Universidade de Brasília
Departamento de Ciência da Computação

CIC0169 - Programação Competitiva - Turma 01

Plano de Ensino 2024/1

Atualizado em 18/03/2024

Prof. Dr. Vinícius Ruela Pereira Borges
viniciusrpb@unb.br
http://viniciusrpb.github.io/

1 Dados gerais

Pré-requisitos: CIC0004 - Algoritmos e Programação de Computadores

Carga horária: 2 horas teóricas, 2 horas práticas

Dia e hora: 2ª-feira e 4ª-feira, 16:00h - 17:50h

Salas: aulas teóricas no PJC BT 108; aulas práticas e mashups no LINF 03.

2 Objetivos

A disciplina Programação Competitiva tem como objetivos:

- introduzir as plataformas e ferramentas para programação competitiva;
- analisar e discutir algoritmos e técnicas de programação necessários para resolução de problemas desafiadores que aparecem em competições de programação;
- tomar decisões sobre o melhor método para a solução de um problema, quais as estruturas de dados adequadas e qual a implementação mais eficiente para a resolução correta do problema.

3 Ementa

Noções de complexidade de algoritmos. Fundamentos das linguagens de programação C e C++. Alocação dinâmica de memória. Operadores Bitwise e Bitmasks. Estruturas de dados lineares (lista, pilha, filas) e não-lineares (árvores binárias de busca). Soma de prefixos e Codificação Delta. Paradigmas de resolução de problemas: busca completa, divisão e conquista e algoritmos gulosos. Dois ponteiros. Introdução a programação dinâmica.

4 Ementa e Tópicos

A ementa da disciplina é dividida nos seguintes tópicos para organização da disciplina no modo remoto:

- Apresentação da disciplina;
- Noções de complexidade de algoritmos;
- Fundamentos das linguagens de programação C e C++;
- Estruturas de Dados:
  - Lineares: lista, filas e pilha;
  - Não-Lineares: árvores binárias de busca.
- Soma de prefixos e Codificação Delta;
- Operações Bitwise e Bitmasks;
- Paradigmas de resolução de problemas:
  - Busca completa (Busca em Largura, Busca em Profundidade, Backtracking, Busca Implícita em Grafos);
  - Divisão-e-conquista (Merge Sort, Busca Binária);
  - Algoritmos gulosos (Problemas clássicos, two Pointers);

5 Metodologia

A plataforma educacional Aprender3[1] será utilizada para apoiar a disciplina, isto é, para disponibilização do material didático, vídeos, informes, comunicados e notas. A comunicação entre o professor e os alunos ocorrerá **oficialmente** pelo Fórum de Avisos do Aprender3. Nesse sentido, o(a) aluno(a) possui total responsabilidade por verificar regularmente esse fórum.

Para acesso ao Aprender3/Moodle da turma, o(a) aluno(a) deve se cadastrar como usuário (basta preencher o formulário de cadastramento na página `http://aprender3.unb.br`), ou caso já esteja cadastrado, o(a) aluno(a) deve se inscrever[2] utilizando a chave de inscrição:

ProgComp_20241

As plataformas de programação competitiva oficial da disciplina serão o Codeforces e o CSES, em que serão disponibilizados os exercícios práticos, as listas de exercícios avaliativas, as avaliações e os contests. O Beecrowd Judge poderá ser utilizado para sugestões de problemas e como uma ferramenta secundária ao Codeforces.

Aulas

O curso consiste de aulas teóricas e práticas. As aulas teóricas abordam o conteúdo contemplado na ementa, em que o foco consiste em apresentar os fundamentos das técnicas de programação competitiva e resolução de problemas pelo professor. Já as aulas práticas ocorrem em laboratório e se baseiam em mashups (simulação de uma competição de programação) avaliativos.

[1] `https://aprender3.unb.br/course/view.php?id=22004`

[2] A página da disciplina no Aprender3 receberá novas inscrições a partir de 18/03/2024.

6 Sistema de Avaliação

As atividades avaliativas da disciplina "Programação Competitiva" compreendem os *mashups* (M), listas de exercícios (LE), participação em aula (P) e a participação em contests (Contests).

6.1 Mashups

Mashups são pequenas competições que serão realizadas semanalmente ou quinzenalmente no horário da aula na forma **presencial**. Cada *mashup* será constituído por 4 a 6 problemas, que podem pertencer a diversos tópicos relacionados com Programação Competitiva.

Para propósitos de avaliação, serão contabilizados apenas os exercícios resolvidos corretamente (Accepted) em cada *mashup* no tempo válido do mashup. A nota final do *mashup* M será calculada conforme a equação abaixo:

$$M = \frac{1}{K} \sum_{i=1}^{K} M_i, \tag{1}$$

em que M_i é a nota obtida do i-ésimo *mashup* dentre os K mashups realizados no decorrer do semestre. Cada mashup terá sua nota calculada com valores entre zero e 10, com base na quantidade de vereditos *Accepted* recebidos. Soluções corretas submetidas após o fim do mashup não serão aceitas para contabilização de nota.

6.2 Upsolving

Upsolving significa resolver as questões do mashup após seu encerramento. A nota do i-ésimo upsolving, denotada por U_i, é calculada com base na quantidade de vereditos *Accepted* recebidos no Mashup i após o período de upsolving. A nota final do Upsolving é calculada como:

$$Upsolving = \frac{1}{K-1} \sum_{i=1}^{K-1} U_i, \tag{2}$$

em que K é a quantidade de mashups realizados no decorrer do semestre. Apenas o último mashup do semestre não contará com período de upsolving. Vale ressaltar que $0 \leq U_i \leq 10$ e que o período de upsolving dura 15 dias após o encerramento do mashup i.

