

CURSO:	Engenharias	SEMESTRE/ANO:	01/2023
DISCIPLINA:	Tópicos Especiais em Programação	CRÉDITOS:	04
CARGA HORÁRIA:	60 horas		
PROFESSOR:	Edson Alves da Costa Júnior		

PLANO DE ENSINO

1 Objetivos da Disciplina

A disciplina Tópicos Especiais em Programação tem como objetivo preparar os alunos do curso de Engenharia de Software da FGA para competições de programação, como a Maratona de Programação. Estes eventos ampliam o horizonte dos alunos e os estimulam a se aprofundarem nos tópicos de programação em geral. Além disso, a disciplina também constitui mais uma oportunidade para estudo e aprimoramento dos alunos em programação, tornando-os engenheiros mais preparados e capazes de atuar com competência no mercado de trabalho.

2 Ementa do Programa

I. *Introdução*

- i. Programação Competitiva
- ii. Maratonas de Programação
- iii. Juízes Eletrônicos
- iv. Dicas para estudo e treinamento
- v. Como começar

II. *Matemática*

- i. Introdução
- ii. Problemas Ad Hoc
- iii. Aritmética Estendida
- iv. Análise Combinatória
- v. Teoria dos Números
- vi. Teoria da Probabilidade
- vii. Detecção de Ciclos
- viii. Teoria dos Jogos

3 Horário das aulas e atendimento

AULAS: sábados, das 08:00 às 11:50 hrs.

ATENDIMENTO: segundas, das 14:00 às 15:50 hrs.

4 Metodologia

A metodologia consiste em aulas expositivas, com o auxílio do quadro branco e projetor digital. A fim de fortalecer a aprendizagem da disciplina, as aulas serão complementadas com exercícios e atividades, presenciais e extra-classe. As comunicações do curso serão feitas exclusivamente através da plataforma Moodle.

O curso também será focado na resolução de exercícios, envolvendo a análise e resolução de problemas oriundos de competições e de *online judges*. Ocasionalmente acontecerão contests ou na plataforma vJudge¹, ou na plataforma Codeforces², ou na plataforma AtCoder³.

5 Critérios de Avaliação

A avaliação do curso se dará por meio de uma única prova, em trios, cuja data está prevista no cronograma. Os trios serão determinado pelo *ranking* do curso. O aluno que atender os critérios descritos a seguir poderá melhorar a menção obtida na prova no *contest* Menção++.

5.1 Prova

A prova P será composta por 9 ou 10 problemas, a serem resolvidos em trio. É permitida a consulta a materias impressos e é vedada a consulta aos colegas de outros trios ou a recursos online.

A solução proposta para um problema será corrigida de acordo com os seguintes critérios: após ser compilada de forma bem sucedida, uma série de testes unitários automatizados alimentarão o programa resultante com entradas válidas e comparará os resultados obtidos com as saídas corretas. Uma solução será considerada aceita se obtiver sucesso em todos os testes unitários.

A menção final do curso será dada em função número N de problemas cujas soluções foram aceitas, de acordo com a tabela abaixo.

N	Menção	Descrição
0	SR	<i>Sem rendimento</i>
1	II	<i>Inferior</i>
2 ou 3	MI	<i>Médio inferior</i>
4 ou 5	MM	<i>Médio</i>
6 ou 7	MS	<i>Médio superior</i>
8 ou mais	SS	<i>Superior</i>

¹<https://vjudge.net>

²<http://codeforces.com>

³atcoder.jp

5.2 Critérios de aprovação

Obterá **aprovação** no curso o aluno que cumprir as **duas** exigências abaixo:

1. Ter presença em 75% ou mais das aulas;
2. Obter menção igual ou superior a MM.

5.3 Ranking do curso

Ao longo do curso será mantido um *ranking* dos alunos do curso, em ordem decrescente de *score*. O *score* S é composto por 4 variáveis: *ranking* no AtCoder (A), *ranking* do Codeforces (C), listas de exercícios (L), *contests* em sala de aula (T):

$$S = 100L + 200R + 300A + 400T$$

Em caso de empate, serão considerados os valores de T , A , R e L , nesta ordem. Persistindo o empate, será utilizada a ordem lexicográfica do nome completo do estudante.

Os trios serão determinados a partir do *ranking* na semana que antecede a prova: o primeiro trio será formado pelos três primeiros colocados, o segundo trio pelo quarto, quinto e sexto, e assim por diante.

5.3.1 Ranking do AtCoder

A nota A do estudante será dada em função do melhor *rating* A_{\max} obtido pelo aluno na plataforma AtCoder⁴ ao longo do semestre. O valor de A será dado por

$$A = \frac{\min(A_{\max}, 1500)}{1500},$$

se o estudante tiver participado de 3 ou mais *contests* ranqueados na plataforma ao longo do semestre, ou zero, caso contrário.

