

**MC202: Estruturas de Dados (turmas E e F)**

Segundo Semestre de 2011

Docente: *Tomasz Kowaltowski*Monitores: *Celina d'Ávila Samogin e Thiago Cavalcante***Descrição da Disciplina  
(versão provisória)****Horário das aulas**

Dia	Horário	Sala
Terça-feira	21 – 23	
Quinta-feira	19 – 21	
Sábado	10 – 12	CC02/CC03

**Atendimento** Será anunciado, oportunamente, um horário e local para atendimento de alunos. Será fixado também um horário de atendimento pelo monitor da disciplina. *Não haverá atendimento em véspera de provas!*

**Exercícios** Durante o desenvolvimento do curso poderão ser indicados vários exercícios e programas. Será exigida a entrega somente no caso das tarefas de laboratório, mas os conhecimentos adquiridos durante a resolução dos exercícios e a implementação dos programas serão cobrados nas provas.

**Avaliação** A nota de aproveitamento será calculada a partir de uma média ponderada  $P$  de notas de três provas teóricas ( $P_1$ ,  $P_2$  e  $P_3$ ) e de uma nota de laboratório  $L$ . A nota de provas será dada por:

$$P = (2P_1 + 3P_2 + 5P_3)/10$$

e a nota de laboratório pela fórmula:

$$L = \frac{\sum_{i=1}^n p_i t_i}{\sum_{i=1}^n p_i}$$

onde  $n$  é o número de tarefas de laboratório,  $p_i$  é o peso atribuído à tarefa ( $0 \leq p_i \leq 1$ ) e  $t_i$  ( $0 \leq t_i \leq 10$ ) é a nota obtida na tarefa. O aproveitamento final do semestre será calculado por:

$$A = \begin{cases} (6P + 4L)/10 & \text{se } P \geq 5 \text{ e } L \geq 5 \\ \min(P, L) & \text{caso contrário.} \end{cases}$$

A *média final* do curso  $F$  será calculada por:

$$F = \begin{cases} (A + E)/2 & \text{se o aluno fez o exame final} \\ A & \text{caso contrário} \end{cases}$$

onde  $E$  é a nota obtida no exame.

As provas e o exame final serão realizados nos dias indicados a seguir, nos horários correspondentes às aulas normais.

- $P_1$ :
- $P_2$ :
- $P_3$ :
- Exame Final:

#### Observações:

1. Será estabelecido um limite para o número de submissões de cada tarefa de laboratório. Este problema deve ser evitado através de testes adequados **antes** da submissão.
2. A submissão de uma tarefa de laboratório poderá ser considerada rejeitada ( $t_i = 0$ ) se não seguir **estritamente** as exigências do enunciado, mesmo que produza resultados corretos nos testes.
3. Qualquer tentativa de fraude nas provas ou nos laboratórios implicará em aproveitamento zero no semestre para todos os envolvidos, sem prejuízo de outras sanções.
4. As transgressões às regras de uso dos sistemas computacionais do Instituto de Computação ou outras unidades da UNICAMP que possam prejudicar outros usuários ou sistemas, dentro ou fora da Universidade, implicarão em aproveitamento zero no semestre para todos os envolvidos, sem prejuízo de outras sanções.
5. Não haverá provas substitutivas.
6. Todas as provas e o exame final serão realizados sem consulta.

#### Programa do Curso

Os itens apresentados a seguir indicam os tópicos a serem discutidos durante o semestre:

1. Introdução à análise de algoritmos
2. Estruturação elementar de dados: matrizes, registros, apontadores
3. Estruturas lineares: pilhas, filas, filas duplas
4. Recursão e retrocesso
5. Árvores binárias: representação, percursos
6. Árvores gerais: representação, percursos
7. Aplicação de árvores: busca, filas de prioridades, árvores AVL, árvores B, árvores rubro-negras
8. Listas generalizadas
9. Espalhamento
10. Processamento de cadeias de caracteres
11. Gerenciamento de memória
12. Algoritmos de ordenação
13. Algoritmos simples em grafos
14. Tipos abstratos e orientação a objetos

**Laboratórios** As tarefas de laboratório serão usadas para complementar e praticar os conceitos adquiridos em aulas teóricas. Normalmente, os enunciados dos problemas serão divulgados com antecedência de uma semana a dez dias antes do prazo de entrega. Haverá entre 10 a 12 tarefas de laboratório.

