Complexidade (2006/07)

Armando B. Matos

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto

1 Objectivos

Na disciplina de Complexidade pretende-se que o aluno tome contacto com algumas das muitas facetas da complexidade, nomeadamente: (i) a complexidade de algoritmos dados, (ii) a questão da existência de algoritmos eficientes para problemas de decisão dados e (iii) a questão da existência de algoritmos para problemas de decisão dados.

2 Programa

1. **Elementos da análise de algoritmos**
   Eficiência e recursos utilizados pelos algoritmos. Funções de $\mathbb{N}$ em $\mathbb{R}^+$: caracterização do comportamento assintótico de através de ordens de grandeza. As ordens de grandeza $O()$, $\Omega()$ e $\Theta()$.
   Caracterização do comportamento dos algoritmos através de recorrências. Métodos de solução de recorrências.
   Minorantes de complexidade. Princípio da informação necessária. Aplicação a alguns problemas de pesquisa e de ordenação. Comparação com os algoritmos existentes mais eficientes.

2. **Fundamentos de computabilidade**
   Bijecções $\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$. Bijecções $\Sigma^\star \rightarrow \mathbb{N}$. Máquinas de Turing e modelos equivalentes. Tese de Church-Turing. Problemas decidíveis e semi-decidíveis. Funções definidas por recursão primitiva ("primitivas recursivas"). Funções recursivas parciais. Caracterização por uma linguagens de registos (`FOR` e `WHILE`). Linguagens recursivas e recursivamente enumeráveis. Funções recursivas (parciais) e recursivas totais. Um problema indecidível: o problema da auto-paragem. Reduções entre linguagens. Teorema de Rice. Outros problemas indecidíveis (questões relativas a gramáticas independentes de contexto, equações diofantina, etc.).

3. **Complexidade estrutural e teoria dos problemas NP-completos**
   Problemas de decisão e instâncias; codificação de instâncias. Reduções entre problemas de decisão; reduções polinomiais. Transitividade. As classes de problemas P e NP. Problemas completos duma classe. Existência de problemas completos em NP: teorema de Cook. Demonstrações de completitude em NP. Fora e dentro da Classe NP: as classes NP e co-NP. Problemas "NP-hard". A hierarquia polinomial. Espaço Polinomial e logarítmico. Classes de complexidade aleatorizadas. Problemas de optimização. Resolução prática de problemas "NP-hard".

4. **A criptografia sob o ponto de vista da complexidade**
   Criptografia de chave privada. Segurança do "one time pad". Criptografia de chave pública. O método RSA e o problema da factorização. Estudo de alguns protocolos criptográficos.

3 Carga horária e docentes

Carga horária:

Aulas teóricas ..... 2 × 1.5 h /semana

Aulas teórico-práticas ..... 2 × 1.5 h /semana

Docente:

Armando Matos (`acm@ncc.up.pt`)

3.1 Bibliografia

Análise de algoritmos

- Apontamentos disponíveis na página da disciplina.
- Brassard and Bratley, *Algorithmics, Theory and Practice*, Prentice-Hall International Editions, 1988:
  - Capítulo I: (generalidades)
  - Capítulo II, páginas: 37–43, 65–75.

Computabilidade e complexidade

- Apontamentos disponíveis na página da disciplina.
- John E. Hopcroft, *Introduction to Automata Theory, Languages and Computation*, Capítulo 8 (indecidibilidade).
- I. C. C. Phillips, *Recursion Theory*, em *Handbook of Logic in Computer Science (S. Abramsky D. M. Gabbay and T.S.E. Maibaum, eds.)*, vol. 1 (background: mathematical structures), Oxford University Press, 1993, pp. 79–187.
  Em especial páginas 122-136.

Teoria dos problemas completos em NP

- Apontamentos disponíveis na página da disciplina.
- M.R. Garey and D.S. Johnson, *Computers and Intractability, a Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, 1979.
  Em especial: capítulos 2 e 3 e secções 7.1 e 7.2.

Outras referências

- R. Sedgewick, P. Flajolet, *An Introduction to the Analysis of Algorithms*.
- George S. Lueker, *Some Techniques for Solving Recurrences*, ACM Computing Surveys, Volume 12 , Issue 4, 1980.
- Udi Manber, *Introduction to Algorithms : A Creative Approach*, Addison-Wesley, 1989.
- Bernard M. Moret, *The Theory of Computation*, Addison-Wesley, 1997.
  Nota. Inclui também a teoria dos problemas completos em NP.
- Piergiorgio Odifreddi, *Classical Recursion Theory*, North-Holland, 1999.

4 Avaliação

Avaliação Todos os alunos são admitidos a exame final. Haverá 3 provas de avaliação nas seguintes áreas: análise de algoritmos, computabilidade e classes de complexidade. O exame final é cotado para 20 valores. Sejam c_1, c_2 e c_3 as classificações obtidas nas provas.

1. Um aluno fica dispensado de exame final se

   $n \geq 10.0$, $c_1 \geq 7.5$, $c_2 \geq 7.5$ e $c_3 \geq 7.5$, onde $n = (c1 + c2 + c3)/3$.

   A nota de dispensa é n.

2. Um aluno dispensado de exame final com a classificação n que vá a exame e obtenha a nota e ficará com a nota final $\max\{n, e\}$.

