



Ficha 2 (variável)

Disciplina: Tópicos em Algoritmos						Código: CI1355/CI355	
Natureza: () Obrigatória (X) Optativa				(X) Semestral () Anual () Modular			
Pré-requisito:		Co-requisito:		Modalidade: () Presencial (X) Totalmente EAD () % EAD ¹			
CH Total: 60 CH semanal: 6	Padrão(PD): PD=10	Laboratório(LB): LB=0	Campo(CP): CP=0	Estágio(ES): ES=0	Orientada(OR): OR=0	Prática Específica(PE): PE=50	
	Estágio de Formação Pedagógica(EFP): EFP=0	Extensão(EX): EX=0	Prática como Componente Curricular(PCC): PCC=0				
EMENTA (Unidade Didática)							
Tópicos diversos de Projeto e Análise de Algoritmos							
PROGRAMA (itens de cada unidade didática)							
Introdução a algoritmos utilizados em programação competitiva.							
1. Entrada/saída em C++, estruturas de dados e algoritmos da STL							
2. Busca completa, teste de permutações e backtracking.							
3. Princípios matemáticos de sequências, funções recursivas e inclusão e exclusão.							
4. Divisão e conquista, busca binária, árvore de Fenwick e de segmentos.							
5. Soluções gananciosas e programação dinâmica.							
6. Busca em grafos, caminho mínimo e menor ancestral comum.							
7. Dois ponteiros, emparelhamento estável, compressão de coordenadas, padrões comuns.							
8. Teoria dos números: Crivo de Eratóstenes, inverso multiplicativo, propriedades do módulo.							
9. Articulações e pontes, componentes conexos, emparelhamento bipartido e fluxo máximo/corte mínimo.							
10. Decomposição em raiz quadrada, cadeias (HLD) e algoritmo de Mo.							
11. Strings: Trie, KMP, expressões regulares, vetor e autômato de sufixos.							
12. Jogos combinatórios, Nim e Teorema de Sprague-Grundy.							
13. Geometria: Representação de pontos e polígonos, varredura linear, fecho convexo.							
14. Outros temas mais avançados dependendo do tempo disponível.							



OBJETIVO GERAL

Apresentar problemas de programação competitiva e técnicas para soluções.

OBJETIVO ESPECÍFICO

1. Expor diversos problemas de programação competitiva que exigem diversas técnicas diferentes para resolução.
2. Apresentar em detalhe certos algoritmos e estruturas de dados e apontar para referências para aprender mais sobre técnicas específicas.

PROCEDIMENTOS DIDÁTICOS

O curso mescla: (1) aulas teóricas expositivas ministradas na forma síncrona, incluindo soluções de problemas, através de plataformas de videoconferência; (2) leituras feitas pelos alunos de material disponibilizado pelo professor; (3) a realização de competições de programação semanais com problemas ligados ao tema de semana que serão usadas para avaliação

A disciplina será desenvolvida através de aulas síncronas de 2h de duração. Nestas aulas serão apresentadas as técnicas descritas para resoluções de problemas, e que serão testadas naquela semana.

Os materiais de leitura e a abertura das competições serão disponibilizadas aos alunos logo após a aula síncrona sobre aquele assunto. Espera-se que o aluno gaste 3 horas no decorrer da semana para resolver os exercícios da prova.

A interação efetiva com os alunos irá ocorrer nas aulas síncronas (na sala <https://bbb.c3sl.ufpr.br/b/andyct-in4-2mt>) e no atendimento remoto via consulta por e-mail. O material das atividades assíncronas será formado de recomendações de leitura, a gravação da aula síncrona e a competição.

Haverá um total de 12 competições semanais, cada uma composta de aproximadamente 10 problemas. A competição terá duração de uma semana a partir da aula semanal e deve ser feita individualmente. Após esta uma semana, é permitida a discussão geral dos problemas, e a solução contribui para a nota.

FORMAS DE AVALIAÇÃO

A cada semana uma nota será atribuída para o desempenho na competição. A média será a média simples das 12 notas. Pontos extras poderão ser atribuídos para problemas resolvidos fora dos prazos de cada competição.

A prova final será no dia 17/12/2021.



