MC202 - Estruturas de Dados
2º semestre de 2013
Prof. Guilherme Telles

Ementa Estruturas básicas para representação de informações: listas, árvores, grafos e suas generalizações. Algoritmos para construção, consulta, e manipulação de tais estruturas. Desenvolvimento, implementação e testes de programas usando tais estruturas em aplicações específicas.

Programa 1. Estruturas ligadas: nó, apontador, variável apontadora, alocação dinâmica de memória. 2. Listas ligadas simples: operações básicas. 3. Comparação de listas ligadas com vetores. 4. Algoritmos gerais para listas simples: enumeração, inversão, cópia, concatenação. 5. Pilhas, filas, e aplicações (inclusive eliminação de recursão). 6. Intercalação (merge) de listas e mergesort; análise informal. 7. Variações: listas circulares, duplamente ligadas, com cabeça. Lista livre. 8. Algoritmos de ordenação. 9. Árvores binárias: representação e percurso (recursivo). 10. Aplicação: árvores de busca (com inserção e remoção). 11. Fila de prioridade (heap) implementação com vetor e heapsort. 12. Árvores gerais: definição, representação por listas, percursos. 13. Listas generalizadas e uso para representar estruturas ligadas em geral. 14. Árvores B e generalizações. 15. Introdução ao espalhamento (hashing): conceito, implementação com listas ligadas. Técnicas de espalhamento para arquivos 16. Grafos: conceito, representação por matrizes e listas ligadas. 17. Percurso de grafos em largura e profundidade. 18. Implementação de estruturas de dados em disco

Avaliação A avaliação será composta de provas e trabalhos de programação, sendo

- n provas-unitárias ao longo do semestre. Cada prova-unitária consiste de uma única questão e será aplicada na primeira meia hora de uma aula teórica.
- uma prova com k questões. A prova será aplicada no horário de uma aula teórica.
- m trabalhos de programação.

A nota de provas P será calculada como

$$P = \frac{U_1 + \ldots + U_n + P_1}{n + k}$$

onde U_i é a nota da prova-unitária i no intervalo $[0, 10]$ e P_1 é a nota da prova no intervalo $[0, 10k]$.

A nota de trabalhos T será calculada como

$$T = \frac{T_1 + \ldots + T_m}{m}$$

onde T_j é a nota do trabalho de programação j no intervalo $[0, 10]$.

A nota do semestre S será calculada como

$$S = \begin{cases} \frac{2PT}{P+T} & \text{se } P + T > 0 \\ 0 & \text{caso contrário} \end{cases}$$

Alunos com $S < 2.5$ reprovam-se. Alunos com $S \geq 5$ aprovam-se. Alunos com $2.5 \leq S < 5$ e freqüência às aulas teóricas maior ou igual a 75% poderão fazer exame. A nota final do semestre F será calculada pela equação abaixo, onde E é a nota do exame. F será a nota lançada no sistema acadêmico.

$$F = \begin{cases} \min(\frac{S+E}{2}, 5) & \text{se o aluno fez exame} \\ S & \text{caso contrário} \end{cases}$$

Todas as provas e exames serão individuais e sem consulta. Todos os trabalhos serão individuais. Não haverá avaliações substitutivas. Qualquer tentativa de fraude nas provas, trabalhos ou no exame implicará em nota final do semestre (F) igual a zero para todos os envolvidos, sem prejuízo de outras sanções administrativas.

Exame O exame consistirá de duas provas contendo entre 4 e 6 questões cada. A nota E será a média aritmética das questões nas provas, cada uma no intervalo $[0, 10]$.

Datas As datas prováveis de provas estão listadas abaixo. A critério do professor as datas serão modificadas. Caso haja modificações nas datas, elas serão comunicadas aos alunos com pelo menos duas aulas de antecedência de alguma prova.

22 de agosto: prova-unitária.
5 de setembro: prova-unitária.
19 de setembro: prova-unitária.
1º de outubro: prova-unitária.
15 de outubro: prova-unitária.
7 de novembro: prova-unitária.
28 de novembro: prova.
10 e 12 de dezembro: exames.

Trabalhos Os trabalhos de programação serão divulgados no início de cada semana e recebidos através do sistema no endereço `http://bug.ic.unicamp.br/cgi-bin/sqtpm.pl`. O sistema fará a recepção e compilação dos programas e os executará com um conjunto de casos-de-teste. A pontuação obtida em cada trabalho é um limite superior para a nota. A nota final de cada trabalho será atribuída manualmente em função dos demais critérios de correção.

Atendimento O professor estará disponível para atendimento a dúvidas ao término das aulas teóricas.

Bibliografia

1. A. V. Aho, J. E. Hopcroft, J. Ullmann. Data Structures and Algorithms. Addison-Wesley, 1983.
2. W. Celes, R. Cerqueira, J. L. Rangel. Introdução a Estruturas de Dados. Campus, 2004.
3. T. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. Algoritmos - Teoria e Prática. Campus, 2002.
4. M. J. Folk e B. Zoellick. File Structures. Addison-Wesley, 1992.
5. F. Lorenzi, P. N. de Mattos, T. P. de Carvalho. Estruturas de Dados. Thomson, 2007.
6. S. L. Pereira. Estruturas de Dados Fundamentais. Érica, 1996.
7. E. M. Reingold e W. J. Hanson, Data Structures. Little-Brown, 1983.
8. R. Sedgewick, Algorithms in C. Addison-Wesley, 1990.
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. Tenembaum. Estruturas de Dados Usando C. Makron Books, 1995.
13. N. Ziviani. Projeto de Algoritmos. Thomson, 2004.