



Ficha 2 (variável)

Disciplina: Algoritmos e Estrutura de Dados II						Código: CI056 / CI1056	
Natureza: (X) Obrigatória () Optativa		(X) Semestral () Anual () Modular					
Pré-requisito: CI1055		Co-requisito:	Modalidade: () Presencial (X) Totalmente EaD () ____ *c.H.EaD				
CH Total: 60 CH semanal: 6		Padrão (PD): 30	Laboratório (LB): 00	Campo (CP): 00	Estágio (ES): 00	Orientada (OR): 00	Prática Específica (PE): 30
Estágio de Formação Pedagógica (EFP):		Extensão (EXT): 00	Prática como Componente Curricular (PCC): 00				
<p>Indicar a carga horária semestral (em PD-LB-CP-ES-OR-PE-EFP-EXT-PCC) *Indicar a carga horária que será à distância.</p> <p style="text-align: center;">EMENTA (Unidade Didática)</p> <p>Recursão, Busca, Ordenação, Heaps, Contagem de recursos computacionais.</p>							
<p style="text-align: center;">PROGRAMA (itens de cada unidade didática)</p> <p>1 - Introdução à Recursão: 1.1 - Exemplos numéricos dentre fatorial, potenciação, Fibonacci, ... 1.2 - Definição de problemas computacionais. 1.3 - Contagem de passos específicos (por exemplo, comparações, somas, trocas, ...) e relações de recorrência (sem notação assintótica)</p> <p>2 – Busca: 2.1 - Mínimo de vetor. 2.2 - Busca em vetor. 2.3 - Busca em vetor ordenado. 2.4 - Busca binária.</p> <p>3 – Ordenação: 3.1 – InsertionSort. 3.2 – SelectionSort. 3.3 - Intercalação e MergeSort. 3.4 - Particionamento e QuickSort.</p> <p>4 – Heaps: 4.1 – Heap. 4.2 – HeapSort. 4.3 - Fila de Prioridade.</p> <p>5 - Removendo Recursão: 5.1 - Removendo recursão de cauda 5.2 - Removendo recursão usando pilha</p>							
<p style="text-align: center;">OBJETIVO GERAL</p> <p>Introduzir a técnica de algoritmos recursivos, aplicada a problemas de busca, ordenação de vetores, problemas numéricos simples e à estrutura de dados heap. Avaliar a quantidade de recursos computacionais necessários para os algoritmos apresentados, usando relações de recorrência simples.</p> <p style="text-align: center;">OBJETIVO ESPECÍFICO</p> <p>Apresentar a técnica de recursividade. Desenvolver algoritmos recursivos para problemas numéricos básicos. Desenvolver algoritmos recursivos de busca. Ser capaz de implementar recursivamente algoritmos de ordenação quadráticos e $O(n \log n)$. Descrever a propriedade heap e o uso de heaps na implementação de filas de prioridade. Avaliar a eficiência em relação a um recurso usado pelo algoritmo (comparações, trocas, somas, ...). Explicar o que se entende pelo melhor e pior caso do comportamento de um algoritmo. Resolver relações de recorrência simples</p>							



(i.e., sem uso de notação assintótica) que contam recursos específicos. Executar estudos empíricos para validar hipóteses sobre os recursos consumidos decorrentes da análise matemática.

PROCEDIMENTOS DIDÁTICOS

O curso mescla aulas: (1) aulas teóricas expositivas ministradas nas formas síncronas e assíncronas, incluindo soluções de exercícios propostos em sala de aula, através de plataformas de videoconferência e transmissão via casting das aulas gravadas (2) aulas práticas nas quais os alunos praticam programação dos exercícios propostos sob a supervisão do professor. Muitos exercícios são propostos como exercícios complementares para serem feitos fora dos horários de aula. (3) nas aulas síncronas, serão respondidas dúvidas dos alunos.

