

INFORMAÇÕES SOBRE A DISCIPLINA

1 A disciplina

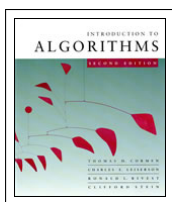
- Carga horária: 60 horas-aula
- Professor: Antonio Alfredo Ferreira Loureiro
Sala 4020, ICEx
E-mail: loureiro@dcc.ufmg.br
- Horário: Segundas e Quartas de 13:00 às 14:40
- Página da disciplina: www.dcc.ufmg.br/~loureiro/paa.html

2 Objetivos da disciplina

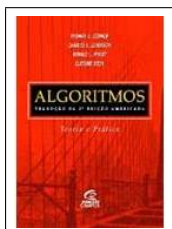
Apresentar ao aluno as técnicas diversas de projeto e análise de algoritmos, com ênfase em estruturas de dados e algoritmos. Dados dois ou mais algoritmos para uma mesma estrutura de dados, o aluno deverá ser capaz de compará-los quanto à complexidade de tempo e de espaço, utilizando para isso técnicas de análise de algoritmos. Ao final da disciplina o aluno deverá ser capaz de projetar e analisar algoritmos para lidar com problemas variados tais como pesquisa, ordenação e indexação, bem como decidir quais problemas admitem soluções computacionais práticas.

3 Livro Texto

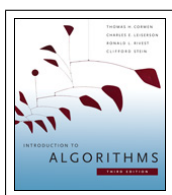
O livro texto recomendado é o “Introduction to Algorithms”, que atualmente está na terceira edição. Existe a tradução da segunda edição para o português. No entanto, outros livros, como os listados na bibliografia suplementar, cobrem o conteúdo visto nesta disciplina.



[Cormen et al., 2001]
Introduction to Algorithms, 2nd edition.
Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein.
2001, 1184 pp.
MIT Press, ISBN 0262531968.

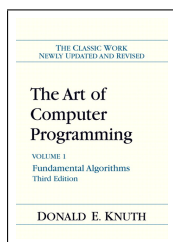


[Cormen et al., 2002]
Algoritmos: Teoria e Prática – Tradução da segunda edição em inglês.
Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein.
2002, 936 pp.
Editora Campus, ISBN 8535209263.

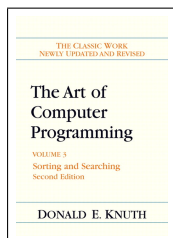


[Cormen et al., 2009]
Introduction to Algorithms, 3rd edition.
Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein.
2009, 1312 pp.
MIT Press, ISBN 9780262033848.
Para mais informações veja
<http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=11866>.

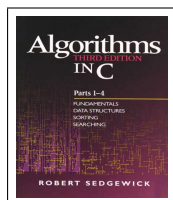
4 Bibliografia complementar



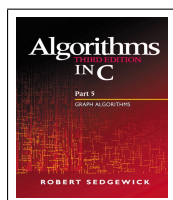
[Knuth, 1997]
The Art of Computer Programming, Volume 1: Fundamental Algorithms, 3rd edition.
Donald E. Knuth.
1997, 672 pp.
Addison-Wesley, ISBN 0201896834.



[Knuth, 1998]
The Art of Computer Programming, Volume 3: Sorting and Searching, 2nd edition.
Donald E. Knuth.
1998, 800 pp.
Addison-Wesley, ISBN 0201896850.



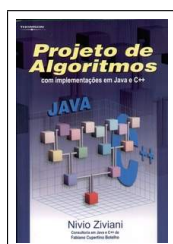
[Sedgewick, 1997]
Algorithms in C, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching, 3rd edition.
Robert Sedgewick.
1997, 720 pp.
Addison-Wesley, ISBN 0201314525.
O autor possui versões deste livro em C++ e Java.



[Sedgewick, 2001]
Algorithms in C, Part 5: Graph Algorithms, 3rd edition.
Robert Sedgewick.
2001, 512 pp.
Addison-Wesley, ISBN 0201316633.
O autor possui versões deste livro em C++ e Java.



[Ziviani, 2010]
Projeto de Algoritmos com Implementações em Pascal e C, 3ª edição.
Nivio Ziviani.
2010, 660 pp.
Editora Cenage, ISBN 8522110506.



[Ziviani, 2006]
Projeto de Algoritmos com Implementações em Java e C++.
Nivio Ziviani.
2006, 642 pp.
Editora Thomson, ISBN 8522105251.

