INSTITUTO DE COMPUTAÇÃO – UNICAMP

MC202 - Estruturas de Dados - Turmas D+E

1º Sem/2004

Prof. Jorge Stolfi

Teoria: 3as 21:00–23:00 e 5as 19:00–21:00, sala PB-01

Laboratório: sábados 10:00–12:00, sala LM-08.

Informações gerais

Atendimento: 3as 20:30-20:55 e 5as 21:05–21:30, no Instituto de Computação, sala 15. Outros horários, se necessário, podem ser combinados em classe, por e-mail, ou pelo telefone 3788-5884. **Atenção:** Não haverá atendimento nos dias de prova.

Listas de exercícios: No decorrer do curso serão fornecidas várias listas de exercícios escritos. A entrega não é obrigatória, mas sua solução é considerada matéria dada para fins de prova.

Laboratório: O curso incluirá $n \approx 10$ exercícios de programação, que devem ser resolvidos individualmente. Cada trabalho prático será considerado para nota apenas se for resolvido em boa parte durante a aula de laboratório correspondente, sob a supervisão do professor ou monitor. Será usada a linguagem de programação C.

Avaliação: A nota final será baseada em três provas escritas P_1, P_2, P_3, com pesos 2, 3, e 3, respectivamente; e nas n notas de exercícios de laboratório $T_1, T_2, \ldots, T_n$. A nota de aproveitamento, para alunos que não fizerem o exame final, será calculada por $A = MS$ onde

$$MS = \frac{MP + MT + 4\min\{MP, MT\}}{6}$$

$$MP = (2P_1 + 3P_2 + 3P_3)/8$$

$$MT = \max\{(\sum_i T_i)/n - 2f, 0\}$$

f = número de trabalhos práticos não entregues.

Note que, segundo estas fórmulas, ficarão para exame alunos que obtiverem nota menor que 4,0 em MP **ou** em MT, ou que faltarem em 3 ou mais aulas de laboratório.

Exame final: Não haverá provas substitutivas propriamente ditas. Porém, haverá um exame final opcional E, teórico e/ou prático, após o fim das aulas. Para alunos que fizerem o exame, a nota final será $A = (MS + E)/2$.

Datas das provas: As provas serão realizadas no horário normal das aulas teóricas, aproximadamente no início de setembro, em meados de outubro, e no fim de novembro. As datas exatas das provas serão estabelecidas no decorrer do curso e serão comunicadas com pelo menos uma semana útil (2 aulas) de antecedência. As provas poderão ser adiadas a qualquer momento, mesmo no dia marcado, valendo nesse caso também o intervalo mínimo de duas aulas entre o aviso de adiamento e a nova data.

As provas serão individuais, em classe, **sem** consulta. Cada prova cobrirá toda a matéria dada, até a aula anterior inclusive, bem como os exercícios escritos e de laboratório cujo prazo de entrega já tiver expirado.

Importante: Qualquer tentativa de fraude — nas provas ou nos trabalhos práticos, detectada na hora ou a posteriori — implicará na atribuição da nota zero *na disciplina*, sem direito a exame, *a todos os envolvidos*, sem prejuízo de outras sanções aplicáveis ao caso.

Bibliografia Não será seguido nenhum livro texto em particular. Os livros relacionados no fim deste documento são sugestões para estudo, que em conjunto cobrem todos os tópicos da ementa. Os livros de Cormen, Leiserson e Rivest [9] e Knuth [10], em particular, oferecem um tratamento mais aprofundado do assunto — incluindo material que será coberto nas disciplinas mais avançadas de algoritmos (MC438 e MC538).

