



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.

Base de Dados

Star Wars é uma franquia do tipo space opera estadunidense criada pelo cineasta George Lucas que conta com uma série de oito filmes de fantasia científica e dois spin-offs. O primeiro filme foi lançado apenas com o título Star Wars em 25 de maio de 1977, e tornou-se um fenômeno mundial inesperado de cultura popular,



sendo responsável pelo início da “era dos blockbusters”: Super produções cinematográficas que fazem sucesso nas bilheterias e viram franquias com brinquedos, jogos, livros, etc. Foi seguido por duas sequências, The Empire Strikes Back e Return of the Jedi, lançadas com intervalos de três anos, formando a trilogia original. Esta primeira trilogia segue o trio icônico: Luke Skywalker, Han Solo e Princesa Leia, que luta na Aliança Rebelde para derrubar o tirano Império Galáctico; paralelamente ocorre a jornada de Luke para se tornar um cavaleiro Jedi e a luta contra Darth Vader, um ex-Jedi que sucumbiu ao Lado Sombrio da Força e ao Imperador.

Depois de 16 anos sem filmes novos lançados, uma nova trilogia chamada de prequela começou em 1999 com The Phantom Menace. Esta volta no tempo para contar como Anakin Skywalker se transformou em Darth Vader e acompanha a queda da Ordem Jedi e da República Galáctica substituída pelo Império. Sendo também lançada com intervalos de três anos, com o último lançado em 2005. As reações à trilogia original foram extremamente positivas, enquanto a trilogia prequela

recebeu reações mistas tanto da crítica especializada como do público. Mesmo assim, todos os filmes foram bem sucedidos nas bilheterias e receberam indicações ou ganharam prêmios no Óscar.

O arquivo **personagens.zip** contém um conjunto de dados de personagens da franquia Star Wars, oriundo de respostas para requisições que foram feitas a API `swapi.co/` em 01/02/2020. Estes arquivos **sofreram algumas adaptações** para ser utilizado neste e nos próximos trabalhos práticos. Tal arquivo deve ser copiado para a pasta `/tmp/`. **Quando reiniciamos o Linux, ele normalmente apaga os arquivos existentes na pasta `/tmp/`.**

Árvores e *Hash*

1. **Árvore Binária:** Crie uma Árvore Binária, fazendo inserções de registros conforme a entrada padrão. A chave de pesquisa é o atributo **Nome do Personagem**. 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 NÃO. 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:** 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 **altura mod 15** e, da outra, é o atributo **Nome do Personagem**.

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 **altura 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 **Nome do Personagem**. Antes de inserir qualquer elemento, crie a primeira árvore, inserindo todos seus nós e respeitando a ordem 7, 3, 11, 1, 5, 9, 12, 0, 2, 4, 6, 8, 10, 13 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:** Refaça a primeira questão deste trabalho com Árvore Alvinegra. O nome do arquivo de log será `matrícula_avinegra.txt`.
5. **Tabela *Hash* Direta com Reserva:** Refaça a primeira questão deste trabalho com Tabela *Hash* Direta com Reserva. A função de transformação será **`altura mod tamTab`** onde `tamTab` (tamanho da tabela) é 21. A área de reserva tem tamanho 9, fazendo com que o tamanho total da tabela seja igual a 30. A saída padrão será a posição de cada elemento procurado na tabela (na *hash* ou na área de reserva). Se o elemento procurado não estiver na tabela, escreva a palavra NÃO. Além disso, o nome do arquivo de log será `matrícula_hashReserva.txt`.
6. **Tabela *Hash* Direta com Rehash:** Refaça a questão anterior com Tabela *Hash* Direta com *Rehash*. A primeira função de transformação será **`altura mod tamTab`** onde `tamTab` (tamanho da tabela) é 25 e a outra, **`(altura + 1) mod tamTab`**. O nome do arquivo de log será `matrícula_hashRehash.txt`.
7. **Tabela *Hash* Indireta com Lista Simples em C:** Refaça a questão anterior com Tabela *Hash* Indireta com Lista Simples. A função de transformação será **`altura mod tamTab`** onde `tamTab` (tamanho da tabela) é 25. O nome do arquivo de log será `matrícula_hashIndireta.txt`.
8. **Árvore *Trie*:** Crie duas árvores do tipo *trie* com os nomes dos Personagens, fazendo inserções de registros conforme a entrada padrão. Não insira um elemento se sua chave estiver na árvore. Em seguida, faça o merge das duas árvores. Na árvore resultante (que não tem nomes repetidos), pesquise alguns nomes. A entrada padrão é igual a da questão de “Pesquisa Sequencial”. Entretanto, após a primeira ocorrência da palavra FIM, temos outros Personagens que devem ser inseridos na segunda árvore. A saída padrão é composta por várias linhas, uma para cada pesquisa resultando nas palavras SIM ou NÃO. Além disso, crie um arquivo de log na pasta corrente com o nome `matrícula_arvoreTrie.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'`.