

# MC658: Projeto e Análise de Algoritmos III

SEGUNDO SEMESTRE DE 2019.

PROF. FLÁVIO K. MIYAZAWA, [fk<sub>\[at\]</sub>ic<sub>\[dot\]</sub>unicamp<sub>\[dot\]</sub>br](mailto:fk@ic.unicamp.br)

## Programa da Disciplina

- CLASSES DE PROBLEMAS: Revisão das classes de Complexidade P, NP, NP-completo e NP-difícil e de reduções entre problemas.
- ALGORITMOS EXATOS: Algoritmos pseudo-polinomiais, Algoritmos de backtracking, Algoritmos de branch-and-bound, Programação Linear Inteira e Programação por Restrições.
- ALGORITMOS DE APROXIMAÇÃO: Aproximação absoluta, Fator de aproximação, Inaproximabilidade em aproximação absoluta, Fator de aproximação. Inaproximabilidade em fator de aproximação. Esquemas de aproximação polinomial. Uso de PL no desenvolvimento de algoritmos de aproximação.
- ALGORITMOS HEURÍSTICOS: Definições básicas. Algoritmos construtivos e algoritmos de busca local; Meta-heurísticas.

## Aulas e Atendimento

- As aulas serão nas segundas-feiras das 19:00 às 21:00 e nas quartas-feiras das 21:00 às 23:00, ambas na sala PB10.
- A disciplina contará com o apoio do PED Deyvison Nogueira, e seu horário de atendimento será divulgado posteriormente. Alunos que desejarem ter atendimento, devem enviar email para o PED com antecedência de pelo menos 24 horas confirmando o atendimento. O atendimento começará no início do horário estabelecido para o atendimento; não havendo outros alunos a serem atendidos, o horário de atendimento daquele dia será encerrado.
- O Atendimento do professor será nas quartas-feiras, das 18:00 às 19:00, na sala IC1-30. Alunos que desejarem ter atendimento, devem enviar email para o professor com antecedência de pelo menos 24 horas confirmando o atendimento. O atendimento começará no início do horário estabelecido para o atendimento; não havendo outros alunos a serem atendidos, o horário de atendimento daquele dia será encerrado. Não haverá atendimento na semana de uma prova ou exame.

## Avaliação

- Serão aplicadas duas provas com duração de uma hora e cinquenta minutos cada uma e  $n$  testes ou projetos de implementação (alguns laboratórios poderão ter tempo de implementação e peso maior, dependendo de sua complexidade). Apenas para aqueles alunos que não lograrem aproveitamento satisfatório no semestre será aplicado um exame final, também com duração de uma hora e cinquenta minutos. As datas das provas e do exame final estão indicadas na tabela abaixo.

Prova 1	Prova 2	Exame Final
07/10/2019	25/11/2019	09/12/2019

## Critério de avaliação

- A média das provas teóricas,  $NP$ , será computada da seguinte forma, onde  $P_j$  é a nota da prova  $j$ :  $NP = (3P_1 + 4P_2)/7$ .
- A média dos testes e projetos de laboratório,  $NL$ , será computada como  $NL = (p_1L_1 + \dots + p_nL_n)/(p_1 + \dots + p_n)$ , onde  $n$  é o número de atividades prático-teóricas avaliadas (planejamos ter  $n \geq 5$ ),  $L_j$  é a nota da atividade  $j$ ,  $p_j$  é o peso da atividade  $j$ . Tipicamente os pesos das atividades  $p_j$ 's serão iguais a 1, 2 ou 3, de acordo com sua complexidade. Cada atividade  $L_j$  pode ser um testinho em aula (avisado com pelo menos 1 semana de antecedência) ou um projeto de implementação em C ou C++ (com prazo de entrega de pelo menos uma semana). Dependendo da complexidade, uma atividade de implementação poderá ser individual ou feito em dupla.
- A média das atividades,  $MA$ , será computada da seguinte forma:  $MA = (7NP + 4NL)/11$ .
- A média do semestre,  $MS$ , será computada da seguinte forma:
  - Se  $[(MA \geq 5.0) \text{ e } (NP \geq 3.0) \text{ e } (NL \geq 3.0)]$  então  $MS = MA$
  - Caso contrário,  $MS = \min\{4.9, NP, NL\}$