Datas propostas para as provas

Prova	**Data**
Prova 1 - Análise de algoritmos	23 Março
Prova 2 - Computabilidade	27 Abril
Prova 3 - Classes de complexidade	25 Maio

5 Sumários das aulas teóricas do ano lectivo 2005/06

– Aula 1
Funcionamento da disciplina: programa, bibliografia, avaliação. Notas gerais sobre a eficiência dos algoritmos.

– Aula 2
Análise de algoritmos: recursos (tempo e espaço). Tempo de execução: pior caso, caso médio e melhor caso; exemplos. Ordens de grandezas das funções de $\mathbb{N}$ em $\mathbb{R}^+$. Ordens $O()$, $\Omega()$, $\Theta()$; relações e propriedades. Exercícios sobre ordens de grandezas.

– Aula 3
Continuação da resolução de exercícios sobre ordens de grandeza. Análise de algoritmos e funções definidas por recorrências. Exemplos de recorrências: ordenação por selecção do mínimo, "mergesort", sequência de Fibonacci e eficiência do algoritmo correspondente.

– Aula 4
Resolução de recorrências. Majoração de funções definidas por recorrências. Método "tabelar / suspeitar / demonstrar". Método da mudança de variável. Método das diferenças finitas.

– Aula 5
Método da equação característica homogénea. Aplicação à sucessão de Fibonacci. Equações características não homogéneas de determinados tipos. Exemplos. Exercícios.

– Aula 6
Minorantes de complexidade. Princípio da informação necessária. Aplicação a alguns problemas de pesquisa e de ordenação. Comparação com os algoritmos existentes.

– Aula 7
Exercícios de revisão sobre ordens de grandeza e recorrências.

– Aula 8
Funções recursivas totais. A classe das funções definida por "recursão primitiva". Caracterização indutiva e através de uma linguagem de registos.

– Aula 9
Exercícios de revisão variados.

– Aula 10
Nenhum modelo de computação caracteriza exactamente as funções recursivas totais; prova por diagonalização. A função de Ackermann.

– Aula 11
Modelos das funções recursivas parciais: linguagem `WHILE`, caracterização indutiva; operador de minimização. Linguagens recursivas e recursivamente enumeráveis; Teorema de Post.

– Aula 12
A tese de Church-Turing. Propriedades das linguagens recursivamente enumeráveis: listagem algoritmica, Teorema do domínio, Teorema do contra-domínio. redução ("muitos para um") entre linguagens e entre problemas de decisão. Breve referência ao Teorema de Rice.

– Aula 13
Demonstração do Teorema de Rice. Problema: dado $\mathtt{i}$, "$\{\mathtt{i}\}$ é uma função total?": indecidibilidade, não decidibilidade e não decidibilidade do problema complementar.
Exercícios diversos.

– Aula 14
Noção de linguagem completa de uma classe de computabilidade. Prova de que $\mathrm{L}_{\mathrm{PGP}}$ é completa na classe RE relativamente à reduçãp "$\leq$". Alguns complementos de computabilidade. Introdução à teoria das classes de complexidade.

– Aula 15
Exercícios variados de "computabilidade".

– Aula 16
redução polinomial $\propto$ entre problemas de decisão. Alguns problemas de decisão importantes: SAT, 3SAT, 2SAT, tautologia, ciclo euleriano (EC), ciclo hamiltoniano, Clique, cobertura por vértices (VC), PART, 3DM, 3COL, 2COL. Exemplos de reduções polinomiais: 3SAT $\propto$ SAT, SAT $\propto$ 3SAT, 3SAT $\propto$ VC, VC $\propto$ Clique.

– Aula 17
As classes de complexidade P, NP e co-NP. A caracterização da classe NP através de máquinas de Turing não determinística. Caracterização da classe NP através de predicados computáveis em tempo polinomial $\mathtt{pr}(\mathtt{x},\mathtt{y})$ onde $\mathtt{x}$ é a instância e $\mathtt{y}$ a solução.

– Aula 18
As classes co-NP e "co-NP completa". Caracterizações através de predicados. Exercícios relativos a reduções polinomiais.

– Aula 19
Existência de problemas completos na classe NP: Teorema de Cook.

– Aula 20
Transitividade da relação "redução polinomial". Outros problemas completos em NP: 3SAT, ciclo hamiltoniano, Clique, cobertura por vértices (VC), PART e 3DM.

– Aula 21
Ideia geral da demonstração do Teorema de Cook. Exemplos de demonstrações de que os seguintes problemas são completos em NP: "partição em ciclos simples disjuntos" e partição de um conjunto finito S em S1 e S2 de modo que qualquer conjunto de uma dada família de conjuntos tem pelo menos um elemento em S1 e outro em S2.

– Aula 22
Codificação das instâncias dos problemas de decisão. Noção de codificações polinomialmente relacionadas. Invariância de classes de complexidade (P, NPC...) relativamente às codificações polinomialmente relacionadas.

– Aula 23
A hierarquia polinomial: definição; alternância de quantificadores e classes da hierarquia. Análise da complexidade de alguns problemas aritméticos: soma, produto, potência modular e máximo divisor comum.

– Aula 24
Exercícios de revisão.