



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

PLANO DE ENSINO

1. Identificação

Disciplina: INE5451 - Tópicos Especiais em Algoritmos I

Nível: Graduação

Carga Horária: 72 horas-aula (Prática: 72)

Período Letivo: 2026-1

Turma: 08208 - CIÊNCIAS DA COMPUTAÇÃO (208)
Horário: 5 13:30 4 (ALOCAR)
Docentes:
Daniel Santana de Freitas (santana.d@ufsc.br)
Maicon Rafael Zatelli (maicon.zatelli@ufsc.br)

2. Ementa

-Ementa livre para assuntos relevantes na área de Algoritmos.

3. Objetivos

3.1 Objetivo Geral:

Resolver problemas computacionais avançados, semelhantes aos encontrados nas Maratonas de Programação realizadas anualmente pela Sociedade Brasileira de Computação (SBC) e International Collegiate Programming Contest (ICPC), especialmente com foco em finais sulamericanas e latino-americanas.

3.2 Objetivos Específicos:

- Escolher o algoritmo adequado à resolução de cada problema (de maratona) proposto
 - Justificar a escolha com base na correteude e na eficiência da solução encontrada
 - Implementar a solução (em C ou C++) e submeter a um juiz online
 - Discutir possíveis alternativas algorítmicas para o mesmo problema
-

4. Conteúdo Programático

- Problemas ad-hoc e de ordenação [8 horas-aula]
- Problemas que dependem de estruturas de dados [8 horas-aula]
- Problemas associados a paradigmas de resolução: Backtracking e Divisão e conquista [8 horas-aula]
- Problemas associados a paradigmas de resolução: Programação dinâmica e Algoritmos gulosos [8 horas-aula]

- 5 Problemas associados a processamento de strings [8 horas-aula]
 - 6 Problemas que envolvem diretamente a matemática [8 horas-aula]
 - 6.1 Sobre teoria de números
 - 6.2 Combinatoriais
 - 6.3 Com probabilidade
 - 7 Problemas que envolvem grafos: Busca em grafos [8 horas-aula]
 - 8 Problemas que envolvem grafos: Árvores geradoras de custo mínimo e Caminhos de custo mínimo [8 horas-aula]
 - 9 Problemas que envolvem Fluxo em grafos e problemas de Geometria Computacional [8 horas-aula]
-

5. Metodologia

As aulas serão práticas e estruturadas em simulados semanais, em que cada simulado consiste em uma lista de problemas avançados semelhantes aos encontrados em finais sulamericanas e latino-americanas da Maratona de Programação e International Collegiate Programming Contest (ICPC). Os simulados poderão ser resolvidos individualmente ou em grupo, a critério do professor.

Alguns problemas dos simulados poderão ser discutidos a fim de apresentar ou fixar algum conteúdo demandado para a resolução destes problemas. É esperado que cada estudante envolva-se ativamente na resolução dos problemas dos simulados e também participe das discussões sobre as soluções para os problemas.

6. Avaliação de Aprendizagem

Haverão até 6 simulados durante o semestre, com até 10 problemas cada. Cada problema dos simulados equivale a um ponto. Cada simulado terá um prazo de até 3 semanas para resolução, a depender do simulado.

Para validar a pontuação das questões resolvidas por cada aluno, a semana seguinte à entrega de cada simulado é reservada para apresentação das soluções feitas pelo aluno. Para a apresentação, será sorteado entre um a três problemas, dentre os problemas resolvidos pelo aluno, e ele deverá apresentar a solução desenvolvida para a turma, detalhando a ideia da solução e ilustrando com trechos de código. Após a apresentação será feita uma breve discussão sobre a solução apresentada e também questões poderão ser feitas. A pontuação da questão resolvida e apresentada será apenas atribuída em caso da apresentação ser satisfatória, ou seja, se o aluno apresentar a mesma de maneira adequada e conseguir discutir a solução com a turma. As questões resolvidas e não sorteadas para apresentação terão sua respectiva pontuação atribuída normalmente.

A média final (MF) será dada com base no percentual da pontuação total dos resolvidos (PR) descontando os problemas sorteados e apresentados de maneira insatisfatória, respeitando os pesos descritos acima, e conforme mapeamento a seguir:

- Sejam:

-- TP = total de problemas propostos no semestre

-- NR = número de problemas resolvidos descontando os problemas sorteados e apresentados de maneira insatisfatória

-- PR = $NR / TP * 100$ = percentagem dos problemas propostos resolvidos

- Então a média final será dada pelo menor inteiro maior ou igual a:
 - $MF = PR / 10$
-

7. Recuperação

Dado que a disciplina apresenta pelo menos 50% da carga horária consistindo de aulas práticas, conforme deliberação do Colegiado do Curso de Ciências da Computação de 18 de março de 2008, ela não prevê a realização de avaliação no final do semestre (recuperação) de que trata o parágrafo 2º do artigo 70 da Resolução 17/CUn/97.

8. Cronograma

A disciplina é inteiramente operacionalizada pela aplicação de simulados semanais. Um novo simulado será disponibilizado a cada semana, sendo que cada simulado consistirá em problemas que poderão envolver qualquer tópico descrito no programa de ensino, em qualquer ordem, objetivando assim construir um simulado que seja próximo ao que ocorre em finais sul-americanas e finais latino-americanas da Maratona de Programação e International Collegiate Programming Contest (ICPC). Discussões sobre problemas do simulado poderão ocorrer antes da realização do mesmo ou na semana seguinte à abertura do simulado.

9. Bibliografia Básica

- [1] Erickson, Jeff. Algorithms. Illinois: Independently published, 2019. ISBN: 978-1-792-64483-2.
 - [2] Feofiloff, P., Algoritmos em linguagem C, Elsevier, 2009.
 - [3] Halim, Steven; Halim, Felix. Competitive Programming: Increasing the Lower Bound of Programming Contests. Morrisville: Lulu Press. 2010.
 - [4] HALIM, Steven; HALIM, Felix. Competitive Programming 2: This increases the lower bound of programming contests. Again. 2011.
 - [5] Halim, Steven; Halim, Felix. Competitive Programming 3: The New Lower Bound of Programming Contests. Morrisville: Lulu Press. 2013.
 - [6] Mehlhorn K. Data Structure and Algorithms 1: Sorting and Search, Elsevier, 1984.
 - [7] Mehlhorn K., Sanders P. Algorithms and Data Structures. The Basic Toolbox, Elsevier, 2008.
-

10. Bibliografia Complementar

- [1] Cormen, T., Leiserson, C., Rivest, R., Stein, R., Introduction to Algorithms, 2nd ed., MIT Press, 2001.
- [2] Skiena, S., Revilla, M., ""Programming Challenges: The Programming Contest Training Manual", Springer-Verlag, New York, 2003.
- [3] Kleinberg, Jon; Tardos, Éva. Algorithm Design. Boston: Addison-Wesley Longman Publishing Co. 2005.
- [4] Mehlhorn K. Data Structure and Algorithms 2: Graph Algorithms and NP-Completeness, Elsevier, 1984.
- [5] Mehlhorn K. Data Structure and Algorithms 3: Multi-dimensional Searching and Computational Geometry, Elsevier, 1984.
- [6] Tutoriais do TopCoder: <https://www.topcoder.com/community/competitive-programming/tutorials>