BIBLIOGRAFIA BÁSICA (mínimo 03 títulos)

- [1] C. H. Papadimitriou e K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Pub., 1998.
- [2] Donald L. Kreher e Douglas R. Stinson. *Combinatorial algorithms: generation, enumeration, and search*. Discrete Mathematics and Its Applications. Boca Raton, Florida: CRC Press, 1999. ISBN: 978-0849339882.
- [3] Thomas H. Cormen et al. *Introduction to Algorithms*. 3ª ed. The MIT Press, 2009.
- [4] Antti Laaksonen. *Guide to Competitive Programming: Learning and Improving Algorithms Through Contests*. Springer International Publishing AG, 2017. DOI: 10.1007/978-3-319-72547-5.
- [5] Steven Halim e Felix Halim. *Competitive Programming 3: The New Lower Bound of Programming Contests*. 3ª ed. 2013.
- [6] Steven S. Skiena. *The Algorithm Design Manual*. 3ª ed. Texts in Computer Science. Springer International Publishing, 2020. DOI: 10.1007/978-3-030-54256-6.

BIBLIOGRAFIA COMPLEMENTAR (mínimo 05 títulos)

- [7] Antti Laaksonen. *Competitive Programmer's Handbook*. 2018. URL: <https://cses.fi/book/book.pdf>.
- [8] Krzysztof Diks et al. *Looking for a Challenge 2: Problems from the Polish Collegiate Programming Contest 2011–2014*. 2ª ed. University of Warsaw, 2019. URL: <https://www.mimuw.edu.pl/~idziaszek/algnotes/looking-for-a-challenge-2-en.pdf>.
- [9] Jon Kleinberg e Éva Tardos. *Algorithm Design*. Pearson Education, 2006.
- [10] Steven S. Skiena e Miguel A. Revilla. *Programming Challenges: The Programming Contest Training Manual*. Springer-Verlag New York, 2003.
- [11] Ivanov Maxim et al. *CP Algorithms*. URL: <http://e-maxx.ru/algo/> e <https://cp-algorithms.com/> e <https://samuraiwesleyjack.github.io/cp-algoritmos/>.
- [12] Arthur Pratti Dadalto. *Ementa Modalidade Programação (OBI)*. URL: <https://olimpiada.ic.unicamp.br/info/ementa/geral/>.
- [13] Victor de Oliveira Colombo. *Material didático sobre algoritmos gulosos*. 2018. URL: <https://linux.ime.usp.br/~colombo/mac0499/monografia.pdf>.
- [14] Yan Soares Couto. *Algoritmos em sequências*. 2016. URL: <https://bcc.ime.usp.br/tccs/2016/yancouto/tcc.pdf>.
- [15] Adrian Vladu e Cosmin Negruşeri. *Suffix arrays – a programming contest approach*. 2005. URL: <http://web.stanford.edu/class/cs97si/suffix-array.pdf>.
- [16] Maxim Akhmedov. *Dynamic programming optimizations*. 2017. URL: <http://maratona.ic.unicamp.br/MaratonaVerao2017/documents/dp.pdf>.
- [17] Victor Lecomte. *Handbook of geometry for competitive programmers*. 2018. URL: <https://vlecomte.github.io/cp-geo.pdf>.

Professor da Disciplina: André Luiz Pires Guedes

Assinatura: _____



Ministério da Educação
Universidade Federal do Paraná
Setor de Ciências Exatas
Departamento de Informática

Chefe de Departamento: Fabiano Silva

Assinatura: _____

OBS (1): ao assinalar a opção % EAD, indicar a carga horária que será à distância.

