

IMD0029 - Estrutura de Dados Básicas 1 –2023.2 – Prova 01
Prof. Eiji Adachi M. Barbosa

Nome: _____

Matrícula: _____

ANTES DE COMEÇAR A PROVA, leia atentamente as seguintes instruções:

- Esta é uma prova escrita de caráter individual e sem consultas a pessoas ou material (impresso ou eletrônico).
- A prova vale 5,0 pontos na Unidade I e o valor de cada questão é informado no seu enunciado.
- Preze por respostas legíveis, bem organizadas e simples.
- As respostas devem ser em caneta. Respostas em lápis serão aceitas, mas eventuais questionamentos sobre a correção não serão aceitos.
- Celulares e outros dispositivos eletrônicos devem permanecer desligados durante toda a prova.
- **Desvios éticos ou de honestidade levarão a nota igual a zero na Unidade 1.**

Questão 1: (1,5ponto) Faça uma função de busca que recebe como entrada o array A, o seu tamanho N e um inteiro K e retorna um inteiro i que indica o índice do elemento mais à direita no array A que seja igual ao valor de K. Caso não sejam encontrados elementos iguais a K, deve-se retornar -1. Por exemplo: considerando o array $A = \{0, 2, 2, 2, 3, 8, 8, 8, 10\}$; se $K = 2$, sua função deve retornar 3, e se $K = 1$, sua função deve retornar -1. Considere que o array A conta com possíveis repetições e já está ordenado em ordem crescente. Sua função deverá obrigatoriamente ser recursiva, ter complexidade $O(\lg(n))$ e seguir a assinatura:

`int busca(int A[], int esq, int dir, int K)`

Obs.: Nesta questão, não podem ser usadas instruções para realizar repetição, como for, while e do-while. Ou seja, você deverá construir sua solução apenas com chamadas recursivas.

Questão2: (1,5 ponto) Dado um número natural N, implemente uma função recursiva chamada `contaImpares` que retorne a quantidade de dígitos ímpares que o número N possui. Por exemplo, se N for igual a 2746, sua função deve retornar 1, pois 2746 possui 1 dígito ímpar, que é o 7. Sua função deverá seguir a assinatura:

`int contaImpares(int N)`

Obs.: Nesta questão, não podem ser usadas instruções para realizar repetição, como for, while e do-while. Ou seja, você deverá construir sua solução apenas com chamadas recursivas.

Questão 3: (2,0 pontos) Para cada uma das afirmações a seguir, marque V (verdadeiro) ou F (falso), **justificando sucintamente** sua resposta. Marcações de V ou F **sem justificativas não serão aceitas.**

- 1 – (☐) Os algoritmos de busca sequencial e de busca binária podem ser empregados em quaisquer arrays passados como entrada.
- 2 – (☐) Os algoritmos de busca sequencial e de busca binária possuem mesma complexidade assintótica para o pior caso.
- 3 – (☐) O algoritmo de ordenação Selection Sort sempre executa o mesmo número de comparações entre elementos, independentemente do array de entrada, tornando-o previsível em termos de tempo de execução.
- 4 – (☐) O algoritmo de ordenação Insertion Sort sempre executa o mesmo número de comparações entre elementos, independentemente do array de entrada, tornando-o previsível em termos de tempo de execução.
- 5 – (☐) Considerando o melhor caso, o Insertion Sort tem menor complexidade assintótica do que o Selection Sort.
- 6 – (☐) Considerando o pior caso, o Selection Sort tem menor complexidade assintótica do que o Insertion Sort.
- 7 – (☐) O algoritmo de ordenação Merge Sort possui a mesma complexidade assintótica no melhor e no pior caso.
- 8 – (☐) O algoritmo de ordenação Quick Sort possui a mesma complexidade assintótica no melhor e no pior caso.