Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM
Disciplina: Estrutura de Dados I – BCC 202
Alunos: Felipe Fontenele e Mateus Lana
Matrículas: 15.1.4331 e 15.1.4340

DOCUMENTAÇÃO

Implementação

São feitas diversas operações ao longo do código, algumas mais importantes e outras menos. Vamos explicar brevemente as mais essenciais.

Primeiramente o programa abre um **arquivo.txt** extraindo todas as palavras e a quantidade das mesmas. Em seguida, as palavras são colocadas em um vetor temporário e seus dados são padronizados (retirando sinais de pontuação e deixando todas as letras minúsculas, por exemplo).

Executamos todas as funções ou procedimentos necessários para a implementação das TADs de Pesquisa Binária, de Árvore de Busca Binária e de Árvore AVL. Posteriormente as palavras são repassadas para os seguintes procedimentos: TBinario_Inserir, Arvore_Inserir e InsereAVL.

- **TBinario_Inserir:** Este procedimento realiza a inserção das palavras na estrutura da pesquisa binária, garantindo que as mesmas fiquem ordenadas devido a chamada recursiva da função TBinario_Pesquisar, que retorna o índice do vetor cuja palavra deverá ser inserida. Em seguida, o procedimento realiza os deslocamentos necessários mediante o índice recebido, assegurando que o custo da operação de inserir seja linear, inclusive no seu pior caso.
- **TBinario_Pesquisar:** Esta função permite que a complexidade de encontrar o índice certo do vetor para a inserção da palavra seja *log n*, até no pior caso. Ela também verifica a repetição de palavras, aumentando seus números aparições a cada vez que encontra palavras iguais.
- **Arvore_Inserir:** O que esta função faz é inserir as palavras na árvore, primeiro ela procura o lugar na mesma e se o dado não estiver presente, insere-o. Se o dado já estiver presente ela incrementa o contador do número de aparições de uma mesma palavra.
- **InsereAVL:** Esta função faz inserções de palavras na árvore, assim como a anteriormente descrita. Porém, além dela inserir as palavras nos lugares corretos e verificar as repetições, ela também realiza o balanceamento da árvore através de outras várias funções.

O tempo é calculado para cada um dos processos acima explicados, para isso foi necessário incluir a biblioteca *ctime*. Além disso, também é calculado o número de comparações realizadas em cada método, utilizando um contador para cada um deles.

Ao final imprimimos todas as palavras em ordem e seus respectivos números de aparições, bem como o tempo de execução do método escolhido e seus números de comparações. É importante ressaltar que para imprimir os resultados dos métodos feitos com árvores, é necessário utilizar o caminhamento central nas mesmas, para garantir a impressão ordenada das palavras. Isso não é necessário para o método de pesquisa binária, já que este, como foi dito anteriormente, garante que os dados fiquem ordenados.

Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM
Disciplina: Estrutura de Dados I – BCC 202
Alunos: Felipe Fontenele e Mateus Lana
Matrículas: 15.1.4331 e 15.1.4340

Análise

Os resultados foram satisfatórios, visto que na Pesquisa Binária, conseguimos obter os resultados esperados, com uma implementação bem otimizada. Em certos casos a Pesquisa Binária é mais rápida até que as Árvores.

Em relação as Árvores, observamos que a Árvore AVL, devido aos balanceamentos que realiza, apresenta um número de comparações inferior ao da Árvore Binária.

Conclusão

Ao realizar este trabalho pudemos aprimorar e compreender melhor a manipulação das estruturas de Pesquisa Binária, Árvore Binária e Árvore AVL, trazendo impactos positivos. Também possibilitou que evoluíssemos no que se diz respeito a manipulação de arquivos de texto e de *strings*.

Tivemos um pouco de dificuldade na parte da Pesquisa Binária, mais especificamente na operação de inserção, que tinha como objetivo inserir mantendo os dados ordenados. Porém conseguimos contornar com sucesso os eventuais problemas.