



Plano de Ensino

1) Identificação

Disciplina: INE5452 - Tópicos Especiais em Algoritmos II
Turma(s): 08208
Carga horária: 72 horas-aula Teóricas: 0 Práticas: 72
Período: 1º semestre de 2020

2) Cursos

- Ciências da Computação (208)

3) Requisitos

- Ciências da Computação (208)
 - INE5403 - Fundamentos de Matemática Discreta para Computação
 - INE5408 - Estruturas de Dados
 - INE5403 - Fundamentos de Matemática Discreta para Computação
 - INE5408 - Estruturas de Dados

4) Professores

- Alvaro Junio Pereira Franco (alvaro.junio@ufsc.br)
- Maicon Rafael Zатели (maicon.zатели@ufsc.br)

5) Ementa

- Ementa Livre para assuntos relevantes na área de Algoritmos

6) Objetivos

Geral: Resolver problemas computacionais semelhantes aos encontrados nas Maratonas de Programação realizadas anualmente pela Sociedade Brasileira de Computação (SBC).

Específicos:

- Escolher o algoritmo adequado à resolução de cada problema (de maratona) proposto
- Justificar a escolha com base na correteza 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

7) Conteúdo Programático

- 7.1) Problemas ad-hoc e de ordenação [8 horas-aula]
- 7.2) Problemas que dependem de estruturas de dados [8 horas-aula]
- 7.3) Problemas associados a paradigmas de resolução: Backtracking e Divisão e conquista [8 horas-aula]
- 7.4) Problemas associados a paradigmas de resolução: Programação dinâmica e Algoritmos gulosos [8 horas-aula]
- 7.5) Problemas associados a processamento de strings [8 horas-aula]
- 7.6) Problemas que envolvem diretamente a matemática [8 horas-aula]
 - Sobre teoria de números
 - Combinatoriais
 - Com probabilidade
- 7.7) Problemas que envolvem grafos: Busca em grafos [8 horas-aula]
- 7.8) Problemas que envolvem grafos: Árvores geradoras de custo mínimo e Caminhos de custo mínimo [8 horas-aula]
- 7.9) Problemas que envolvem Fluxo em grafos e problemas de Geometria Computacional [8 horas-aula]

8) Metodologia

A presente disciplina possui um foco essencialmente prático. As aulas serão estruturadas em módulos semanais, conforme descrito no cronograma. Uma breve introdução sobre cada tópico e resolução de algum problema de exemplo será disponibilizada ao aluno por meio de uma videoaula. As videoaulas serão disponibilizadas no Youtube e seu link de acesso será compartilhado no Moodle. Além disso, material extra sobre cada conteúdo será disponibilizado em formato digital, tais como páginas de internet, apostilas, slides, entre outros que o professor considerar pertinente.

Visto que esta disciplina é essencialmente voltada em preparar os alunos para resolver problemas de programação competitiva, a maior parte da carga da disciplina compreenderá em o aluno resolver uma série de problemas computacionais semelhantes aos encontrados nas Maratonas de Programação realizadas anualmente pela Sociedade Brasileira de Computação (SBC). Tratando-se das características desta disciplina, é esperado que o aluno participante seja proativo e se esforce para construir soluções para os problemas propostos, além de pesquisar na internet outros conteúdos que o mesmo julgar necessário maior aprofundamento e/ou revisão para resolver alguns dos problemas propostos. É também característica da disciplina que os problemas propostos tenham diferentes dificuldades, o que também visa simular um cenário de competição real, assim, espera-se que alguns dos problemas demandem do aluno maior dedicação do que outros.

Listas com problemas serão divulgadas quase que semanalmente com o objetivo de fazer o aluno revisar e fixar o conteúdo. As soluções dadas pelos alunos para os problemas serão avaliadas por juízes online, que simulam os juízes da Maratona de Programação, tais como o URI Online Judge, UVa Online Judge, Codeforces, Sphere Online Judge, entre outros que o professor julgar pertinentes.

Além disso, detalham-se abaixo mais aspectos da metodologia que visam atender à resolução normativa Nº 140/2020/CUn:

Bibliografia online e ferramentas para prática da disciplina: as aulas são baseadas em material (em formato de slides, textos, apostilas, etc) produzido pelo próprio professor e disponibilizado no ambiente Moodle da disciplina. Além disso, bibliografia online adicional (links para páginas da internet, textos, apostilas, cursos, entre outros materiais) será disponibilizada no Moodle juntamente com os materiais criados pelo professor.

Plataformas de ensino a serem utilizadas: As videoaulas da disciplina serão disponibilizadas no Youtube ou na página do professor e seus links compartilhados no Moodle, as aulas estarão sempre disponíveis de maneira assíncrona. Encontros síncronos serão realizados no horário da aula e apenas com o objetivo de que os alunos possam sanar dúvidas a respeito do conteúdo, portanto, conteúdos extras/novos não serão dados por meio desses encontros síncronos. As ferramentas a serem utilizadas para os encontros síncronos serão o BigBlueButton ou Google Meet, cujo link da sala virtual estará disponível no Moodle. Aos alunos que tiverem dúvidas e não puderem eventualmente participar dos encontros síncronos, os mesmos poderão encaminhar dúvidas via e-mail ou fórum da disciplina no Moodle.

