

IMD0029 - Estrutura de Dados Básicas 1 –2024.1 – Prova 02
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 II e o valor de cada questão é informado no seu enunciado.
- Preze por respostas legíveis, bem organizadas e simples.
- As respostas devem ser fornecidas preferencialmente à caneta. Respostas fornecidas à 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 2.**

Questão 1 (Responda na folha extra): (1,5 ponto) Considere uma lista duplamente encadeada com sentinelas cabeça e cauda, conforme a estrutura abaixo:

| | |
|---|--|
| <pre>Lista { No* cabeca; No* cauda; }</pre> | <pre>No{ No* proximo; No* anterior; int valor; }</pre> |
|---|--|

Nesta questão, implemente o método:

```
void Lista::removerRepetidos()
```

Este método remove da lista todos elementos com valores duplicados, exceto o primeiro; ou seja, caso haja elementos com valores duplicados, o primeiro deles deve ser mantido na lista e os demais removidos. Por exemplo, a lista encadeada: 1 <-> 2 <-> 2 <-> 2 <-> 3 após a chamada ao método `removerRepetidos` ficará no seguinte estado: 1 <-> 2 <-> 3.

A sua implementação deverá ter, obrigatoriamente, complexidade $O(n)$. Assuma nesta questão que a lista sempre estará ordenada em ordem crescente.

Questão 2: (1,5 ponto) Na computação, Conjunto é um Tipo Abstrato de Dados (TAD) que armazena uma coleção de elementos únicos, isto é, não há elementos repetidos num conjunto. As principais operações do TAD Conjunto são: (1) adicionar um elemento ao conjunto, (2) remover um elemento do conjunto e (3) checar se um determinado elemento está contido ou não no conjunto.

Suponha que um Conjunto foi implementado em C++ usando um array.

Considerando a operação de checar se um determinado elemento está contido no conjunto, e considerando sempre o pior caso dessa operação, existe vantagem em manter os elementos da lista sempre ordenados? Justifique sua resposta do ponto de vista da complexidade de tempo da operação de checar se um elemento está contido no conjunto.

Questão 3 (Responda nos espaços abaixo de cada afirmação): (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 – () Ao implementar uma estrutura de dados seguindo a estratégia *First-In First-Out* (FIFO) – primeiro a entrar primeiro a sair – usando uma lista simplesmente encadeada com ponteiro para o início, é possível implementar as operações de inserir e remover com complexidade de tempo constante.

2 – () Ao implementar uma estrutura de dados seguindo a estratégia *First-In Last-Out* (FILO) – primeiro a entrar último a sair – usando uma lista simplesmente encadeada com ponteiro para o início, é impossível implementar as operações de inserir e remover com complexidade de tempo constante.

3 – () Numa lista simplesmente encadeada, é possível remover o último elemento da lista em complexidade de tempo constante, bastando manter um ponteiro "fim" sempre apontando para o último elemento da lista.

4 – () Numa lista simplesmente encadeada, é possível remover o primeiro elemento da lista em complexidade de tempo constante, bastando manter um ponteiro "início" sempre apontando para o primeiro elemento da lista.