Programa da Disciplina

1. Introdução.
   - Intuição sobre projeto e análise de algoritmos.
   - Notação $O(\cdot)$.
   - Exemplos simples (busca sequencial e binária, ordenação por inserção, mochila).
2. Estruturas sequenciais.
   - Representação interna de vetores: cálculo de endereços.
   - Representação de matrizes: linearização de índices.
   - Extração de linhas ou colunas.
3. Estruturas ligadas.
   - Conceitos básicos: nó, campo, apontador.
   - Propriedades: compartilhamento, localização relativa, manipulação rápida.
   - Variáveis apontadoras; passagem de argumentos.
   - Caminhando numa estrutura.
   - Alocação dinâmica de nós.
   - Construção e modificação de estruturas.
   - Apontadores em outras linguagens (C, Java, Fortran, etc.).
4. Listas.
   - Definição abstrata.

- Operações abstratas: indexação, vizinhos, busca, inserção, remoção.
- Representação sequencial e ligada: operações básicas.
- Comparação das duas representações.
- Alguns algoritmos úteis: enumeração, inversão, concatenação, cópia.
- Algumas aplicações (edição de texto, *mergesort*).
- Truques de programação: listas com descritores (cabeças), listas circulares.

5. Pilhas.
   - Definição abstrata.
   - Representações sequencial e ligada.
   - Exemplos de aplicações (paciência, eliminação de recursão).
6. Filas.
   - Definição abstrata.
   - Representação linear (circular) e ligada.
   - Exemplos de aplicações.
7. Listas duplamente ligadas.
   - Definição.
   - Operações básicas: inserção, remoção.
   - Algumas aplicações (simulação, ordenação topológica).
8. Listas multi-ligadas.
   - Definição.
   - Aplicação: listas invertidas, matrizes esparsas.
9. Árvores binárias.
   - Definição abstrata.
   - Ordens de percurso: em profundidade (ordem prefixa, posfixa, infixa, geral) e em largura.
   - Algoritmos abstratos de percurso usando recursão ou pilha (profundidade) e fila (largura).
   - Representação ligada.
   - Algoritmos de inserção e remoção de folhas.
   - Algumas aplicações (fórmulas algébricas, código de Huffman).

10. Filas de prioridade.

    - Definição abstrata.
    - Implementação sequencial simples; custo.
    - Implementação por árvore binária (*heap*).
    - Representação sequencial de *heap*.
    - Algumas aplicações (simulação discreta, *heapsort*).

11. Algoritmos e estruturas de busca.

    - Busca em lista sequencial (linear, binária); análise.
    - Truques: sentinelas, arranjo ótimo, rearranjo dinâmico.
    - Custo de inserção e remoção.
    - Árvore binária de busca; algoritmos de busca, inserção, remoção.
    - Árvores de busca balanceadas: conceito e análise.
    - Árvores B: definição e algoritmos básicos.
    - Tabelas e funções de espalhamento.

12. Grafos.

    - Definição abstrata; operações básicas.
    - Representação por matriz de adjacência.
    - Representação por vetor de arestas.
    - Representação ligada.
    - Percurso em largura e profundidade.
    - Truques: marcação e numeração de nós.
    - Algumas aplicações (caminho mínimo).

Bibliografia

[1] Nivio Ziviani: *Projeto de Algoritmos.* (*2ª edição*). Thomson, 2004.

[2] J. L. Szwarcfiter, L. Markenzon: *Estruturas de Dados e seus Algoritmos.*

[3] A. V. Aho, J. E. Hopcroft, J. D. Ullman: *Data Structures and Algorithms.*

[4] N. Wirth: *Algorithms + Data Structures = Programs.*

[5] A. Tannenbaum, M.J. Augenstein: *Data Structures Using Pascal.*

[6] G. M. Schneider: *An Introduction to Programming and Problem Solving with Pascal.*

[7] E. Horowitz, S. Sahni: *Fundamentals of Data Structures in Pascal.* (*Obs: evitem a edição em português - está cheia de erros.*)

[8] D. F. Stubbs, N. W. Webre: *Data Structures with Abstract Data Types and Pascal.*

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest: *Introduction to Algorithms.* (*Avançado.*)

[10] D. E. Knuth: *The Art of Computer Programming*, vol. I. (*Avançado.*)