IC-9, <http://www.ic.unicamp.br/~rdahab>, (19) 3521-5874, rdahab@ic.unicamp.br
- Auxiliares didáticos
  - Mayara Midori Omai e João Paulo da Silva (PEDs)
  - Victor Andrietta Razoli Silva e Natan Rodrigues de Oliveira (PADs)

## Locais e horários

- Aulas teóricas: Terças e quintas-feiras, 16-18h, salas CB-11 e CB-15, respectivamente.
- Aulas práticas: Quartas-feiras, 16-18h, salas SI-05 e SI-10.
- Horário de atendimento do professor: sala 28 do IC-1, em horário a ser divulgado.
- Horário de atendimento dos auxiliares didáticos: a ser divulgado

## Objetivos da disciplina

Esta disciplina visa enriquecer as técnicas de projeto de algoritmos introduzidas em MC102, explorando para isso duas vertentes: (i) a abstração dos tipos de dados, enfatizando as operações que devem ser suportadas pelos diferentes tipos; e (ii) a exemplificação de como tais abstrações podem ser materializadas com diferentes estruturas, resultando em algoritmos com diferentes desempenhos. A análise de desempenho será feita numa linguagem simples, já preparando o estudante para análise formal da complexidade de algoritmos em disciplinas mais avançadas.

## Programa

Transição de Python para C • Estruturas ligadas: nó, apontador, variável apontadora, alocação dinâmica de memória • Listas ligadas simples: operações básicas • Comparação de listas ligadas com vetores • Algoritmos gerais para listas simples: enumeração, inversão, cópia, concatenação • Pilhas, filas, e aplicações • Intercalação (merge) de listas e merge-sort; análise informal • Variações: listas circulares, duplamente ligadas, com cabeça. Lista livre • Algoritmos de ordenação • Árvores binárias: representação e percurso (recursivo) • Aplicação: árvores de busca (com inserção e remoção) • Árvores binárias de busca balanceadas • Fila de prioridade (heap) implementação com vetor e heapsort • Árvores B e generalizações • Introdução ao espalhamento (hashing): conceito, implementação com listas ligadas. • Grafos: conceito, representação por matrizes e listas ligadas • Percurso de grafos em largura e profundidade.

## Formato das aulas

As aulas teóricas seguirão o padrão usual, com uso de misto de transparências e explicações no quadro. As aulas práticas serão conduzidas nos laboratórios de ensino da UNICAMP, e serão baseadas na resolução de problemas de programação, em parte discutidos em aula, e em parte deixados como tarefas para serem resolvidas individualmente pelos alunos.

## Livro-texto

Não há um livro-texto que será seguido à risca, mas usaremos como referência bibliográfica principal o livro "Algorithms in C - Third Edition" de R. Sedgewick. Outras referências interessantes encontram-se relacionadas abaixo, na seção "Referências".

## Linguagens de programação

A linguagem de programação a ser utilizada nas tarefas práticas é a linguagem C.

## Avaliação

- Forma de avaliação
  - A avaliação se dará por meio de três provas teóricas e várias atividades de laboratório (labs), todas feitas individualmente. Os labs poderão ter pesos e prazos de entrega diferentes. Os prazos podem variar uma a duas semanas. Todos serão corrigidos usando o sistema SuSy de submissão e correção automática de programas.
- Critério de notas
  - Sejam P1, P2, P3 as notas das provas, todas entre 0 e 10, com pesos 3, 3 e 4, respectivamente. A média de provas, MP, será igual à média ponderada das notas das provas.
  - Sejam L1, L2, ..., as notas das atividades de laboratórios, todas entre 0 e 10 e seus respectivos pesos. A média de labs, ML, será igual à média ponderada das notas dos labs.
  - A média de aproveitamento, MA, será igual a  $(6MP + 4ML)/10$ .
  - Se  $MA \geq 5$ , o(a) aluno(a) estará aprovado(a), com média final  $MF = MA$ . Se  $MA < 2.5$ , o(a) aluno(a) estará reprovado(a) com média final  $MF = MA$ . Finalmente, caso  $2.5 \leq MA < 5$ , poderá fazer o exame e  $MF = \min(5.0, (MA+E)/2)$ , onde E é a nota obtida no exame.
- *Observações importantes:*
  - *A falta a uma das provas somente poderá ser justificada por motivo de saúde, com apresentação de atestado médico. Neste caso, o(a) aluno(a) poderá fazer o exame final em substituição à nota da prova à qual faltou; caso o(a) aluno(a) não obtenha média de aproveitamento (MA), e, portanto, necessite da nota do exame para aprovar-se, a nota da prova substituída será também usada como nota do exame final.*
  - *Qualquer tentativa de fraude nas provas ou nas atividades de laboratório resultará em média final  $MF = 0$  (zero) para todos os envolvidos.*

## Datas importantes

- Prova 1: 17/9
- Prova 2: 24/10
- Prova 3: 28/11
- Exame final: terça-feira, 10/12.

## Referências

1. R. Sedgewick, Algorithms in C, 3rd. edition, Addison-Wesley, 1990.
2. T. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. Algoritmos — Teoria e Prática. Campus, 2002.
3. A. V. Aho, J. E. Hopcroft, J. Ullmann. Data Structures and Algorithms. Addison-Wesley, 1983.
4. W. Celes, R. Cerqueira, J. L. Rangel. Introdução a Estruturas de Dados. Campus, 2004.
5. M. J. Folk e B. Zoellick. File Structures. Addison-Wesley, 1992.
6. F. Lorenzi, P. N. de Mattos, T. P. de Carvalho. Estruturas de Dados. Thomson, 2007.
7. S. L. Pereira. Estruturas de Dados Fundamentais. Érica, 1996.
8. E. M. Reingold e W. J. Hanson, Data Structures. Little-Brown (1983).
9. J. L. Szwarcfiter e L. Markenzon. Estruturas de Dados e Seus Algoritmos. Editora LTC (1994).

10. D. E. Knuth, The Art of Computer Programming, Vol I: Fundamental Algorithms. Addison-Wesley (1978).
11. N. Wirth, Algorithms + Data Structures = Programs. Prentice-Hall (1976).
12. A. M. Tenenbaum. Estruturas de Dados Usando C. Makron Books, 1995.
13. N. Ziviani. Projeto de Algoritmos. Thomson, 2004.