5 Ementa

Modelos computacionais. Técnicas de análise de algoritmos. Paradigmas de projeto de algoritmos. Tópicos: Algoritmos em grafos; Algoritmos para casamento de padrões; Compressão de dados. Problemas NP-Completo.

6 Programa

A disciplina está dividida basicamente em quatro partes principais conforme descrito na ementa acima e seguirá o programa descrito a seguir.

6.1 Projeto e análise de algoritmos

1. Conceito de modelo computacional.
 2. Medidas de Tempo e Espaço de um algoritmo.
 3. Notações O , Ω , θ , o e ω .
 4. Técnicas de análise de algoritmos: somatórios, recorrências, árvores de decisão, limite inferior, oráculos, teorema Mestre.
 5. Exemplos de análise de algoritmos.
- Referência:
[Cormen et al., 2001] capítulos 1 a 4.
- Bibliografia suplementar:
[Knuth, 1997] capítulo 1.
[Sedgewick, 1997] capítulos 1 e 2.
[Ziviani, 2010] capítulo 1.

6.2 Paradigmas de projeto de algoritmos

1. Indução.
 2. Recursividade.
 3. Tentativa e erro.
 4. Divisão e conquista.
 5. Balanceamento.
 6. Programação dinâmica.
 7. Algoritmos gulosos.
 8. Algoritmos aproximados (heurísticas).
 9. Algoritmos paralelos.
- Referência:
[Cormen et al., 2001] capítulo 2, seção 2.3.1 (divisão e conquista); capítulo 15, seções 15.1 a 15.3 (programação dinâmica); capítulo 16, seções 16.1 e 16.2 (algoritmos gulosos); capítulo 35 (algoritmos aproximados).
- Bibliografia suplementar:
[Sedgewick, 1997] capítulo 5, seção 5.1 (recursividade), seção 5.2 (divisão e conquista), seção 5.3 (programação dinâmica).
[Ziviani, 2010] capítulo 2.

6.3 Tópicos

Algoritmos em grafos:

1. Conceitos básicos.
 2. Busca em profundidade.
 3. Busca em largura.
 4. Ordenação topológica.
 5. Componentes fortemente conectados.
 6. Árvore geradora mínima.
 7. Caminhos mais curtos.
- Referência:
[Cormen et al., 2001] capítulos 22 e 23.
- Bibliografia suplementar:
[Sedgewick, 2001] capítulos 17, 18, 20 e 21.
[Ziviani, 2010] capítulo 7.

Algoritmos para casamento de padrões:

1. Casamento exato.
 2. Casamento aproximado.
- Referência:
[Cormen et al., 2001] capítulo 32.
- Bibliografia suplementar:
[Ziviani, 2010] capítulo 8, seção 8.1.

Compressão de dados:

1. Compressão de textos em linguagem natural.
 2. Compressão de Huffman usando palavras.
 3. Codificação de Huffman usando bytes.
 4. Pesquisa em texto comprimido.
- Bibliografia suplementar:
[Ziviani, 2010] capítulo 8, seção 8.2.

6.4 Problemas NP-Completo

1. Classificação de problemas computacionais.
2. As classes $\mathcal{P}$, $\mathcal{NP}$ -difícil, $\mathcal{NP}$ e $\mathcal{NP}$ -completo.
3. Exemplos de problemas dessas classes.
4. Transformação polinomial.

7 Avaliação da Aprendizagem

4 Provas (4×10)	40
Trabalho prático de aquecimento	10
4 listas de exercícios (4×5)	20
Trabalho prático final	30
Total	100

Informações sobre a avaliação:

• Provas:

- Não haverá prova suplementar.
- As provas são individuais e sem consulta.
- A primeira prova irá tratar do conteúdo descrito na seção 6.1, a segunda na seção 6.2, a terceira na seção 6.3 e a quarta na seção 6.4.
- Datas previstas das provas:

PROVA	1ª	2ª	3ª	4ª
DATA	2/4	2/5	30/5	22/6

• Linguagem de programação:

- Qualquer exercício que envolva programação deve ser escrito na linguagem especificada no trabalho.
- O programa deve ser compilado e executado no ambiente de rede do DCC/UFMG.

• Exercícios:

- Todos os exercícios são individuais. Isto, evidentemente, não significa que seja proibido trocar idéias e discutir alternativas de solução. No entanto, cada aluno deve preparar a sua própria versão no momento de escrevê-la.