5.3.2 Ranking do Codeforces

A nota R do estudante será dada em função do melhor *rating* R_{\max} obtido pelo aluno na plataforma Codeforces⁵ ao longo do semestre. O valor de R será dado por

$$R = \frac{\min(R_{\max}, 2000)}{2000},$$

se o estudante tiver participado de 3 ou mais *contests* ranqueados na plataforma ao longo do semestre, ou zero, caso contrário.

⁴atcoder.com

⁵codeforces.com

5.3.3 Listas de exercícios

A cada aula será proposta uma lista de exercícios na plataforma vJudge. Estas listas terão duração de 15 dias e são individuais. Cada lista terá N_i exercícios a serem resolvidos e, após seu encerramento, as novas soluções não serão contabilizadas para o *ranking*.

É vedada a colaboração entre os estudantes da disciplina e a consulta a soluções de outras pessoas. A verificação de algum indício de fraude ou cola em qualquer um dos exercícios da lista resultará na exclusão imediata do aluno do *ranking*.

O valor de L é dado por

$$L = \frac{\min\left(\sum_i AC_i, 100\right)}{100},$$

onde AC_i é o número de exercícios com veredito AC na lista i .

5.3.4 Contests semanais

Na segunda parte de cada aula acontecerá um *contest* individual, a ser feitos na plataforma AtCoder ou na plataforma vJudge⁶. Cada *contest* terá N_i exercícios a serem resolvidos, com início às 9:00 hrs e término às 10:40 hrs.

É vedada a colaboração entre os estudantes e a consulta à soluções de outras pessoas ou materiais digitais. É permitida a consulta a material impresso. A verificação de algum indício de fraude ou cola em qualquer um dos *contests* resultará na exclusão imediata do aluno do *ranking* da disciplina.

O valor de C é dado por

$$C = \frac{1}{N} \sum_{i=1}^N \frac{D_i - P_i + 1}{D_i}$$

onde D_i é o número de estudantes que participaram do *contest* i e P_i é a posição final do estudante no *contest* i .

5.4 Contest Menção++

Após a prova P , os alunos que tiveram 75% ou mais de presença nas aulas e tiverem notas $L \geq 0.3$ e $A \geq 0.3$ terão uma oportunidade de melhorar sua menção por meio do *contest* Menção++.

Este *contest* será individual e seguirá o mesmo formato e condições da prova P , com a diferença que a menção só poderá ser incrementada uma única vez, conforme mostra a tabela abaixo.

⁶vjudge.net

Menção anterior	N	Nova menção
SR	1 ou mais	II
II	2 ou mais	MI
MI	4 ou mais	MM
MM	6 ou mais	MS
MS	8 ou mais	SS

6 Cronograma

Semana	Aula	Data	Conteúdo
01	1	01/04	<i>Apresentação do curso. Ambientação às ferramentas online</i>
02	-	08/04	Sexta-Feira da Paixão
03	2	15/04	<i>Lógica Booleana. Teoria dos Conjuntos</i>
04	-	22/04	Tiradentes
05	3	29/04	<i>Funções. Simulações.</i>
06	4	06/05	<i>Sequências e Séries. Malhas</i>
07	5	13/05	<i>Polinômios. Representação binária</i>
08	6	20/05	<i>Operações binárias. Exponenciais.</i>
09	7	27/05	<i>Logaritmos. Função logaritmo e função exponencial</i>
10	8	03/06	<i>Divisibilidade. Números primos</i>
11	9	10/06	<i>Funções multiplicativas. Teorema Fundamental da Aritmética</i>
12	10	17/06	<i>Aritmética Modular. Permutações</i>
13	11	24/06	<i>Arranjos. Combinações</i>
14	12	01/07	<i>Números de Fibonnaci e Catalan. Princípio da Inclusão/Exclusão</i>
15	13	08/07	<i>Funções Geradoras. Partições</i>
16	-	15/07	Prova
17	-	22/07	Contest Menção++

7 Bibliografia

LIVRO TEXTO

HALIM, Steven S. and **HALIM**, Felix. *Competitive Programming*, 1ª ed, Lulu, 2010. (Open Access)

CORMEN, Thomas H. **LEISERSON** and Charles E. and **RIVEST**, Ronald L. and **STEIN**, Clifford. *Algoritmos: Teoria e Prática*, Editora Campus, 2ª ed, 2002.

DROZDEK, Adam. *Estruturas de Dados e Algoritmos em C++*, Thomsom, 2001.

LITERATURA COMPLEMENTAR

KERNIGHAN, Brian and **RITCHIE**, Dennis M. *The C Programming Language*, Prentice Hall, 1988.

JOSUTTIS, Nicolai M. *The C++ Standard Library*, Addison-Wesley, 1999.

SOLTYS-KULINICZ, Michael. *Introduction to the Analysis of Algorithms*, World Scientific Publishing Co, 2012. (*eBrary*)

STEPHENS, Rod. *Essential Algorithms: A Practical Approach to Computer Algorithms*, John Wiley & Sons, 2013. (*eBrary*)

BALDWIN, Douglas; **SCRAGG**, Gregg. *Algorithms and Data Structures: The Science of Computing*, Charles River Media, 2004. (*eBrary*)