



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 IV

Regras Básicas

- extends Trabalho Prático 03
- Fique atento ao Charset dos arquivos de entrada e saída.

Observação:

Nas questões de árvore, utilizamos o mostrar pré.

Não será necessário implementar a opção de remoção nas TADs abaixo.



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.

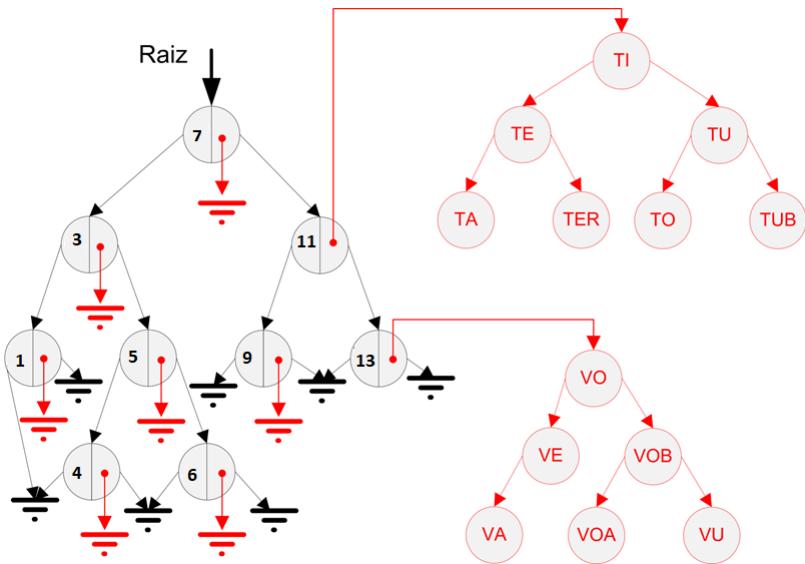
O arquivo **data.csv** contém mais de 175.000 músicas coletadas da plataforma Spotify Web API, que podem ser agrupados por artista, ano ou gênero. Tal arquivo deve ser copiado para a pasta `/tmp/`. Quando reiniciamos o Linux, ele normalmente apaga os arquivos existentes na pasta `/tmp/`.

Árvores

Observação: ATENÇÃO para os algoritmos de árvore que já estão implementados no [Github!](#)

1. **Árvore Binária em Java:** Crie uma Árvore Binária, fazendo inserções de registros conforme a entrada padrão. A chave de pesquisa é o atributo **id da música**. Não insira um elemento se sua chave estiver na árvore. Em seguida, pesquise se alguns registros estão cadastrados na Árvore, mostrando seus respectivos caminhos de pesquisa. A entrada padrão é igual a da questão de “Pesquisa Sequencial”. A saída padrão é composta por várias linhas, uma para cada pesquisa. Cada linha é composta pelo caminho ou sequência de ponteiros (**raiz, esq ou dir**) utilizados na pesquisa e, no final, pelas palavras SIM ou NAO. Além disso, crie um arquivo de log na pasta corrente com o nome matrícula_arvoreBinaria.txt com uma única linha contendo sua matrícula, tempo de execução do seu algoritmo e número de comparações. Todas as informações do arquivo de log devem ser separadas por uma tabulação '\t'.
2. **Árvore Binária de Árvore Binárias em Java:** Refaça a questão anterior, contudo, considerando a estrutura de árvore de árvore. Nessa estrutura, temos uma árvore binária tradicional na qual cada nó tem um ponteiro para outra árvore binária. Graficamente, a primeira árvore está no plano xy e a árvore de seus nós pode ser imaginada no espaço tridimensional. Temos dois tipos de nós. O primeiro tem um número inteiro como chave, os ponteiros esq e dir (ambos para nós do primeiro tipo) e um ponteiro para nós do segundo tipo. O outro nó tem uma String como chave e os ponteiros esq e dir (ambos para nós do segundo tipo). A chave de pesquisa da primeira árvore é o atributo **duration mod 15** e, da outra, é o atributo **id da musica**. Conforme a figura abaixo.

Destaca-se que nossa pesquisa faz um “mostrar” na primeira árvore e um “mostrar” na segunda. Faremos um “mostrar” na primeira árvore porque ela é organizada pelo **duration mod 15**, permitindo que o valor desejado esteja na segunda árvore de qualquer um de seus nós. Faremos o “mostrar” na segunda porque ela é organizada pelo atributo **id da musica**. Antes de inserir qualquer elemento, crie a primeira árvore, inserindo todos seus nós e respeitando a ordem **7, 3, 11, 1, 5, 9, 13, 0, 2, 4, 6, 8, 10, 12 e 14**. O arquivo de log será matrícula_arvoreArvore.txt.



3. **Árvore AVL em C:** Refaça a primeira questão deste trabalho com Árvore AVL em C. O nome do arquivo de log será matrícula_avl.txt.
4. **Árvore Alvinegra em Java:** Refaça a primeira questão deste trabalho com Árvore Alvinegra. O nome do arquivo de log será matrícula_avinegra.txt.
5. **Tree sort em Java:** Um tipo de árvore que é um algoritmo de ordenação. Este constrói uma árvore de busca binária dos elementos a serem classificados, em seguida, percorre a árvore (em ordem) para que os elementos sair em ordem de classificação. Efetue a ordenação das músicas pelo **id** usando o algoritmo **Tree sort**. Ao seu arquivo de log, contabilize a quantidade de comparações efetuadas para inserção de todas as músicas na árvore. O nome do arquivo de log será matrícula_treesort.txt.