O desenvolvimento dos programas será feito em linguagem *C*, preferencialmente no ambiente *Linux*. A submissão das tarefas de laboratório será através do sistema automático de submissão e testes *SuSy*, utilizando um navegador como *Mozilla*, *Firefox*, *Chrome*, *Safari*, *Internet Explorer* etc.

As principais ferramentas a serem utilizadas sob o sistema *Linux* serão:

- `gcc`: *Gnu C Compiler* – um compilador para a linguagem *C*.
- `gdb`: *Gnu Debugger* – um depurador para várias linguagens.
- `emacs`: um editor de textos que pode ser utilizado para a preparação dos programas e outras tarefas.

Está disponível, na página da disciplina, a documentação eletrônica, que deve ser usada para aproveitar plenamente toda a potencialidade das ferramentas.

Caso os programas sejam desenvolvidos em outros ambientes que não *Linux*, eles deverão ser compatíveis com o compilador indicado acima que é usado pelo sistema automático de submissão. O comando de compilação normalmente utilizado pelo sistema é da forma:

```
gcc -std=c99 -pedantic -Wall -lm -o principal *.c
```

O endereço do portal para a submissão automática das tarefas é:

```
http://susy.ic.unicamp.br:9999/mc202ef.
```

Mais informações sobre a disciplina podem ser encontradas na página:

```
http://www.ic.unicamp.br/~tomasz/mc202.
```

Existem inúmeros textos relativos a estruturas de dados. As referências citadas são apenas algumas sugestões. Consulte a biblioteca e procure o texto que mais lhe convier. A referência [5] (apostila) bem como as transparências utilizadas em aula estarão disponíveis para cópia pelos alunos.

## Referências

- [1] A. V. Aho, J. E. Hopcroft, and J. Ullman. *Data Structures and Algorithms*. Addison-Wesley, 1983.
- [2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and Clifford Stein. *Introduction to Algorithms, second edition*. MIT Press, 2009.
- [3] A. Drozdek. *Estrutura de Dados e Algoritmos em C++*. Thomson, 2002.
- [4] J. L. Szwarcfiter e L. Markenzon. *Estruturas de Dados e seus Algoritmos*. LTC Editora, 1994.
- [5] C. L. Lucchesi e T. Kowaltowski. *Estruturas de Dados e Técnicas de Programação*. Instituto de Computação – UNICAMP, 2003.
- [6] P. Feofiloff. *Algoritmos em Linguagem C*. Elsevier Editora Ltda., 2009.
- [7] G. H. Gonnet. *Handbook of Algorithms and Data Structures*. Addison-Wesley, 1984.
- [8] E. Horowitz and S. Sahni. *Fundamentals of Data Structures in Pascal*. Computer Science Press, 1984.
- [9] D. E. Knuth. *The Art of Computer Programming*, volume I: Fundamental Algorithms. Addison-Wesley, 1978.
- [10] E. M. Reingold and W. J. Hanson. *Data Structures*. Little Brown and Company, 1983.
- [11] R. Sedgewick. *Algorithms in C*. Addison-Wesley, 1990.
- [12] D. F. Stubbs and N. W. Webre. *Data Structures with Abstract Data Types and Pascal*. Brooks/Cole, 1985.
- [13] A. M. Tenenbaum, Y. Langsam, and M. J. Augenstein. *Data Structures using C*. Prentice-Hall, 1990.
- [14] N. Wirth. *Algorithms + Data Structures = Programs*. Prentice-Hall, 1976.
- [15] N. Ziviani. *Projeto de Algoritmos (2a. ed.)*. Thomson, 2004.

*Processado em 27 de julho de 2011.*