MC521A/MC721A – Desafios de Programação I e III
Primeiro semestre de 2023

Professor responsável:
Fábio Luiz Usberti (fusberti@ic.unicamp.br) – sala 15 (IC1).

Monitores:
Luis Henrique Pauleti Mendes (PED) – luishpmendes@gmail.com

---

1 Página da Disciplina

Página da disciplina (Moodle):
https://moodle.ggte.unicamp.br/

Página de Maratona de Programação IC:
https://sites.google.com/view/maratonic/

2 Horário das Aulas

Dia	Horário	Sala
Sextas-feiras	14 – 18	Lab CC03 (IC3)

3 Ementa

Estruturas de dados. Busca por padrões. Ordenação. Combinatória. Teoria dos Números. Backtracking. Algoritmos em grafos. Programação dinâmica. Reticulados. Geometria computacional.

4 Programa

1. Introdução à programação competitiva.
2. Estruturas de dados e bibliotecas.
3. Busca exaustiva.
4. Divisão-e-conquista.
5. Algoritmos gulosos.
6. Programação dinâmica.
7. Grafos.
8. Matemática discreta.
9. Cadeias de caracteres.
10. Geometria computacional.

5 Critério de Avaliação

A avaliação da disciplina será composta por exercícios de programação. Em cada semana serão disponibilizados pelo menos 10 exercícios. Ao concluir um exercício, o aluno receberá uma pontuação, correspondente à data da submissão:

Submissão	Pontuação
Durante as primeiras 4 horas em que o exercício foi disponibilizado	2
Após às 4 horas em que o exercício foi disponibilizado	1
Não entregue	0

De acordo com as regras de maratona de programação[1], em cada placar os alunos serão classificados pelo número de problemas resolvidos e pelo tempo de submissão, incluindo penalizações por submissões incorretas. Seguindo essa classificação, **ao final das 4 horas** serão bonificados com uma pontuação adicional o primeiro, segundo e terceiro colocados (contanto que tenham resolvido **pelo menos um problema**) da seguinte forma:

Colocação	Pontuação extra
Primeiro colocado	3
Segundo colocado	2
Terceiro colocado	1

Pontos extras também podem ser adquiridos da seguinte forma:

- **(+2 pontos)** Submeter através do EA um relatório contendo uma descrição dos algoritmos para pelo menos dois problemas do último placar. O relatório, adotando o limite de uma página para cada problema, deve demonstrar a **corretude dos algoritmos**. A submissão pode ser feita até o final do semestre.

Cálculo da média final

Considere:

- X, a pontuação do aluno.
- MF, a média final do aluno.

A média final MF do aluno será calculada da seguinte forma:

$$MF = \frac{X}{12}$$

Situação do aluno

- O aluno com média final $MF \geqslant 5.0$ estará **aprovado**.
- O aluno com média final $MF < 5.0$ estará **reprovado por nota**.

[1] http://maratona.ime.usp.br/

6 Observações Importantes

- Esta disciplina **não possui exame**.
- O critério de avaliação requer que o aluno, para ser aprovado, tenha **pontuação maior ou igual a 60**. Isso corresponde a 30 problemas resolvidos em aula ou 60 problemas resolvidos ao longo do semestre ou alguma combinação desses.
- Um dos objetivos desta disciplina consiste em preparar o aluno para um bom desempenho nas maratonas de programação. Para isso, é essencial que o aluno seja **auto-didata**, consultando referências bibliográficas que permitam um maior aprofundamento dos assuntos tratados ao longo do curso.
- Será permitido que cada aluno produza um **repositório de algoritmos** para solução de problemas clássicos que pode ser consultado para a solução dos problemas apresentados na disciplina. Caso o algoritmo não seja de autoria do próprio aluno, torna-se necessário a citação da fonte. Caso o aluno utilize o repositório para a solução de um problema, essa informação deve estar explícita nos comentários do código-fonte da solução submetida.
- Qualquer tentativa de fraude, por exemplo cópia de soluções de outros alunos ou de terceiros, ou uso de algoritmos não documentados no repositório do próprio aluno, implicará em **média final zero** para todos os envolvidos, sem prejuízo de outras sanções previstas no regimento da universidade.

7 Atendimento

Para atendimento, envie uma mensagem por e-mail para os monitores ou para o professor.

8 Bibliografia

1. S. Halim e F. Halim. Competitive Programming 2, Second Edition Lulu (www.lulu.com), 2011. (IMECC – 005.1 H139c)
2. S. S. Skiena, M. A. Revilla. Programming Challenges: The Programming Contest Training Manual, Springer, 2003.
3. T.H. Cormen, C.E. Leiserson, R.L.Rivest e C. Stein. Introduction to Algorithms. 2nd Edition, McGraw-Hill, 2001. (no. chamada IMECC – 005.133 Ar64j 3.ed.)
4. U. Manber. Introduction to Algorithms: A Creative Approach. Addison-Wesley. 1989. (no. chamada IMECC – 005.133 Ec53t 2.ed.)

Tabela 1: Calendário da disciplina.

Mês	Dia	Evento	Obs.
Março	10, sexta-feira	Início das aulas	Apresentação da disciplina
Março	17, sexta-feira		Introdução à programação competitiva
Março	24, sexta-feira		Estruturas de dados avançadas
Março	31, sexta-feira		Busca completa
Abril	7, sexta-feira	Não haverá atividades	
Abril	14, sexta-feira		Divisão e conquista
Abril	21, sexta-feira	Não haverá atividades	
Abril	28, sexta-feira		Programação dinâmica 1
Maio	5, sexta-feira		Programação dinâmica 2
Maio	12, sexta-feira		Programação dinâmica 3
Maio	19, sexta-feira		Algoritmos gulosos
Maio	26, sexta-feira		Percursos em grafos
Junho	2, sexta-feira		Caminhos mínimos de um para todos
Junho	9, sexta-feira	Não haverá atividades	
Junho	16, sexta-feira		Caminhos mínimos de todos para todos
Junho	23, sexta-feira		Árvores geradoras mínimas
Junho	30, sexta-feira		Fluxos em Rede
Julho	7, sexta-feira	Liberação de Notas	