

# Plano de desenvolvimento da disciplina

## (Revisto em 20/3/2020 em razão da COVID-19)

### MC202 Estruturas de Dados

Turmas A, B e C - 1º Semestre de 2020

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

#### Preâmbulo

Este plano contém trechos do plano original e do plano revisto. Os trechos em verde correspondem a adições ao plano original. **Trechos críticos estão em vermelho.** As revisões se baseiam, principalmente, na resolução [GR 25/2020](#), da reitoria da Unicamp, que disciplina as atividades para os cursos de graduação e pós-graduação Unicamp diante da pandemia de coronavírus.

#### Professor

- Professor: Ricardo Dahab - Sala IC-9, <http://www.ic.unicamp.br/~rdahab>, (19) 3521-5874, rdahab@ic.unicamp.br

#### Locais e horários

- ~~Aulas teóricas: Terças e sextas feiras, 10-12h, salas PB-16 e PB-13, respectivamente.~~
- ~~Aulas práticas: Sextas feiras, 14-16h, salas SI-03, SI-08 e SI-10~~
- ~~Horário de atendimento do professor: a combinar por email.~~
- ~~Horário de atendimento dos auxiliares didáticos: a ser divulgado~~
- Aulas teóricas: Como se sabe, não haverá aulas teóricas presenciais. Veja abaixo, na seção [Formato das aulas](#), como serão substituídas essas aulas,
- Aulas práticas: Como se sabe, também não haverá aulas práticas presenciais. Veja abaixo, na seção [Formato das aulas](#), como serão substituídas essas aulas.
- Horário de atendimento do professor: em princípio o professor estará disponível por email. Adicionalmente, deverá disponibilizar horas online de atendimento, a serem divulgadas no início da semana de 23 de março.
- Horário de atendimento dos auxiliares didáticos será divulgado no início da semana de 23 de março.

#### 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.~~

Em vez de aulas presenciais, haverá indicação de leituras semanais e aulas de cursos à distância já testados por outros professores, e aprovados pelo professor desta disciplina. Serão publicados os nossos slides de acompanhamento, cuja publicação já estava prevista. O professor poderá, eventualmente, publicar vídeos com suas próprias aulas ou intervenções que julgar necessárias.

Quanto às aulas práticas, serão publicados, semanalmente, exercícios práticos para serem resolvidos, implementados e submetidos para avaliação, com prazo de entrega de uma semana. No horário anteriormente alocado para as aulas práticas, todos os monitores estarão disponíveis para tirarem dúvidas sobre esses exercícios. Adicionalmente, serão publicadas listas de exercícios teóricos para serem resolvidas pelos alunos, mas que não farão parte da avaliação; serão, sim, discutidas nas sessões de atendimento remoto dos monitores da disciplina, cujos horários serão divulgados no início da semana de 23 de março. O número total de horas semanais de atendimento não será inferior a 20.

Os meios de comunicação e postagem de material com os alunos serão: o endereço de email googlegroups já definido anteriormente e de conhecimento de todos; a página web da disciplina (<http://www.ic.unicamp.br/~rdahab/cursos/mc202/>) e o sistema SuSy para avaliação dos trabalhos práticos. Também pode vir a ser usado o Google Classroom.

## 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 e maioria durante os horários de aula nos labs. As tarefas de labs poderão ter pesos diferentes. Todos serão corrigidos usando o sistema SuSy de submissão e correção automática de programas.~~
  - A avaliação se dará por meio dos exercícios práticos, um por semana, a serem resolvidos e implementados nos computadores pessoais de cada estudante. Caso o aluno não tenha seu próprio equipamento, algum lab do IC será disponibilizado para um número limitado de estudantes. Hoje há uma diretiva da Unicamp nesse sentido, mas é possível que tudo mude, dependendo da evolução da pandemia.
  
- 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.~~
  - Sejam L1, L2, ... as notas das atividades práticas de programação, todas entre 0 e 10 e seus respectivos pesos, definidos no momento de publicação de cada atividade. A média de aproveitamento, MA, será igual à média ponderada dessas notas.
  - 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  $2.5 \leq MA < 5$ , o aluno poderá fazer um exame prático, de tempo limitado, supervisionado remotamente, 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: 14/4
- Prova 2: 22/5
- Prova 3: 30/6
- Exame final: terça-feira, 14/7.

As datas de todas as atividades são dependentes da progressão da pandemia e as resoluções da reitoria da Unicamp.

## Referências

1. R. Sedgwick, 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.

*Esta página é mantida pelo Prof. R. Dahab.*