Registro de frequência: para aferir a presença dos alunos na disciplina será utilizada a ferramenta de conclusão de atividades do Moodle. O próprio aluno deverá marcar os slides e vídeos visitados e o aluno que tiver 75% ou mais dos itens marcados terá considerada frequência suficiente. Não será cobrada frequência nos encontros síncronos, embora seja recomendada a participação do aluno na medida do possível.

9) Avaliação

A avaliação da disciplina será composta pelos resultados de 10 simulados durante o semestre, cada um com 4 a 13 problemas. Os simulados serão realizadas de maneira assíncrona, sendo o simulado disponibilizado no Moodle ao início do horário da disciplina no dia da semana em que a disciplina ocorrer. O tempo estimado para a resolução do simulado será de aproximadamente 300 minutos (mesmo tempo de duração que a Maratona de Programação), porém, o prazo para a entrega da resolução de cada simulado pelo aluno poderá ser de até 168 horas (ou uma semana) a partir do seu início, visando já contemplar alunos que eventualmente tiverem problemas de conexão ou outros imprevistos. Durante este período de 168 horas, será garantido o tempo do dia em que a disciplina ocorrer durante a semana, assim, o aluno poderá também utilizá-lo para a resolução do simulado. Durante o simulado, o professor estará disponível na Sala Virtual no horário previsto da disciplina. A fim de melhor avaliar o aprendizado do aluno, o professor poderá também requerer uma entrevista com o aluno posteriormente à data de entrega de cada simulado. A entrevista será de maneira síncrona e em horário a combinar com cada aluno, dando preferência que ocorra no horário da disciplina. Na entrevista, o aluno deverá fazer uma breve apresentação sobre as soluções ou tentativas de soluções dadas para certos problemas do simulado e depois será feita uma breve discussão.

A média final (MF) será dada com base no percentual de problemas resolvidos dos simulados (PR), conforme mapeamento a seguir:

PR \geq 95% = MF 10.0

95% > PR \geq 85% = MF 9.5

85% > PR \geq 80% = MF 9.0

80% > PR \geq 75% = MF 8.5

75% > PR \geq 70% = MF 8.0

70% > PR \geq 65% = MF 7.5

65% > PR \geq 60% = MF 7.0

60% > PR \geq 55% = MF 6.5

55% > PR >= 50% = MF 6.0
 50% > PR >= 45% = MF 5.0
 45% > PR >= 40% = MF 4.5
 40% > PR >= 35% = MF 4.0
 35% > PR >= 30% = MF 3.5
 30% > PR >= 25% = MF 3.0
 25% > PR >= 20% = MF 2.5
 20% > PR >= 15% = MF 2.0
 15% > PR >= 10% = MF 1.5
 10% > PR >= 5% = MF 1.0
 5% > PR > 0% = MF 0.5
 PR == 0% = MF 0.0

Conforme parágrafo 2º do artigo 70 da Resolução 17/CUn/97, o aluno com frequência suficiente (FS) e média final no período (**MF**) entre 3,0 e 5,5 terá direito a uma nova avaliação ao final do semestre (**REC**), sendo a nota final (**NF**) calculada conforme parágrafo 3º do artigo 71 desta resolução, ou seja: $NF = (MF + REC) / 2$.

10) Cronograma

SEMANAS #1 e #2 - Introdução à programação competitiva - Exemplos de problemas, entrada e saída de dados - Problemas ad-hoc e de ordenação

SEMANAS #3 e #4 - Problemas que dependem de estruturas de dados

SEMANAS #5 e #6 - Problemas associados a paradigmas de resolução: Backtracking e Divisão e conquista

SEMANAS #7 e #8 - Problemas associados a paradigmas de resolução: Programação dinâmica e Algoritmos gulosos

SEMANAS #9 - Problemas associados a processamento de string

SEMANAS #10 - Problemas que envolvem diretamente a matemática

SEMANAS #11 e #12 - Problemas que envolvem grafos: Conceitos básicos, busca, ordenação topológica,

SEMANAS #13 e #14 - Problemas que envolvem grafos: Árvores geradoras de custo mínimo, caminhos de custo mínimo, fluxo máximo, emparelhamento

SEMANAS #15 e #16 - Problemas que envolvem técnicas e estruturas avançadas (Fenwick tree, segment tree, sqrt decomposition, entre outras) - Problemas de geometria computacional

Os simulados ocorrerão quase que semanalmente, a partir da segunda semana de aula e outras listas de problemas serão disponibilizadas a partir da primeira semana de aula para que os alunos possam praticar cada conteúdo. Ao longo do semestre, os materiais referentes a cada aula (vídeo, slides e exercícios) serão abertos no Moodle seguindo o calendário a ser disponibilizado também no Moodle.

11) Bibliografia Básica

- Erickson, Jeff. Algorithms. Illinois: Independently published, 2019. ISBN: 978-1-792-64483-2.
- Feofiloff, P., "Algoritmos em linguagem C", Elsevier, 2009.
- Halim, Steven; Halim, Felix. Competitive Programming: Increasing the Lower Bound of Programming Contests. Morrisville: Lulu Press. 2010.
- Halim, Steven; Halim, Felix. Competitive Programming 3: The New Lower Bound of Programming Contests. Morrisville: Lulu Press. 2013.
- Mehlhorn K. Data Structure and Algorithms 1: Sorting and Search, Elsevier, 1984.
- Mehlhorn K., Sanders P. Algorithms and Data Structures. The Basic Toolbox, Elsevier, 2008.

12) Bibliografia Complementar

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