TRABALHO PRÁTICO II
RELATÓRIO

Luiz Eduardo Pereira

0021619

Instituto Federal de Minas Gerais, Formiga, MG.

INTRODUÇÃO

Este trabalho tem como objetivo a implementação e a verificação de desempenho de diferentes estruturas de dados para a construção de um índice remissivo.

IMPLEMENTAÇÃO

Este trabalho foi desenvolvido na linguagem C e foi dividido em 4 estruturas: Hash Externa, Lista Encadeada, Árvore Binaria de Busca não balanceada e Árvore AVL.

Hash

A Hash utilizada neste trabalho foi a de melhor resultado no trabalho I. Como eu não fiz o trabalho, me foi passado que a melhor hash é a hash de encadeamento externo, com 50% de tamanho em relação à quantidade de palavras chaves. Está hash utiliza uma lista encadeada em cada posição. E em cada campo da lista, existe outra lista que contem o índice remissivo.

Lista

A Lista Encadeada foi implementada utilizando o mesmo código da hash, com uma única diferença, o tamanho da hash sempre é 1, transformando-a em uma lista.

Árvore Binaria de Busca

A Árvore Binaria não possui nenhuma função que a torne balanceada. Portanto a primeira palavra chave que entrar nela, será o nó raiz. Sendo assim, todas as palavras menores vão para a esquerda e as maiores, para a direita.

Árvore AVL

A Árvore AVL é basicamente uma árvore binaria de busca, mas em seu registro possui um campo a mais, a altura. Este campo é utilizado para a verificação da altura da árvore para saber se ela precisa ser balanceada. A Árvore AVL também possui funções para fazer a rotação de nós da árvore. Isto é o que define a arvore AVL, com essas rotações, garantimos que a arvore está sempre balanceada, sendo possível fazer qualquer consulta em O(Log N).

TESTES

Para testar a eficiência de cada estrutura foi utilizado os testes disponibilizados pelo professor. Só é possível ver a diferença entre elas em casos em que o arquivo é muito grande, por isso, dos testes 1 a 4 o resultado foi praticamente o mesmo.

A Tabela a seguir contém os tempos de cada estrutura e os testes correspondentes. Os valores estão em milissegundos.

Teste	Hash	Lista	Árvore BB	Árvore AVL
1	7	7	7	7
2	4	4	4	4
3	7	6	6	6
4	25	27	25	25
5	710	759	614	496

Como é possível ver, a Árvore AVL conseguiu executar o teste 5 em menos tempo, mais para definir se essa é a estrutura recomendada para determinado problema é necessário levar em conta: o programa possuirá mais inserções ou mais pesquisas? Por ser sempre balanceada, a árvore possui um tempo de pesquisa mais rápido O (Log n), mas pode perder eficiência se o programa fizer mais inserções, porque rebalancear a árvore possui um custo.

Então, para fazer a escolha entre Hash ou Árvore, é necessário saber como o programa funcionará para escolher a melhor estrutura. No caso do índice remissivo, onde as inserções acontecem apenas no inicio, a Árvore AVL é mais recomendada.

CONCLUSÃO

Este trabalho exemplifica uma utilização real das estruturas propostas, permitindo que verifiquemos se o que foi estudado em sala de aula realmente acontece na pratica. Desse modo, podemos ver o funcionamento de cada estrutura, o que ela possui de bom e o que não é tão interessante.