FORMAS DE AVALIAÇÃO

2 Provas e 2 trabalhos práticos.

BIBLIOGRAFIA BÁSICA (mínimo 03 títulos)

- 1 - R. Sedgewick, K. Wayne, Algorithms, Addison-Wesley, 4a Edição, 2011.
- 2 - P. Feofiloff. Algoritmos em linguagem C. Elsevier, 2009.
- 3 - T. Cormen, C. Leiserson, R. Rivest, C. Stein, Algoritmos: Teoria e Prática, Elsevier, 2012.

BIBLIOGRAFIA COMPLEMENTAR (mínimo 05 títulos)

- 1 - S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani, Algoritmos, McGraw-Hill, 2009.
- 2 - J. Kleinberg and E. Tardos, Algorithm Design, Addison-Wesley, 2006.
- 3 - U. Manber, Introduction to Algorithms: A Creative Approach, Addison-Wesley, 1989.
- 4 - D. E. Knuth, The Art of Computer Programming, Addison-Wesley, 2a edição, 1998.
- 5 - K. H. Rosen. Discrete mathematics and its applications. McGraw-Hill Higher Education, 6a edição, 2007.

Professor da Disciplina: André Luís Vignatti

Chefe de Departamento ou Unidade equivalente: Fabiano Silva



CRONOGRAMA – 2021 - 1º SEMESTRE

Abaixo, a descrição das 30 unidades, cada unidade com 2 horas, totalizando 60h.

- 01 - 24/set - (LIVE) APRESENTAÇÃO DA DISCIPLINA
- 02 - 27/set - Introdução e Problemas Computacionais
- 03 - 29/set - Recursão - Fatorial, Potenciação, Fibonacci, etc.
- 04 - 01/out - (LIVE) - PRÁTICA DE ALGORITMOS RECURSIVOS (exercícios variados)
- 05 - 04/out - Mínimo de Vetor
- 06 - 06/out - Busca em Vetor
- 07 - 08/out - (LIVE) - PRÁTICA DE EXERCÍCIOS TEÓRICOS (resolver recorrências simples)
- 08 - 07/out - Busca em Vetor Ordenado
- 09 - 09/out - Busca Binária
- 10 - 11/out - (LIVE) - PRÁTICA DE ALGORITMOS RECURSIVOS (medir tempo recursivo x iterativo)

- 11 - 11/out - Insertion Sort
- 12 - 13/out - SelectionSort e MergeSort
- 13 - 15/out - (LIVE) - PRÁTICA DE EXERCÍCIOS TEÓRICOS (resolver recorrências mais complexas com melhor e pior caso)
- 14 - 18/out - QuickSort
- XX - 20/out - DATA RESERVADA PARA REPOSIÇÃO
- 15 - 22/out - PROVA 1

- 16 - 25/out - Filas de prioridade, heap, max-heapify
- 17 - 27/out - Construção de um heap, Ordenação por heap (heapsort)
- 18 - 29/out - (LIVE) - IMPLEMENTAÇÃO DAS OPERAÇÕES DE HEAP
- 19 - 01/nov - Panorama dos algoritmos de ordenação
- 20 - 03/nov - Limite inferior para ordenação
- 21 - 05/nov - (LIVE) - PANCAKE SORT
- 22 - 08/nov - CountingSort, RadixSort, BucketSort
- 23 - 10/nov - Permutações e Combinações
- 24 - 12/nov - (LIVE) - GERAR TODOS VETORES BINÁRIOS E SUBCONJUNTOS
- 25 - 15/nov - Backtracking Recursivo
- 26 - 17/nov - Backtracking com Pilha
- 27 - 19/nov - (LIVE) - PASSEIO DO CAVALO
- 28 - 20/nov - Remoção de Recursão (de cauda e com pilha)
- 29 - 24/nov - DATA RESERVADA PARA REPOSIÇÃO
- 30 - 03/dez - PROVA 2

- XX - 10/dez - PROVA FINAL

contato: vignatti@inf.ufpr.br

sala virtual: informações em www.inf.ufpr.br/vignatti/v/cursos/ci056.html