6.3 Listas de Exercícios

As listas de exercícios (LE) serão disponibilizadas no grupo da disciplina no site do Codeforces após a realização das aulas teóricas associadas aos temas da disciplina. Cada lista de exercícios ficará disponível por 15 dias após seu início na página do grupo no Codeforces.

Para propósitos de avaliação, serão contabilizados apenas os exercícios resolvidos corretamente (Accepted) em cada lista. Problemas cujas soluções forem submetidas após encerrado o período de 15 dias de disponibilização da lista não serão contabilizados na nota de cada lista de exercícios. A nota final da lista de exercícios LE será calculada conforme a equação abaixo:

$$LE = \frac{1}{N} \sum_{i=1}^{N} L_i, \tag{3}$$

em que N é a quantidade de listas de exercícios, sendo que cada lista L_i possui nota máxima 10.

IMPORTANTE: As listas de exercícios não serão reabertas após serem finalizadas em hipótese nenhuma.

6.4 Contests

O(a) aluno(a) deverá participar de 4 (quatro) competições (Contests) no site Codeforces ou AtCoder, ou em competições remotas organizadas pelos grupos MaratonasDF e UnBalloon, devendo-se resolver corretamente, ao menos, um problema dentre todas as questões. A nota Contests tem valor máximo de $10,0$ pontos, logo a participação em cada contest conforme os critérios estabelecidos contabiliza $5,0$ pontos cada. Diversos contests são regularmente divulgados nos sites mencionados e o(a) aluno(a) poderá escolher o momento mais apropriado para participar de alguns deles. Já para as maratonas de programação realizadas pelo grupo MaratonasDF, o(a) aluno(a) deverá enviar o certificado de participação para o professor para propósitos comprobatórios.

Participações virtuais ou aquelas realizadas após a finalização de um contest no Codeforces ou AtCoder não serão contabilizadas. Hackatons ou competições remotas/presenciais não constantes nesse plano de ensino também não serão contabilizadas. O prazo para realização dos contests é até **07/07/2024**.

6.5 Participação (P)

O objetivo da nota de participação é garantir que o(a) discente esteja envolvido(a) nos processos de aprendizado da disciplina, como também em todas as atividades durante o horário da aula. A nota de participação (P) é calculada como:

$$P = \frac{5,0 \times F + 5,0 \times Ex}{10,0}, \tag{4}$$

em que F é calculada de acordo com a frequência do discente em relação ao total de aulas ministradas. Ex é a resolução de exercícios durante as aulas práticas, em que o professor avalia o desempenho do discente nos exercícios práticos a serem disponibilizados. Os critérios são a evolução do discente no decorrer das aulas práticas e a resolução correta dos exercícios.

6.6 Menção final

Se $LE \geq 5,0$, $Contests \geq 5,0$, $P \geq 5,0$ e $Mashups \geq 5,0$, a média final (MF) será calculada considerando as avaliações a serem realizadas durante o semestre:

$$MF = \frac{3,5 \times Mashups + 1,0 \times Upsolving + 3,0 \times LE + 1,5 \times Contests + 1,0 \times P}{10,0}, \tag{5}$$

caso contrário, a média final (MF) será calculada como:

$$MF = min(Mashups, P, LE, Contests). \tag{6}$$

Conforme o regimento da Universidade de Brasília, a menção final do(a) aluno(a) será determinada associando-se NF de acordo com os critérios abaixo:

Menção final	MF
SS (Superior)	$9,0 \leq MF \leq 10,0$
MS (Médio Superior)	$7,0 \leq MF < 9,0$
MM (Médio)	$5,0 \leq MF < 7,0$
MI (Médio Inferior)	$3,0 \leq MF < 5,0$
II (Inferior)	$0 \leq MF < 3,0$

O(a) aluno(a) que não obtiver frequência mínima de 75% em relação ao número total de aulas estará reprovado(a) por faltas, recebendo menção final SR, independentemente do valor da Nota Final NF.

7 Frequência

As frequências dos discentes nas aulas serão contabilizadas por meio da assinatura da lista de presença, que será passada durante o horário da aula, após a chamada oral pelo nome de todos da turma.

IMPORTANTE: A assinatura da lista de presença é obrigatória para contabilizar a frequência. Uma falta poderá ser justificada para reposição de avaliação mediante atestado médico ou de trabalho devidamente assinado. Fonte: "GRADUAÇÃO UnB - Manual para estudantes" (página 35).

8 Ética acadêmica

Todos os códigos-fontes submetidos nas avaliações serão passarão por verificação de plágio utilizando-se um software de detecção de similaridade entre códigos-fontes. Por isso, o(a) aluno(a) que copiar ou plagiar código-fonte proveniente de repositórios da Internet ou de seus colegas, seja nas listas de exercícios, nos mashups ou nos Contests, será automaticamente reprovado(a) na disciplina por questões éticas. Nesse caso, o(a) aluno(a) receberá a menção final II.

9 Bibliografia

BÁSICA

- Halim S., Halim F., Competitive Programming 4: The New Lower Bound of Programming Contests, 2021.
- Laaksonen A., Competitive Programmer's Handbook, disponível online, 2018.
- Skiena S., Revilla M., Programming Challenges: The Programming Contest Training Manual, Springer-Verlag, 2003.

COMPLEMENTAR

- Cormen T., Algoritmos: Teoria e Prática. 3a ed., Elsevier Campus, Rio de Janeiro, 2012.
- Paul Zeitz, The Art and Craft of Problem Solving, John Wiley & Sons, 1999.