

1º Trabalho de Implementação de Algoritmos e Estruturas de Dados (2023-2):

Referência cruzada de palavras usando ABPs (ou AVLs)

Descrição do problema:

Um conjunto de palavras pode ter um processo de busca por palavra otimizado se o armazenamento das palavras for realizado em uma árvore de pesquisa, ABP (ou AVL). A árvore de pesquisa vai reduzir o número de comparações realizadas na busca dos dados, uma vez que os mesmos foram inseridos de forma ordenada permitindo uma pesquisa binária.

Para um número grande de palavras, essa otimização nas buscas pode melhorar ainda mais se forem utilizadas várias ABPs (ou AVLs), cada uma para armazenar as palavras iniciadas por uma letra do alfabeto. **O objetivo deste trabalho é implementar uma estrutura ordenada de acesso às palavras organizadas por uma referência cruzada, utilizando árvores binárias de pesquisa (ABPs ou AVLs).** O A estrutura ordenada de acesso é organizada através de um conjunto de ABPs (ou AVLs), uma ABP (ou AVL) por letra do alfabeto, tendo como critério de ordenação a ordem alfabética das palavras.

Organização: trabalho individual ou em duplas

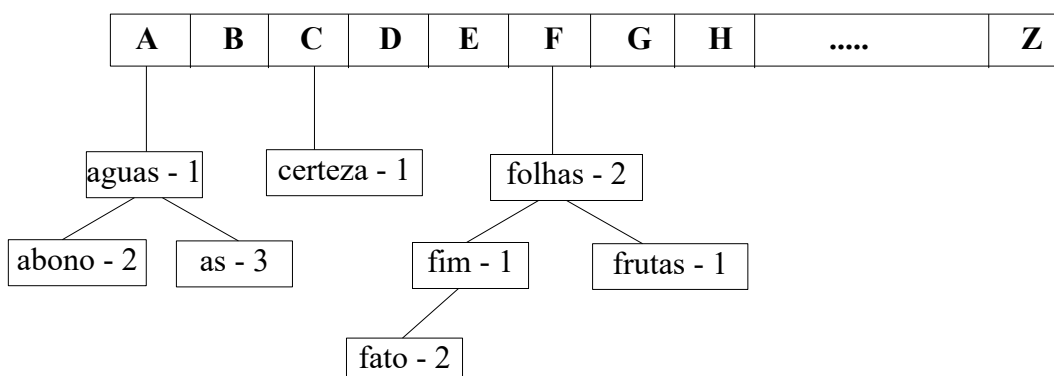
Apresentação:

A apresentação do trabalho será realizada para a professora. Para a apresentação, se o trabalho for feito em dupla ambos os participantes devem estar presentes e participar da explicação sobre a implementação realizada.

Até o dia da apresentação, fazer *upload* do trabalho na tarefa correspondente no AVA do trabalho: código-fonte da implementação.

Representação:

1) uma lista de letras (como uma tabela *hash* por encadeamento aberto, representada por vetor ou lista encadeada), sendo que cada letra (cada item da tabela) referencia a ABP (AVL) das palavras que começam com aquela letra:



A **lista de letras** (pode ser uma tabela *hash*), pode ser implementada como vetor ou como lista encadeada, ou como uma ABP (ou AVL). Se for uma lista encadeada ou ABP (AVL), pode conter apenas as letras que tem palavras associadas.

2) ABPs (ou AVLs) para as palavras que começam por uma letra. Cada ABP (AVL) é composta por nodos que contêm as seguintes informações:

- palavra
- **número de ocorrências** daquela palavra no texto (1 é o menor número de ocorrências)
- referências para filhos da esquerda e da direita

O texto a ser submetido à referência cruzada pode estar originalmente num arquivo (arquivo *texto.txt*), ou pode ser informado pelo usuário (texto inteiro, ou palavra por palavra).

Implementação:

Pode ser implementado em C, C++, C#, Java ou Python, desde que as árvores sejam implementadas no programa (referências entre os nodos manipuladas explicitamente pelo programa, com referência a filho da esquerda e da direita). Não pode usar classe, método ou função pronta da linguagem nem para o *hash* nem para listas encadeadas ou árvores, nem utilizar estruturas de listas específicas como dicionários.

Cada nodo deverá conter as informações acima enumeradas (a palavra, nro de ocorrências, referências), mas pode conter outras informações necessárias para a realização das operações.

A interface pode ser bem simples (textual), o importante é que as operações funcionem.

Haverá um bônus na nota em caso de implementação de AVL no lugar de ABP (com ABP a nota será sobre 9,0, com AVL sobre 10,0).

Algumas operações básicas a serem oferecidas pelo programa:

- INSERE: permite a inserção de uma palavra do texto na estrutura. Palavras não podem ser duplicadas. Ao inserir uma palavra, seu número de ocorrências é inserido também.
- CONSULTA: consulta uma palavra na estrutura, informando seu número de ocorrências.
- REMOVE: remove uma palavra da estrutura (remove o nodo com a palavra), mantendo a ABP na qual ela estava situada ordenada.
- CONTA: retorna o número total de palavras da estrutura.
- CONTA_OCORRÊNCIAS: retorna o número total de ocorrências das palavras da estrutura (soma os contadores)
- EXIBE_PALAVRAS: exibe a lista das palavras da estrutura (só as palavras, ou com os respectivos contadores, em ordem alfabética (de A a Z) ou em ordem alfabética inversa (de Z a A) de palavras.
- EXIBE_PALAVRAS_LETRA: exibe a lista das palavras da estrutura iniciadas por uma determinada letra (só as palavras ou com contadores), em ordem alfabética (de A a Z) ou em ordem alfabética inversa (de Z a A) de palavras.
- MAIOR_NRO_OCORRENCIAS: exibe a palavra (ou as palavras) com maior número de ocorrências.
- UMA_OCORRENCIA: exibe, em ordem alfabética, as palavras que só tem uma ocorrência.
- outras operações que você defina (pelo menos uma operação extra).