

MC521AB/MC721A – Desafios de Programação I e III

Primeiro semestre de 2020

Professor responsável:

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

Monitores:

Carlos V. D. Araújo (PED) – carlosvdaraujo@gmail.com

Erick Leonardo de Sousa Monteiro (PED - voluntário) – erick.san.monteiro@gmail.com

1 Página da Disciplina

Página da disciplina (Moodle):

<https://moodle.ggte.unicamp.br/>

Página de Maratona de Programação IC:

<http://www.ic.unicamp.br/~maratona/wiki/>

2 Ementa

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

3 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.

4 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 24h em que o exercício foi disponibilizado	2
Durante o mês em que o exercício foi disponibilizado	1
Não entregue	0

De acordo com as regras de maratona de programação¹, 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 24 horas** serão bonificados com uma pontuação adicional o primeiro, segundo e terceiro colocados (contanto que tenham resolvido **pele 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 (não triviais) do último placar. O relatório, adotando o limite de uma página para cada problema, deve demonstrar a **corretude dos algoritmos**. A submissão deve ser feita até as 23:59 da quinta-feira seguinte à aula em que o placar foi liberado.

Cálculo da média final

Considere:

- X_i , a pontuação do i -ésimo aluno.
- X_{\min} , pontuação mínima dentre os alunos com pelo menos 60 pontos.
- X_{\max} , pontuação máxima dentre os alunos com pelo menos 60 pontos.
- MF_i , a média final do i -ésimo aluno.
- MF_{\min} , média final mínima dentre os alunos com pelo menos 60 pontos.
- MF_{\max} , média final máxima dentre os alunos com pelo menos 60 pontos.

¹<http://maratona.ime.usp.br/>

A média final MF_i do i -ésimo aluno será calculada da seguinte forma:

Caso 1: $X_i < 60$

$$NB_i = 5.0 \times \frac{X_i}{60} \quad (1)$$

Caso 2: $X_i \geq 60$

$$MF_i = \begin{cases} MF_{\max} & \text{se } X_{\max} = X_{\min} \\ MF_{\min} + (MF_{\max} - MF_{\min}) \left(\frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \right) & \text{caso contrário} \end{cases} \quad (2)$$

A Equação (2) corresponde a uma interpolação linear das notas, conforme o segmento de reta representado (em vermelho) na Figura 1:

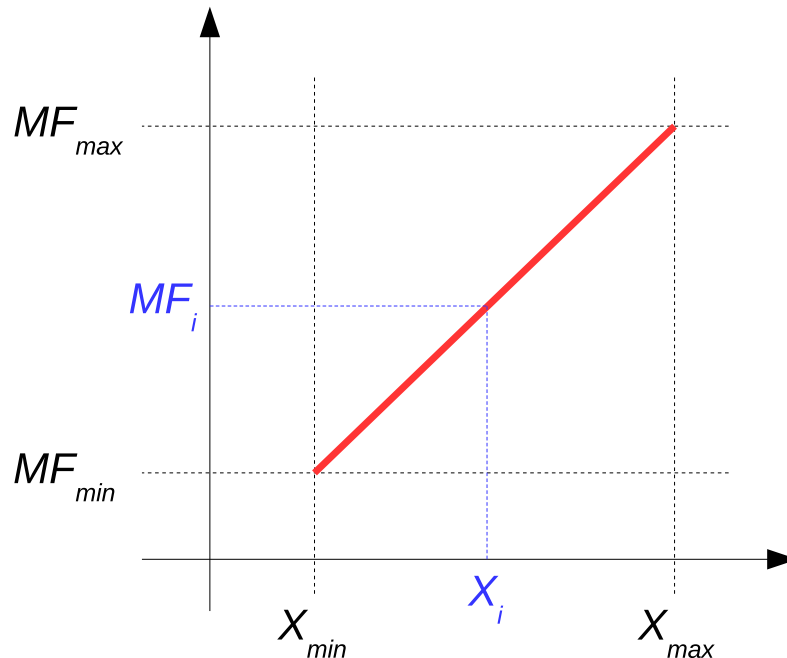


Figura 1: Distribuição linear das notas dos alunos com pelo menos 60 pontos.

As médias finais mínima MF_{\min} e máxima MF_{\max} são valores que dependem da menor e maior pontuações, respectivamente, obtidas pelos alunos com pelo menos 60 pontos, conforme determina a Tabela 1.

Situação do aluno

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

Tabela 1: Valores das médias finais mínima MF_{\min} e máxima MF_{\max} .

Pontuação mínima	MF_{\min}	Pontuação máxima	MF_{\max}
$60 \leq X_{\min} < 72$	5.0	$60 \leq X_{\max} < 72$	5.0
$72 \leq X_{\min} < 84$	6.0	$72 \leq X_{\max} < 84$	6.0
$84 \leq X_{\min} < 96$	7.0	$84 \leq X_{\max} < 96$	7.0
$96 \leq X_{\min} < 108$	8.0	$96 \leq X_{\max} < 108$	8.0
$108 \leq X_{\min} < 120$	9.0	$108 \leq X_{\max} < 120$	9.0
$X_{\min} \geq 120$	10.0	$X_{\max} \geq 120$	10.0

5 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.
- Como a correção dos exercícios é feita de modo automático, em servidores externos, **nenhuma pontuação atribuída a uma atividade será revista**. A única revisão possível será na contagem da frequência e deverá ser solicitada pelo aluno **até dia 03/07**.

6 Atendimento

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

7 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 2: Calendário da disciplina.

Mês	Dia	Evento	Obs.
Março	6, sexta-feira	Início das aulas	Apresentação da disciplina
Março	13, sexta-feira		Introdução à programação competitiva
Março	20, sexta-feira		Estruturas de dados avançadas
Março	27, sexta-feira		Busca completa
Abril	3, sexta-feira		Divisão e conquista
Abril	10, sexta-feira	Não haverá atividades	
Abril	17, sexta-feira		Programação dinâmica 1
Abril	24, sexta-feira		Programação dinâmica 2
Maio	1, sexta-feira	Não haverá atividades	
Maio	8, sexta-feira		Programação dinâmica 3
Maio	15, sexta-feira		Algoritmos gulosos
Maio	22, sexta-feira		Percursos em grafos
Maio	29, sexta-feira		Caminhos mínimos de um para todos
Junho	5, sexta-feira		Caminhos mínimos de todos para todos
Junho	12, sexta-feira	Não haverá atividades	
Junho	19, sexta-feira		Árvores geradoras mínimas
Junho	26, sexta-feira		Fluxos em rede