PUC-Rio — Departamento de Informática
Período: 1999.2 — Horário: 2as e 4as feiras, de 14 às 16 horas
Prof. Celso Carneiro Ribeiro — Graduação

ANÁLISE DE ALGORITMOS

Objetivos:

O estudo de algoritmos é um tema central para todas as áreas da Ciência da Computação (algoritmos estão presentes, por exemplo, na pesquisa de dados em uma organização de arquivos, na classificação de dados segundo um determinado critério, na realização de consultas a bancos de dados, na execução das funções de um sistema operacional, no escalonamento de tarefas e programa em uma máquina paralela, na compilação de um programa, no tratamento de uma expressão aritmética, no cálculo de operações envolvendo matrizes e polinômios, na demonstração automática de teoremas, na exploração de alternativas e regras dentro de um sistema especialista, no roteamento de mensagens dentro de uma rede de processadores distribuídos, em protocolos de comunicação de dados, no cálculo de distâncias e interseções entre objetos em computação gráfica, no projeto e teste de circuitos integrados, etc.) e é relevante para grande parte da ciência e de suas aplicações tecnológicas. São os seguintes os principais objetivos deste curso:

1. Apresentar técnicas que permitem analisar e avaliar a eficiência computacional de algoritmos e comparar diferentes algoritmos que possam ser utilizados para a solução de um mesmo problema.
2. Estudar os princípios, métodos e técnicas fundamentais para o projeto de algoritmos corretos e eficientes.
3. Estudar a complexidade intrínseca de problemas e seu impacto sobre o desenvolvimento e a eficiência de algoritmos para sua solução.
4. Estudar problemas e algorimos freqüentemente utilizados em aplicações em áreas tais como otimização em grafos e redes, classificação e seleção, geometria computacional, operações com matrizes e polinômios, processamento de cadeias e alocação ótima de recursos.
5. Apresentar um introdução aos modelos de arquiteturas paralelas e aos algoritmos paralelos.

Descrição e ementa do curso:

1. **Introdução aos principais conceitos de algoritmos:** problemas, o "problema algorítmico", algoritmos e sua definição, solução de problemas, um pouco de história, critérios de análise, correção, eficiência, limitações.
2. **Análise de algoritmos:** desempenho de algoritmos e programas, tempo de processamento, notação O, complexidade de pior caso, análise de algoritmos, algoritmos polinomiais, comparação de algoritmos, indução, recursividade e algoritmos recursivos, comparação de implementações recursiva e iterativa de um mesmo algoritmo, algoritmos não-polinomiais, algoritmos pseudopolinomiais.

3. **Fundamentos de estruturas de dados:** complexidade de algoritmos para tratamento (criação, inserções, eliminações, atualizações) de listas, pilhas, filas, árvores, *heaps*, árvores binárias de busca, conjuntos, grafos.

4. **Técnicas de projeto de algoritmos:**

   (a) Algoritmos gulosos

   (b) Divisão e conquista

   (c) Programação dinâmica

   (d) Melhoria iterativa (busca local)

   (e) Caminhamento e busca em grafos

   (f) Aplicações: busca e seleção, ordenação, otimização em grafos e redes, processamento de cadeias, geometria computacional, álgebra, polinômios e matrizes.

5. **Fundamentos de Teoria da Complexidade:** problemas de decisão e de otimização, classes P e NP de problemas de decisão, problemas NP-completos, classe Co-NP, problemas NP-difíceis, classe P-space, hierarquia de complexidade.

6. **Aplicações:** caixeiro viajante, carteiro chinês, mochila, coloração de grafos, cobertura de nós, roteamento de veículos, escalonamento de tarefas, alocação de processadores, particionamento e recobrimento.

7. **Algoritmos paralelos:** modelos de arquiteturas paralelas, máquinas a memória compartilhada, redes de interconexão, algoritmos fundamentais: busca, ordenação, matrizes, grafos.

Bibliografia:

