

IMD0029 - Estrutura de Dados Básicas 1 –2019.2 – 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 (*head*) e cauda (*tail*) implementada conforme a estrutura abaixo:

```
List{
    Node* head;
    Node* tail;
}
```

```
Node{
    Node* next;
    Node* previous;
    int value;
}
```

Dada esta estrutura de lista duplamente encadeada, e considerando que os elementos da lista já estão em ordem crescente de acordo com o atributo 'value', implemente um método iterativo que remove os elementos repetidos da lista. Em outras palavras, caso existam elementos repetidos na lista, este método deve deixar apenas um destes elementos, removendo os demais. Ex.: A lista [2 ↔ 4 ↔ 4 ↔ 4 ↔ 5 ↔ 5 ↔ 6 ↔ 7 ↔ 7 ↔ 8] após a execução do método deverá ficar igual a [2 ↔ 4 ↔ 5 ↔ 6 ↔ 7 ↔ 8]. A sua implementação deverá ter, obrigatoriamente, complexidade $O(n)$ e deverá seguir a assinatura:

```
int List::removeRepeated() - Remove os elementos repetidos da lista, retornando quantos elementos foram removidos.
```

Obs.: Você pode criar métodos auxiliares, caso acho necessário, desde que seu código também seja apresentado.

Questão 2 (Responda na folha extra): (1,5 ponto) Na computação, Conjunto é um Tipo Abstrato de Dados (TAD) que armazena uma coleção de elementos únicos, isto é, não existe 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 queremos implementar um Conjunto de elementos inteiros usando arrays em C++. Podemos realizar tal implementação seguindo duas abordagens: uma que mantém ordenados os elementos no array, seja em ordem crescente, ou decrescente; e outra que não mantém os elementos ordenados. Considerando apenas as operações de inserir e remover elementos do Conjunto, existe vantagem, em termos de complexidade assintótica, em manter o array já ordenado? Justifique sua resposta.

Questão 3 (Responda neste espaço): (1,0 ponto) Defina o que são e explique para que servem os nós sentinelas usados em listas encadeadas.

Questão 4 (Responda nos espaços abaixo de cada afirmação): (1,5 ponto) 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 – (**F**) Em listas simplesmente encadeadas, não é possível realizar as operações de inserir e remover no início da lista em tempo constante.

Falso. Como a lista possui um ponteiro para o começo, é SIM possível realizar operações de inserir e remover no início da lista, por conta da referência direta

2 – (**F**) Em listas simplesmente encadeadas, não é possível realizar as operações de inserir e remover no fim da lista em tempo constante.

Falso. Como a lista possui um ponteiro para o fim, é SIM possível inserir um elemento no fim da lista em tempo constante

3 – (**V**) Usando uma lista simplesmente encadeada com apenas ponteiro para o início (primeiro nó), não é possível implementar uma estrutura de dados seguindo a estratégia *First-In First-Out* (FIFO) – primeiro a entrar primeiro a sair – com operações de inserir e remover elementos em tempo constante.

Verdadeiro. No cenário descrito, não é possível realizar a operação de inserir em tempo constante, pois precisaríamos adicionar novos elementos no final da lista, sendo preciso percorrer todos os outros para achar sua referência.

4 – (**F**) Usando uma lista simplesmente encadeada com apenas ponteiro para o início (primeiro nó), não é possível implementar uma estrutura de dados seguindo a estratégia *First-In Last-Out* (FILO) – primeiro a entrar último a sair – com operações de inserir e remover elementos em tempo constante.

Falso. Pelo fato das operações ocorrerem em um mesmo lado da lista, é possível adicionar e remover em tempo constante tendo somente um ponteiro para o início