

IMD0029 - Estrutura de Dados Básicas 1 – 2017.1 – 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 10,0 pontos 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 em caneta. Respostas fornecidas a 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 à anulação da prova do candidato (nota igual a zero).

Questão 1: (3,0 pontos) Após serem perseguidos por um bando de orcs, Gandalf e os anões separam-se de Bilbo Bolseiro, deixando-o desacordado para trás. Logo após acordar, caminhando pela caverna onde foi deixado, Bilbo avista a criatura Gollum, que propõe um jogo de adivinhação. Se Bilbo ganhar, Gollum lhe mostrará a saída. Em sua última charada, Gollum apela e apresenta o seguinte problema:

*“A expressão tem inteiro e tem operador,
Inteiros vem primeiro, operadores vem depois,
São quatro operadores diferente,
E tem dois operandos para cada operador,
Na entrada eu dou uma expressão, mas na saída, qual é o valor?”*

Nesta questão, ajude Bilbo a resolver a charada-problema de Gollum para que ele possa fugir da caverna. Para isso, implemente uma função que recebe como entrada uma string representando uma expressão aritmética em notação pós-fixa e retorna o seu valor numérico correspondente, usando obrigatoriamente uma estrutura **PILHA** já implementada. Considere que a string de entrada é válida e contém apenas inteiros, e símbolos para as quatro operações aritméticas (+ - * /), e que os elementos básicos da expressão (números inteiros e símbolos das operações) são separados por um espaço em branco. Sua função deverá seguir a assinatura abaixo:

```
/* exp - expressão aritmética em notação pós-fixa.  
tamExp - quantidade de caracteres da string exp*/  
int AvaliarExpressao( String exp, int tamExp );
```

Dica: Notação pós-fixa é aquela em que os operadores são escritos depois dos operandos. Por exemplo, a expressão aritmética em notação infixa $1+2*3$ é representada na notação pós fixa como $1\ 2\ 3\ *\ +$.

Obs.: Assuma que a estrutura pilha já está implementada e você pode reusar todas as suas operações básicas.

Questão 2: (3,5 pontos) Após fugirem de Moria, onde Gandalf lutou com um balrog, os membros da Sociedade do Anel encontram refúgio junto a Galadriel, em Lothlórien. Certa noite, Galadriel enche uma bacia prateada de água de uma fonte e cria um espelho mágico. Ela permite que Frodo olhe no espelho, mas o adverte sobre as dificuldade de entender as visões. Quando Frodo olha no espelho, ele vê diversas imagens do seu futuro, mas não as compreende bem. As visões aparecem numa lista simplesmente encadeada com nós sentinelas cabeça (head) e cauda (tail). Tal lista encadeada deveria estar ordenada em ordem cronológica crescente. No entanto, por algum motivo que Frodo não consegue entender, nem todas as imagens do seu futuro estão corretamente ordenadas. Em particular, as primeiras imagens do seu futuro aparecem em ordem cronológica crescente, mas a partir de um determinado ponto as imagens do seu futuro aparecem em ordem cronológica decrescente, formando uma sequência bitônica. De modo esquemático, a lista encadeada que Frodo vê no espelho de Galadriel tem o seguinte formato: {1 -> 2 -> 3 -> 4 -> 10 -> 9 -> 8 -> 7 -> 6 -> 5}. Nesta questão, ajude Frodo a organizar as imagens do seu futuro. Para isto, implemente uma função **ITERATIVA** que recebe como entrada uma lista encadeada representando uma sequência bitônica (como exemplificado acima) e modifica a disposição dos nós da lista de modo a deixá-la ordenada em ordem cronológica crescente. Sua função deverá seguir a assinatura abaixo:

```
void OrganizarLista( Lista l );
```

Considere para a Questão 2 a seguinte estrutura de lista encadeada:

<pre>List{ Node* head; Node* tail; }</pre>	<pre>Node { Node* next; int time; }</pre>
--	---

Dica: Sequência bitônica é aquela formada por uma sub-sequência inicial crescente, seguida de uma sub-sequência decrescente.