Disciplina: Tópicos em Algoritmos - Desafios de Programação

Docente responsável: André Guedes (andre@inf.ufpr.br)

Colaboradores: Fernando Monteiro Kiotheka (fmk17@inf.ufpr.br)
e Raul Gomes Pimentel de Almeida (rgpa18@inf.ufpr.br)

Carga horária: 60 horas, distribuídas em 12 semanas (5 horas por semana, em média)

Modalidades e meios:

- Atividades síncronas:
videoconferência via BBB (na sala <https://bbb.c3sl.ufpr.br/b/and-yct-in4-2mt>)
- Atividades assíncronas:
página da disciplina

Cronograma detalhado:

1a semana - 2021-09-24

1. Introdução a online judges, entrada/saída em C++
2. Estrutura do problema, como lidar com múltiplos casos de teste.
3. Templates, competições.
4. Estruturas de dados em C++: vector, stack, priority_queue, queue.
5. Funções de ordenação, STL em geral em C++. Pair. nth-element.
6. Análise de algoritmos básica. Complexidade e estimativa.
7. minstack/minqueue.
https://cp-algorithms.com/data_structures/stack_queue_modification.html

2a semana - 2021-10-01

- Busca completa.
1. Bitset e geração de permutações. (2^n e etc).
 2. Conjuntos (se mescla com set e bitset).
 3. Testar todas as permutações com next_permutation.
 4. Backtracking: 8 queens e problema dos tapetes de karatê.
- Matemática.
6. Fórmula de progressão aritmética, fórmula de progressão geométrica.
 7. Triângulo de pascal (fatorial).
 8. Exponenciação binária.
 9. Inclusão/exclusão.
 10. Janela deslizante.

3a semana - 2021-10-08

1. Divisão e Conquista: Merge Sort (inversão).
2. Busca binária.
3. Árvore de Fenwick.
4. Árvore de Segmentos (não preguiçoso).

4a semana - 2021-10-15

1. Soma de prefixos.

2. Solução gananciosa/gulosa.
 3. Programação dinâmica. Memoização, iterativa.
 4. Problema da mochila.
 5. Longest Increasing Subsequence.
 7. Contagem de soluções (com módulo de alguma coisa, etc).
 8. Paths in a Grid.
 9. Probabilidade
 10. Contagem de caminhos no Robot in a Grid (AtCoder).
- Requer inverso multiplicativo, dá pra usar $^{-1}$ e falar depois.

5a semana - 2021-10-22

1. DFS, BFS.
2. Detecção de ciclos com DFS.
3. Busca de caminho mínimo. Dijkstra, Floyd-Warshall, Bellman-Ford.
4. Árvore Geradora Mínima: Prim.
5. Kruskal e DSU.
6. Busca do menor ancestral comum.

6a semana - 2021-10-29

Semana leve, só técnicas aleatórias

1. Dois ponteiros.
2. Emparelhamento estável.
3. 4-vizinhança, 8-vizinhança.
4. Compressão de coordenadas.
5. Multiset/Bag ordenado por frequência.
6. Teto de divisão por inteiros.

7a semana - 2021-11-05

Teoria dos números.

1. Fatoração.
2. Maior divisor comum/Mínimo múltiplo comum.
3. Crivo de Erastótenes e variações.
4. Número de divisores.
5. Função totiente de Euler.
6. Módulo.
7. Euclides estendido, e inverso multiplicativo.
8. Teorema chinês do resto.
9. Exponenciação binária de matrizes: Fibonacci.
10. Pollard Rho.

8a semana - 2021-11-12

1. Articulações e pontes.
2. Componentes conexos: Kosaraju e Tarjan.
3. Emparelhamento bipartido de grafos.
4. Fluxo máximo e corte mínimo.
5. Stoer-Wagner.

9a semana - 2021-11-19

Decomposições.

1. Decomposições em raiz quadrada.
2. Algoritmo de Mo.
3. Árvore de segmentos preguiçosa.
4. Decomposição em cadeias (HLD).

10a semana - 2021-11-26

String.

1. Trie.
2. KMP.
3. Regex.
4. Vetor de sufixos.
5. Autômato de sufixos.
6. Árvore de palíndromo.

11a semana - 2021-12-03

Teoria dos jogos.

1. Nim.
2. Teorema de Sprague-Grundy.
3. Slow k-Nim.
4. Staircase Nim.

12a semana - 2021-12-10

Geometria.

1. Representação de polígonos.
2. Varredura linear.
3. Fecho convexo.
4. Voronoi.

13a semana (opcional)

1. Probabilidade.
2. Árvore de Fenwick 2D.
3. Número de Catalão.
4. Função Z.
5. Treap.
6. FFT.
7. Coisas mais complicadas.

Prova final - 2021-12-17