

INF01056 – Desafios de Programação – 2012/2 – Prof. João Comba

- Aulas: Sextas (8:30–11:50), Lab. 103 – Prédio Novo
- Contato: Sala 230, Prédio 43425, ramal 6930
- E-mail: comba “@” inf.ufrgs.br
- Créditos: 4 (60 horas)
- Pré-requisitos: INF01120 – Técnicas de Construção de Programas e INF05515 – Complexidade de Algoritmos B
- Departamento: Informática Aplicada
- Etapa Aconselhada no Curso: 7

Súmula

O objetivo deste curso é analisar e discutir algoritmos e técnicas de programação necessários para resolução de problemas desafiadores que aparecem em Maratonas de Programação. O curso é estruturado em uma sequência de aulas teóricas e práticas, onde tópicos são individualmente discutidos, seguindo a ordem: apresentação do problema, implementação da solução pelos alunos, verificação da solução com os robôs de correção, discussão das soluções e avaliação dos tópicos. Como objetivo final, o aluno é preparado para analisar um problema, determinar qual o melhor método de solução, quais as estruturas de dados adequadas e qual a implementação mais eficiente para a resolução correta do problema.

Calendário, Conteúdo e Material Didático das Aulas

Aula	Data	Conteúdo
01/02	31/08	Introdução. Objetivos da Disciplina. Ambiente de programação. Tipos de problemas. Programação: problemas básicos.
03/04		Estruturas de Dados: Moodle (Problemas distribuídos 31/08 – entrega até 16/09) – Estes problemas não receberão bônus de realização em aula
05/06	14/09	Grafos 1
07/08	21/09	Grafos 2
09/10	28/09	Programação Dinâmica 1
11/12	05/10	Maratona de Programação 1 (Semana Acadêmica)
	12/10	Feriado
13/14	19/10	Discussão Maratona de Programação 1
15/16	26/10	Programação Dinâmica 2
	02/11	Feriado
17/18	09/11	Matemática
19/20	16/11	Strings
21/22	23/11	Maratona de Programação 2
23/24	30/11	Discussão Maratona de Programação 2
25/26	07/12	Geometria
27/28	14/12	Tópicos Avançados
29/30	21/12	Maratona de Programação 3

OBS: Conteúdo das aulas é sujeito a mudanças devido ao andamento do semestre

Avaliação

A avaliação constará de notas de duas maratonas de programação e pelos problemas de programação distribuídos em aula. A nota final será calculada da seguinte forma:

$(MP1 * 1.5 + MP2 * 1.5 + MP3 * 1.5 + NTF * 1.5 + NTM * 2.0 + NTD * 2.0 + EXTRA * 1.0)$, onde:

- MP1: Nota da maratona de programação 1
- MP2: Nota da maratona de programação 2
- MP3: Nota da maratona de programação 3
- NTF: Notas dos trabalhos de nível fácil
- NTM: Notas dos trabalhos de nível médio
- NTD: Notas dos trabalhos de nível difícil
- EXTRA: Participação em competições como Topcoder e CodeForce serão avaliadas e contam pontos extra

O cálculo das notas dos trabalhos (NT) se dará da seguinte forma:

- Problemas podem ser submetidos em times de até 3 pessoas em AULA somente
- FORA DA AULA, problemas somente podem ser resolvidos individualmente.
- Em cada aula prática será distribuída uma lista de 3 problemas: 1 fácil, 1 médio, 1 difícil
- A solução dos problemas devem ser submetidos via Moodle SOMENTE QUANDO ACEITOS na UVA. A data limite para recebimento de soluções será definida no Moodle.
- Problemas resolvidos anteriormente por algum aluno não irão contar como resolvidos.
- Cada problema resolvido (ACEITO na UVA) durante a aula prática vale 1.5 ponto (informar instrutor durante a aula).
- Cada problema resolvido (ACEITO na UVA) que for entregue até o prazo estipulado vale 1 ponto.
- O cálculo das notas NTF, NTM e NTD respectivamente divide o número de pontos obtidos em cada categoria pelo número de problemas da categoria. Não haverá truncamento nesse cálculo.
- Entrega das soluções: arquivo fonte de cada solução (trabalhos em grupo precisam ser entregues por cada integrante do grupo)

A maratona de programação será realizada com times de até 3 integrantes. O cálculo da nota de cada maratona de programação irá considerar o número de problemas resolvidos e o tempo corrigido de cada time. **Os resultados das avaliações (maratonas) serão entregues ao final de cada maratona.**

Bibliografia

BIBLIOGRAFIA BÁSICA:

- **Programming Challenges: The Programming Contest Training Manual.** S. Skiena e M. Revilla. Springer-Verlag 2003.
- **Competitive Programming 2.** Stevem Halim e Felix Halim. 2011

BIBLIOGRAFIA COMPLEMENTAR:

- **Introduction to Algorithms.** Cormen et al . MIT Press 2001.
- **The Algorithm Design Manual.** Steven S. Skiena. Ed Telos 1997.
- **Hacker's Delight.** H. Warren Jr. Addison-Wesley 2003.
- **The Art and Craft of Problem Solving.** Paul Zeitz. John Wiley & Sons 1999

Links

- [Maratona de Programação Brasil](#)
- [Universidad de Valladolid Problems Sets and Online-Judge](#)
- [Programming Challenges website](#)
- [Course Notes from Skienna's Lectures](#)
- [Programs from Programming Challenges](#)
- [ACM Programming Challenges Training Web Site](#)
- [ACM International Collegiate Programming Contest \(ICPC\)](#)
- [ACM ICPC Problem Set Archive](#)
- [Standard Template Library Programmer's Guide](#)
- [RoqueWave STL Documentation](#)
- [C++ Documentation](#)

- [Dijkstra Applet](#)
- [Graph Applets](#)
- [Handbook of Integer Sequences](#)
- [Pick's Theorem Applet](#) (NEW)