



Pontifícia Universidade Católica de Minas Gerais
Curso de Ciência da Computação
Disciplina: Algoritmos e Estruturas de Dados II
Profs.: Felipe Domingos da Cunha, Max do Val Machado e
Rodrigo Richard Gomes

Trabalho Prático III

Regras Básicas

1. `extends Trabalho Prático 02`
2. Fique atento ao Charset dos arquivos de entrada e saída.



Spotify é um serviço de streaming de música, podcast e vídeo que foi lançado oficialmente em 7 de outubro de 2008. É o serviço de streaming mais popular e usado do mundo. Ele é desenvolvido pela startup Spotify AB em Estocolmo, Suécia. Ele fornece conteúdo protegido de conteúdo provido de restrição de gestão de direitos digitais de gravadoras e empresas de mídia. O Spotify é um serviço freemium; com recursos básicos sendo gratuitos com propagandas ou limitações, enquanto recursos adicionais, como qualidade de transmissão aprimorada e downloads de música, são oferecidos para assinaturas pagas.

O Spotify está disponível na maior parte da Europa, parte da América, Austrália, Nova Zelândia e partes da Ásia. Está disponível para a maioria dos dispositivos modernos, incluindo computadores Windows, macOS e Linux, bem como smartphones e tablets com iOS, Windows Phone e Android. As músicas podem ser navegadas ou pesquisadas por artista, álbum, gênero, lista de reprodução ou gravadora. Usuários podem criar, editar ou compartilhar playlists, compartilhar faixas em redes sociais ou fazer playlists com outros usuários. O Spotify fornece acesso a mais de 30 milhões de músicas. Em julho de 2019, contava mais de 232 milhões de usuários ativos, incluindo 108 milhões de assinantes pagantes.

O arquivo `data.csv` contém mais de 175.000 músicas coletadas da plataforma Spotify Web API, que podem ser agrupados por artista, ano ou gênero. Tal arquivo deve ser copiado para a pasta `/tmp/`. Quando reiniciamos o Linux, ele normalmente apaga os arquivos existentes na pasta `/tmp/`.

Estruturas Sequenciais

1. **Lista com Alocação Sequencial em Java:** Crie uma Lista de registros baseada na de inteiros vista na sala de aula. Sua lista deve conter todos os atributos e métodos existentes na lista de inteiros, contudo, adaptados para a classe *Música*. Lembre-se que, na verdade, temos uma lista de ponteiros (ou referências) e cada um deles aponta para um registro. Neste exercício, faremos inserções, remoções e mostraremos os elementos de nossa lista.

Os métodos de inserir e remover devem operar conforme descrito a seguir, respeitando parâmetros e retornos. Primeiro, o `void inserirInicio(Música musica)` insere um registro na primeira posição da Lista e remaneja os demais. Segundo, o `void inserir(Musica musica, int posição)` insere um registro na posição p da Lista, onde $p < n$ e n é o número de registros cadastrados. Em seguida, esse método remaneja os demais registros. O `void inserirFim(Musica musica)` insere um registro na última posição da Lista. O `Musica removerInicio()` remove e retorna o primeiro registro cadastrado na Lista e remaneja os demais. O `Musica remover(int posição)` remove e retorna o registro cadastrado na p -ésima posição da Lista e remaneja os demais. O `Musica removerFim()` remove e retorna o último registro cadastrado na lista.

A entrada padrão é composta por duas partes. A primeira é igual a entrada da primeira questão. As demais linhas correspondem a segunda parte. A primeira linha da segunda parte tem um número inteiro n indicando a quantidade de registros a serem inseridos/removidos. Nas próximas n linhas, tem-se n comandos de inserção/remoção a serem processados neste exercício. Cada uma dessas linhas tem uma palavra de comando: **I** inserir no início, **I*** inserir em qualquer posição, **IF** inserir no fim, **RI** remover no início, **R*** remover em qualquer posição e **RF** remover no fim. No caso dos comandos de inserir, temos também o nome do arquivo que contém o registro a ser inserido. No caso dos comandos de “em qualquer posição”, temos também esse nome. No **Inserir**, a posição fica imediatamente após a palavra de comando. A saída padrão tem uma linha para cada registro removido sendo que essa informação será constituída pela palavra “(R)” e o atributo **nome**. No final, a saída mostra os atributos relativos a cada registro cadastrado na lista após as operações de inserção e remoção.