1. D. Harel, *Algorithmics: The Spirit of Computing*, Addison-Wesley, 1987, Wokingham.

2. T.H. Cormen, C.E. Leiserson e R.L. Rivest, *Introduction to Algorithms*, McGraw-Hill, 1990, New York.

3. G. Brassard e P. Bratley, *Algorithmics: Theory and Practice*, Prentice Hall, 1988, Englewood Cliffs.

4. U. Manber, *Introduction to Algorithms: A Creative Approach*, Addison-Wesley, 1989, Reading.

5. R. Sedgewick, *Algorithms*, Addison-Wesley, 1988, Reading.

6. M.A. Weiss, *Data Structures and Algorithm Analysis*, Benjamin Cummings, 1992, Redwood City.

7. S. Baase, *Computer Algorithms: Introduction to Design and Analysis*, Addison-Wesley, 1988, Reading.

8. A.V. Aho e J.D. Ullman, *Foundations of Computer Science, Vol. I*, W.H. Freeman, 1992, San Francisco.

9. A.V. Aho, J.E. Hopcroft e J.D. Ullman, *Data Structures and Algorithms*, Addison-Wesley, 1983, Reading.

10. T.C. Hu, *Combinatorial Algorithms*, Addison-Wesley, 1982, Reading.

11. C.H. Papadimitriou e K. Steiglitz, *Combinatorial Optimization*, Prentice-Hall, 1982, Englewood Cliffs.

12. R.L. Kruse, *Data Structures and Program Design*, Prentice-Hall, 1987, Englewood Cliffs.

13. S.S. Skiema, *The Algorithm Design Manual*, Springer-Verlag, 1998, New York.

14. J.A. McHugh, *Algorithmic Graph Theory*, Prentice-Hall, 1990, Englewood Cliffs.

15. R.E. Campello e N. Maculan, *Algoritmos e Heurísticas: Desenvolvimento e Avaliação de Performance*, Editora da UFF, 1994, Niterói.

16. R. Terada, *Desenvolvimento de Algoritmo e Estruturas de Dados*, McGraw-Hill, 1991, São Paulo.

17. N. Ziviani, *Projeto de Algoritmos*, Livraria Pioneira Editora, 1993, São Paulo.

18. M. Gondran e M. Minoux, *Graphs and Algorithms*, Wiley, 1984, Chichester.

19. A. Gibbons, *Algorithmic Graph Theory*, Cambridge University Press, 1985, Cambridge.

20. J.L. Szwarcfiter e L. Markenzon, *Estruturas de Dados e seus Algoritmos*, LTC, 1994, Rio de Janeiro.

21. J.L. Szwarcfiter, *Grafos e Algoritmos Computacionais*, Editora Campus, 1984, Rio de Janeiro.

22. S. Even, *Graph Algorithms*, Computer Science Press, 1979, Rockville.

23. F.P. Preparata e M.I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, 1985, New York.

24. K. Melhorn, *Data Structures and Algorithms 3: Multi-dimensional Searching and Computational Gemetry*, Springer-Verlag, 1984, Berlin.

25. M.R. Garey e D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, 1979, San Francisco.

26. D.H. Greene e D.E. Knuth, *Mathematics for the Analysis of Algorithms*, Birkhäuser, 1982, Boston.

27. J.-P. Barthélemy, G. Cohen e A. Lobstein, *Complexité Algorithmique*, Masson, 1992, Paris.

28. E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, e D.B. Shmoys (editores), *The Traveling Salesman Problem*, Wiley, 1985, Chichester.

29. M.M. Syslo, N. Deo, e J.S. Kowalik, *Discrete Optimization Algorithms*, Prentice-Hall, 1983, Englewood Cliffs.

30. N. Christofides, *Graph Theory: An Algorithmic Approach*, Academic Press, 1975, London.

31. C.C. Ribeiro, "Parallel Computer Models and Combinatorial Algorithms", in *Surveys in Combinatorial Optimization* (S. Martello, G. Laporte, M. Minoux and C.C. Ribeiro, eds.), *Annals of Discrete Mathematics* 31, p.325-364, North-Holland, 1987, Amsterdam.

32. J. JáJá, *An Introduction to Parallel Algorithms*, Addison-Wesley, 1992, Reading.

33. M.J. Quinn, *Parallel Computing: Theory and Practice*, McGraw-Hill, 1994, New York.

34. V. Kumar, A. Grama, A. Gupta e G. Karypis, *Introduction to Parallel Computing: Design and Analysis of Algorithms*, Bejamin/Cummings, 1994, Redwood City.

35. R.M. Karp e V. Ramachandran, "Parallel Algorithms for Shared-Memory Machines", in *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity* (J. Van Leeuwen, ed.), p.871-941, Elsevier, 1990, Amsterdam.

36. S.G. Akl, *The Design and Analysis of Parallel Algorithms*, Prentice-Hall, 1989, Englewood Cliffs.

37. A. Gibbons e W. Rytter, *Efficient Parallel Algorithms*, Cambridge University Press, 1988, Cambridge.

38. P. Chaudhuri, *Parallel Algorithms: Design and Analysis*, Prentice Hall, 1992, Sydney.

Elementos para avaliação final:

- Trabalhos computacionais (1-2), em grupo
- Seminário ou monografia sobre um artigo ou tópico do curso, em grupo
- Provas (2) individuais
- Testes individuais

Pré-requisitos:

- Desenvoltura em programação
- Estruturas de dados (vetores, matrizes, listas, pilhas, filas, árvores, grafos, conjuntos)
- Conhecimentos de matemática discreta