

MC202 — ESTRUTURAS DE DADOS — TURMA DEFG
PROFESSOR: IAGO AUGUSTO DE CARVALHO
E-MAIL: iagoac@ic.unicamp.br
SITE: <http://iagoac.github.io/mc202.html>

PRIMEIRO SEMESTRE DE 2020

Atendimento

O horário de atendimento será prestado sempre depois das aulas pelo professor e em horários a serem divulgados no começo do semestre pelos monitores da disciplina.

Programa da Disciplina

• 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

Linguagens de Programação

A linguagem de programação *C* será utilizada com um compilador disponível no laboratório alocado para a disciplina.

Laboratórios

Haverá diversos laboratórios a serem entregues durante o semestre. Todos os laboratórios terão um prazo total de 7 dias ou mais para a primeira entrega e deverão ser feitos **individualmente**.

Para a correção dos laboratórios, será utilizado o sistema de submissão *SuSy* desenvolvido pelo professor Tomasz Kowaltowski. Os laboratórios serão submetidos pela página na Internet:

<https://susy.ic.unicamp.br:9999/mc202e>

Cada programa desenvolvido pelo aluno para um laboratório específico será automaticamente avaliado por este sistema em vários testes.

Os laboratórios serão corrigidos de duas possíveis formas: (i) a nota será proporcional ao número de casos de teste fechados resolvidos; ou (ii) em laboratórios mais simples, será dado a nota total caso a solução resolva todos os casos de teste fechados e 0 caso contrário. Por padrão, a correção seguirá a primeira forma e o enunciado deixará claro quando a correção ocorrer pela segunda forma.

A nota pode sofrer descontos de acordo com a qualidade do programa apresentado ou caso o programa submetido não satisfaça os critérios estabelecidos em seu enunciado. Desta forma, mesmo que o código seja capaz de resolver todos os casos de teste fechados, a nota final ainda pode ser menor do que a nota total.

Cada laboratório terá uma data para a primeira entrega da solução. Porém, até o final do semestre, os alunos poderão submeter laboratórios que não tenham sido submetidos dentro do prazo da primeira entrega ou resubmeter laboratórios já entregues com o objetivo de aumentar a nota obtida no laboratório. Porém, nesse caso a nota adicional obtida terá um desconto de 25% da nota. Exemplo:

- Um aluno que não entregou o laboratório no prazo inicial, ao tirar nota máxima no laboratório entregue até o final do semestre terá nota igual a $0,75 \times \text{nota total}$.

Cada laboratório será corrigido uma única vez durante o semestre. O aluno pode optar por entregar até o prazo estabelecido no enunciado ou ao fim do semestre, atentando-se ao desconto de 25% da nota.

Avaliação

- Cada laboratório terá um peso $LP_i \in \{1, 2, 3, 4, 5\}$. A média M , antes do exame, será a média ponderada dos m laboratórios.
- Caso o aluno tenha média $2,5 \leq M < 5,0$, ele poderá, opcionalmente, fazer um exame final.
- Seja E a nota do exame. A nota final, F , será calculada como:

$$F = \begin{cases} \min \left\{ 5,0; \frac{M + E}{2} \right\} & \text{caso } 2,5 \leq M < 5,0 \text{ e o aluno tenha realizado o exame,} \\ M & \text{caso contrário.} \end{cases}$$

A aprovação do aluno na disciplina está condicionada a dois requisitos:

- Obter nota final F maior ou igual a 5,0.
- Obter, pelo menos, 75% de presença no curso.

Caso qualquer um destes dois requisitos não seja alcançado, o aluno será considerado reprovado.

Data do Exame

- 14/07/2020

Observações

- Qualquer tentativa de fraude nos laboratórios implicará em nota final $F = 0$ (zero) para todos os envolvidos, sem prejuízo de outras sanções. Exemplos de fraudes são:
 - Compartilhar trechos de código de qualquer forma.
 - Utilizar trechos de códigos da internet ou de outras fontes.
 - Copiar ou comprar um laboratório.

Referências

O professor não seguirá um livro texto específico, entretanto, os livros abaixo cobrem o que será visto em aula. Em particular, as principais referências são os livros 1 e 2 da seguinte lista.

1. R. Sedgewick, Algorithms in C. 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.