Obs.: Não é permitido usar outras estruturas auxiliares, como listas ou arrays; só é permitido operar sobre a lista de entrada. Para tanto, é permitido a criação de ponteiros auxiliares. As operações básicas sobre a lista não podem ser reusadas; faça sua solução apenas manipulando os nós da lista.

Questão 3: (2,0 pontos) A Batalha do Abismo de Helm chega ao fim. Durante toda a batalha, Legolas e Gimli travaram uma competição interna para ver quem conseguia derrotar mais criaturas. Cada um registrou numa lista simplesmente encadeada com ponteiro para o início (first) as criaturas derrotadas. Cada nó da lista possui um inteiro que indica a pontuação equivalente à criatura derrotada. Nesta questão, ajude Legolas e Gimli a definir quem foi o vencedor da competição deles. Para isto, implemente uma função que recebe como entrada uma lista simplesmente encadeada descrevendo as criaturas derrotadas por um dos heróis, e retorna o valor numérico correspondente a pontuação total alcançada. Sua solução deverá obrigatoriamente ser **RECURSIVA** e deverá seguir a assinatura abaixo:

```
int CalcularPontos( Lista l );
```

Considere para esta questão a seguinte estrutura para a lista encadeada:

<pre>List{ Node* first; }</pre>	<pre>Node { Node* next; int value; }</pre>
-------------------------------------	--

Obs.: Não é permitido usar outras estruturas auxiliares, como listas ou arrays; só é permitido operar sobre a lista de entrada. É permitido usar funções auxiliares, desde que a solução como um todo seja recursiva. As operações básicas sobre a lista não podem ser reusadas; faça sua solução apenas manipulando os nós da lista.

Questão 4: (1,5 ponto) Finalmente o Um Anel é destruído, juntamente com o espírito de Sauron. É o fim da saga do Um Anel. Aragorn é coroado Rei e casa-se com Arwen. A paz também retorna à terra dos hobbits, o Condado. O velho Bilbo, junto a Frodo, Galadriel, Elrond e Gandalf, partem num barco em direção a Valinor. Durante a viagem, Bilbo escreve suas últimas memórias num velho diário. Um capítulo especial deste diário contém as memórias de Bilbo sobre tipos abstratos de dados élficos. Sem confiar completamente em sua já fraca memória, Bilbo pede ajuda a você para revisar se suas afirmações são verdadeiras, ou falsas. Desta forma, para cada uma das afirmações a seguir, marque V (verdadeiro) ou F (falso), **JUSTIFICANDO TODAS SUAS RESPOSTAS**.

- () As operações de inserir e remover no início e no fim da lista são realizadas em tempo constante numa lista simplesmente encadeada com ponteiros para início e fim.
- () As operações de inserir e remover no início e no fim da lista são realizadas em tempo constante numa lista duplamente encadeada com sentinelas cabeça e cauda.
- () De posse de uma implementação do TAD Sequência, é possível criarmos uma estrutura que segue a estratégia F.I.L.O. (first-in, last-out) da seguinte forma: ao realizarmos uma *inserção* nós inserimos no início da sequência e ao realizarmos uma *remoção* nós removemos do fim da sequência.
- () De posse de uma implementação do TAD Sequência, é possível criarmos uma estrutura que segue a estratégia F.I.F.O. (first-in, first-out) da seguinte forma: ao realizarmos uma *inserção* nós inserimos no início da sequência e ao realizarmos uma *remoção* nós removemos do fim da sequência.
- () Ao implementarmos os TADs Pilha e Fila com base em arrays, não conseguimos prover as funções de inserir (Pilha.push e Fila.queue) e remover (Pilha.pop e Fila.dequeue) em tempo constante.