2. **Pilha com Alocação Sequencial em Java:** Crie uma Pilha de registros baseada na pilha de inteiros vista na sala de aula. Neste exercício, faremos inserções, remoções e mostraremos os elementos de nossa pilha. A entrada e a saída padrão serão como as da questão anterior, contudo, teremos apenas os comandos **I** para inserir na pilha (empilhar) e **R** para remover (desempilhar).
3. **Fila Circular com Alocação Sequencial em Java:** Crie uma classe *Fila Circular* de *Musica*. Essa fila deve ter tamanho cinco. Em seguida, faça um programa que leia vários registros e insira seus atributos na fila. Quando o programa tiver que inserir um registro e a fila estiver cheia, antes, ele deve fazer uma remoção. A entrada padrão será igual à da questão anterior. A saída

padrão será um número inteiro para cada registro inserido na fila. Esse número corresponde à média **arredondada da duration** dos registros contidos na fila após cada inserção. Além disso, para cada registro removido da fila, a saída padrão também apresenta a palavra “(R)” e alguns atributos desse registro. Por último, a saída padrão mostra os registros existentes na fila seguindo o padrão da questão anterior.

Estruturas Flexíveis

4. **Lista com Alocação Flexível em Java:** Refazer a Questão 1 “Lista com Alocação Sequencial” usando lista dinâmica simples em Java.
5. **Pilha com Alocação Flexível em Java:** Refazer a Questão 2 “Pilha com Alocação Sequencial” em Java.
6. **Fila com Alocação Flexível em C:** Refazer a Questão 3 “Fila Circular com Alocação Sequencial” em C. Lembre-se que essa fila terá tamanho máximo igual a cinco.
7. **Pilha com Alocação Flexível em C:** Refaça a questão 3 deste TP com alocação flexível na linguagem C.
8. **Quicksort com LISTA DINÂMICA DUPLAMENTE ENCADEADA em C:** Refaça a Questão “Quicksort” 10 do Trabalho Prático II - com lista duplamente encadeada. O nome do arquivo de log será matrícula_quicksort2.txt.
9. **Quicksort com LISTA DINÂMICA DUPLAMENTE ENCADEADA em Java:** Refaça a questão anterior deste TP na linguagem Java.
10. **Matriz Dinâmica em Java:** Complete o código da classe matriz dinâmica visto na sala de aula. A primeira tarefa consiste em, no construtor da classe Matriz, dados os números de linha e coluna, fazer as devidas alocações de células. As demais tarefas são as implementações dos métodos Matriz soma(Matriz), Matriz multiplicacao(Matriz), void mostrarDiagonalPrincipal() e void mostrarDiagonalSecundaria(). A entrada padrão é composta por vários casos de teste sendo que o número de casos é um inteiro contido na primeira linha da entrada. Em seguida, temos cada um dos casos de teste. Cada caso é composto por duas matrizes. Para cada caso de teste, temos que suas duas primeiras linhas contêm um número inteiro cada representando os números de linhas e de colunas da primeira matriz, respectivamente. Em seguida, temos os elementos da primeira matriz que estão representados nas próximas l linhas onde l é o número de linhas dessa matriz. Cada uma dessas linhas têm c colunas onde c é o número de colunas dessa matriz. Nas duas linhas seguintes, temos os números de linhas e colunas da segunda matriz do caso de teste. As $l2$ linhas seguintes têm $c2$ colunas contendo os elementos da segunda matriz. $l2$ e $c2$ correspondem aos números de linhas e colunas da segunda matriz do caso de

teste, respectivamente. A saída padrão contém várias linhas para cada caso de teste. As duas primeiras linhas de saída de um caso de teste correspondem às diagonais principal e secundária da primeira matriz, respectivamente. As demais ls linhas de um caso de teste correspondem as linhas matriz obtida pela soma das duas matrizes do caso de teste sendo que essas linhas contêm cs colunas referentes às colunas da matriz de soma. Da mesma forma, as linhas seguintes do caso teste contêm lm linhas com cm colunas representando os elementos da matriz de multiplicação onde lm e cm são os números de linhas e colunas da matriz de multiplicação.