

MC621A/MC821A – Desafios de Programação II e IV

Segundo semestre de 2017

Professor responsável:

Diego de Freitas Aranha (dfaranha@ic.unicamp.br) – sala 28 (IC1)

Monitores:

Tiago Bruno Silva dos Reis (PED) – tiagob.reis@gmail.com

Darwin Danilo Saire Pilco (PED) – darwinsaireonline@gmail.com

1 Página da Disciplina

Página da disciplina:

<http://www.ic.unicamp.br/~dfaranha/cursos/desafios>

Página de Maratona de Programação IC:

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

2 Horário das Aulas

Dia	Horário	Sala
Sextas-feiras	14 – 18	(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 aula serão disponibilizados 10 exercícios. Ao concluir um exercício, o aluno receberá uma pontuação, correspondente à data da submissão:

Submissão	Pontuação
Durante a aula 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 da aula** 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 no portfólio do ensino aberto 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 **correção 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 nota bruta

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.
- NB_i , a nota bruta do i -ésimo aluno.
- NB_{\min} , nota bruta mínima dentre os alunos com pelo menos 60 pontos.
- NB_{\max} , nota bruta máxima dentre os alunos com pelo menos 60 pontos.
- $\delta = 10^{-6}$, constante infinitesimal para evitar divisão por zero.

¹<http://maratona.ime.usp.br/>

A nota bruta NB_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: $X_i \geq 60$

$$NB_i = NB_{\min} + (NB_{\max} - NB_{\min}) \left(\frac{X_i - X_{\min}}{X_{\max} - X_{\min} + \delta} \right) \quad (2)$$

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

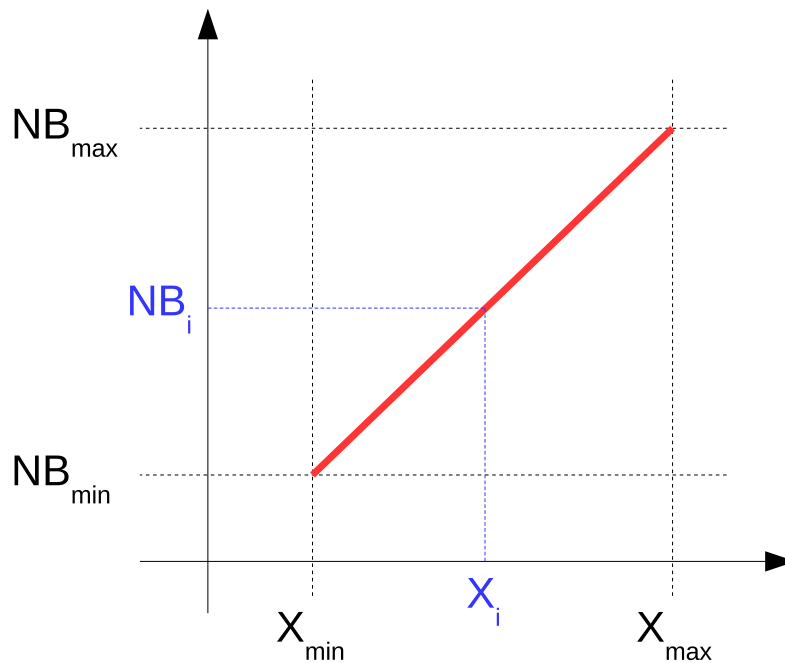


Figura 1: Distribuição linear das notas dos alunos com pelo menos 60 pontos.

As notas brutas mínima NB_{\min} e máxima NB_{\max} são valores que dependem da menor e maior pontuações, respectivamente, obtidas pelos alunos com pelo menos 60 pontos, conforme determina a Tabela 1. A média final MF_i do i -ésimo aluno corresponderá à sua nota bruta NB_i .

Situação do aluno

Se o aluno possuir frequência maior ou igual a 75% e média final $MF_i \geq 5.0$ ele estará **aprovado**. Caso contrário, ele estará **reprovado**.

Tabela 1: Valores das notas brutas mínima NB_{\min} e máxima NB_{\max} .

Pontuação mínima	NB_{\min}	Pontuação máxima	NB_{\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

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.
- A presença do aluno será verificada **duas vezes por aula**: no início (14:00) e fim (18:00) de cada aula. Desse modo, cada lista de presença comprova o comparecimento do aluno em duas horas-aula.
- Considerando que a presença será verificada 30 vezes ao longo do semestre, o aluno deverá assinar pelo menos 23 listas de presença para não ser **reprovado por frequência**.
- 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.
- Qualquer tentativa de fraude, por exemplo cópia de programas de outros alunos ou de terceiros, implicará em **média final igual a 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, por meio de programas computacionais, **nenhuma pontuação atribuída a uma atividade será revista**. A única revisão possível será na contagem da frequência e deverá ser solicitada pelo aluno **até o último dia de aula** (veja Tabela 2).
- Antes do início de cada aula, começando as 13:00, haverá a exposição de algum tópico de interesse da disciplina.
- Haverá treino por equipes na última sexta-feira de cada mês, em um total de 3 treinos. Para fins de avaliação, cada membro da equipe receberá metade dos pontos da equipe acumulados durante o treino.

7 Atendimento

Para atendimento extra-classe, envie uma mensagem por e-mail para o monitor ou professor.

Tabela 2: Calendário da disciplina.

Mês	Dia	Evento	Obs.
Agosto	04, sexta-feira	Aula 1	início das aulas
Agosto	11, sexta-feira	Aula 2	
Agosto	18, sexta-feira	Aula 3	
Agosto	25, sexta-feira	Aula 4	competição em equipes
Setembro	01, sexta-feira	Aula 5	
Setembro	08, sexta-feira		não haverá atividades
Setembro	15, sexta-feira	Aula 6	
Setembro	22, sexta-feira	Aula 7	
Setembro	29, sexta-feira	Aula 8	competição em equipes
Outubro	06, sexta-feira	Aula 9	
Outubro	13, sexta-feira		não haverá atividades
Outubro	20, sexta-feira	Aula 10	
Outubro	27, sexta-feira	Aula 11	competição em equipes
Novembro	03, sexta-feira		não haverá atividades
Novembro	10, sexta-feira	Aula 12	
Novembro	17, sexta-feira	Aula 13	
Novembro	24, sexta-feira	Aula 14	
Dezembro	01, sexta-feira	Aula 15	competição interna em equipes
Dezembro	07, quinta-feira		divulgação das notas

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.)