- Trabalho de aquecimento:
 - Veja a especificação na página da disciplina.
 - Data de entrega: 11/4/2011.
- Trabalho prático final:
 - Veja a especificação na página da disciplina.
 - Associado à cada TP deve haver um artigo técnico de até oito páginas sobre o tema escrito em português ou inglês no formato definido pela ACM. Para informações sobre a formatação veja a página <http://www.acm.org/sigs/pubs/proceed/template.html>.
 - Data de entrega: 4/7/2011.

8 Planejamento das aulas

As nossas aulas serão às segundas e quartas indicadas no calendário abaixo pelo dia em negrito e, entre colchetes, o número da aula. De acordo com o calendário oficial da UFMG, em alguns dias letivos não haverá aulas devido a feriados (F) e recesso (R). Todos os outros dias entre 10/3/2011 (início) e 9/7/2011 (encerramento), exceto domingos, são dias letivos (incluindo os sábados), quando poderão ser marcadas aulas e provas.

MARÇO 2011						
DOM	SEG	TER	QUA	QUI	SEX	SAB
		1	2	3	4	5
6	7	8	9	10 ^①	11	12
13	14 ^[1]	15	16 ^[2]	17	18	19 ^[3]
20	21 ^[4]	22	23 ^[5]	24	25	26
27	28 ^[6]	29	30 ^[7]	31		

MAIO 2011						
DOM	SEG	TER	QUA	QUI	SEX	SAB
1 ^[F]	2 ^[15]	3	4 ^[16]	5	6	7
8	9 ^[17]	10	11 ^[18]	12	13	14
15	16 ^[19]	17	18 ^[20]	19	20	21
22	23 ^[21]	24	25 ^[22]	26	27	28
29	30 ^[23]	31				

JULHO 2011						
DOM	SEG	TER	QUA	QUI	SEX	SAB
					1	2
3	4 ^[33]	5	6 ^[34]	7	8	9 ^②

ABRIL 2011						
DOM	SEG	TER	QUA	QUI	SEX	SAB
					1	2 ^[8]
3	4 ^[9]	5	6 ^[10]	7	8	9
10	11 ^[11]	12	13 ^[12]	14	15	16
17	18 ^[13]	19	20 ^[14]	21 ^[F]	22 ^[F]	23 ^[R]
24	25 ^[R]	26	27 ^[R]	28	29	30

JUNHO 2011						
DOM	SEG	TER	QUA	QUI	SEX	SAB
			1 ^[24]	2	3	4
5	6 ^[25]	7	8 ^[26]	9	10	11
12	13 ^[27]	14	15 ^[28]	16	17	18
19	20 ^[29]	21	22 ^[30]	23 ^[F]	24 ^[R]	25 ^[R]
26	27 ^[31]	28	29 ^[32]	30		

DIA	FERIADO
21/4	Tiradentes
22/4	Sexta-feira da Paixão
1/5	Dia do Trabalho
23/6	Corpus Christi

Observação: A Mostra das Profissões da UFMG irá ocorrer de 26 a 28 de abril (terça a quinta-feira). No ICEX, o recesso será de 25 a 29 de abril (segunda a sexta-feira), e não haverá aulas nem avaliações durante esse período nesta disciplina.

9 Comentários sobre a área de algoritmos

Os comentários abaixo aparecem na seguinte referência:

Breaking Intractability, by J.F. Traub and H. Wozniakowski, **Scientific American**, January 1994.

- ... Even though scientists have computers at their disposal, problems can have so many variables that no future increase in computer speed will make it possible to solve them in a reasonable amount of time.
- ... certain scientific questions can never be answered because the necessary computing resources do not exist in the universe.

- ...An impediment even more serious than intractability may occur: a problem may be unsolvable. A problem is unsolvable if one cannot compute even an approximation at finite cost.
- ...Because these results are intrinsic to the problem, one cannot get around them by inventing other methods.
- ...One possible way to break unsolvability and intractability is through randomization.
- ...Unfortunately, the proof of this result does not tell us what the points and algorithms are, thus leaving a beautiful challenge for the future.
- ...One important achievement of mathematics over the past 60 years is the idea that mathematical problems may be undecidable, noncomputable or intractable. Kurt Gödel proved the first of these results. He established that in a rich mathematical system, such as arithmetic, there are theorems that can never be proved.