



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
Rua Dom Manoel de Medeiros, s/n – Dois Irmãos 52171-900 Recife-PE
Fone: 81 3302 1000 www.dc.ufrpe.br

DISCIPLINA: Algoritmos e Estruturas de Dados	CÓDIGO: 06214
DEPARTAMENTO: Computação	ÁREA: Informática
CURSO: Licenciatura em Computação	
PROFESSOR RESPONSÁVEL: Luciano Demétrio Santos Pacífico	
DATA MÁXIMA DE ENTREGA: 29-08-2020 (prazo estendido e final!)	

Regras da Lista de Exercícios 01

1. As **respostas** das questões da Lista de Exercícios 01 devem ser enviadas através de **um único arquivo, no formato “.pdf”**.
2. As respostas da Lista de Exercícios 01 devem ser submetidas **unicamente através da tarefa criada no Google Classroom para este propósito**.
3. As questões que solicitam **escrita de código** devem ser resolvidas **apenas através dos recursos oferecidos pela pseudolingagem definida para a disciplina**, sendo eles: variáveis, constantes e tipos primitivos, expressões, estruturas condicionais, estruturas de repetição, sub-rotinas, estruturas de dados homogêneas (Arrays) e estruturas de dados heterogêneas.
4. **Todos as Estruturas e Dados e Algoritmos devem ser implementados pelo aluno**. Não é permitido o uso de Estruturas de Dados, Algoritmos e estruturas otimizadas de Linguagens de Programação reais.
5. **Todos os alunos envolvidos em cópias terão suas notas ANULADAS nas referidas questões**.

Lista de Exercícios 01 – Introdução ao Estudo de Algoritmos

1. Escreva um **algoritmo em Linguagem Natural** descrevendo, em, **no mínimo, 10 passos**, sua rotina diária durante a quarentena. Obs: ao menos **um comando de decisão** e **um comando de repetição** devem ser usados. (1.5 ponto)
2. Escreva um **algoritmo (procedimento)**, na **pseudolingagem** adotada na disciplina, que receba **um vetor de dados x n -dimensional**, e retorne o resultado da seguinte função: (2.0 pontos)

$$f(x) = \sum_{i=1}^n x_i^i$$

Obs: usar **laços de repetição** e **comandos de decisão** (estruturas condicionais), além de **operações aritméticas fundamentais** (suponha que a linguagem **não possui a função para potenciação**).

3. Implemente, na **pseudolinguagem** adotada na disciplina, um **algoritmo** que receba duas matrizes $\mathbf{A}_{n \times m}$ e $\mathbf{B}_{s \times p}$, e calcule o **produto matricial** entre as mesmas. Obs.: Os usuários poderão fornecer **qualquer tipo de matrizes bidimensionais** como entrada, **cabendo ao algoritmo avaliar** se é possível calcular o produto matricial entre A e B. Caso não seja possível, uma mensagem de erro deve ser impressa indicando o problema. (3.0 pontos)
4. Supondo que três algoritmos (A_1 , A_2 e A_3) sejam capazes de resolver o mesmo problema, e que cada um deles realize o seguinte número de operações durante sua execução: (1.5 ponto)
 - $A_1(n) = 1000n + 9000$ operações.
 - $A_2(n) = n^2 + 15n$ operações.
 - $A_3(n) = 10^n + 100n$ operações.

Responda:

- a. Qual dos três algoritmos seria a melhor escolha, em termos de complexidade, para resolver nosso problema, considerando a análise assintótica?
 - b. A partir de qual valor de n esse algoritmo passa a ser o menos custoso?
5. Demonstre o funcionamento do algoritmo *Insertion-Sort* no vetor $A = \{52, 3, 17, 10, 45, 13, 5, 9\}$, informando a cada **iteração do laço para**, o número de trocas realizadas, assim como a configuração do vetor de dados. Qual o número total de trocas? (2.0 pontos)

[illegible]