- A média final,  $MF$ , será computada da seguinte forma:
  - Se  $[(MS < 2.5) \text{ ou } (MS \geq 5)]$  então  $MF = MS$  e o aluno não poderá prestar exame.
  - Caso contrário, o aluno deve fazer o exame, e sua média final será computada como  $MF = \min\{5.0, (MS+E)/2\}$ , onde  $E$  é a nota do exame.

## Observações

Não serão ministradas provas antecipadas nem substitutivas. Não será permitida qualquer tipo de consulta durante as provas ou exame.

**Aviso:** *Qualquer tentativa de cola ou fraude acarretará nota zero na disciplina para todos os implicados.*

## Exercícios

Serão indicados exercícios à medida que cada tópico for coberto. Além de servir para maior fixação do material apresentado em aula, questões de prova ou de exame podem ser extraídas diretamente ou baseadas nos exercícios. Os exercícios não serão recolhidos para correção. É importante que os alunos procurem resolver ao máximo os exercícios baseando-se no conteúdo visto em aula e na bibliografia sugerida. Posteriormente, sugere-se que os alunos apresentem suas resoluções nos atendimentos do PED e do professor.

## Bibliografia.

1. G. Brassard, P. Bratley, Fundamentals of Algorithmics, Prentice Hall, 1995.
2. T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, MIT Press, Third Edition, 2009.
3. S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani. Algorithms 1ed.. 2006. McGraw-Hill Education.
4. J. Erickson, Models of Computations - Lecture Notes, 2015,.
5. J. Kleinberg, E. Tardos, Algorithm Design, ADDISON WESLEY, 2005.
6. M.H. Carvalho, M.R. Cerioli, R. Dahab, P. Feofiloff, C.G. Fernandes, C.E. Ferreira, K.S. Guimarães, F.K. Miyazawa, J.C. Pina Jr., J. Soares, Y. Wakabayashi. Uma introdução sucinta a algoritmos de aproximação, 2001. 23o Colóquio Brasileiro de Matemática, IMPA, Rio de Janeiro. M.R. Cerioli, C.G. Fernandes, P. Feofiloff, F.K. Miyazawa (eds).
7. C. E. Ferreira, Y. Wakabayashi, Combinatória Poliédrica e Planos-de-Corte Faciais., livro para a X Escola de Computação, UNICAMP, julho de 1996.
8. U. Manber, Algorithms: A Creative Approach, Addison-Wesley, 1989.
9. F.K. Miyazawa, Programação Inteira, XI Escola Regional de Informática SBC - Paraná, pp. 49-90, Setembro, 2003.
10. F.K. Miyazawa e C.C. de Souza, Introdução à Otimização Combinatória, Jornadas de Atualização em Informática - Congresso da Sociedade Brasileira de Computação - JAI-SBC, 2015.
11. Z. Michalewicz, D. B. Fogel. How to Solve It: Modern Heuristics. Springer-Verlag, 2004.
12. I. Parberry <http://www.eng.unt.edu/ian/books/free/>. Problems on Algorithms.
13. C. H. Papadimitriou, K. Steiglitz. Combinatorial Optimization: Algorithms and Complexity, Dover, 1982.
14. R. Sedgewick, K. Wayne. Algorithms, 4th ed. Addison-Wesley, 2011.
15. V. Vazirani. Approximation Algorithms. 2001. Springer-Verlag.
16. D.P. Williamson and D.B. Shmoys. The Design of Approximation Algorithms. Cambridge University Press, 2011.
17. N. Ziviani, Projeto de Algoritmos, Thompson, segunda edição, 2004.