

UNIVERSIDADE FEDERAL DO MATO GROSSO *CAMPUS* ARAGUAIA

**ANTHONY MUNIZ PRADO DE OLIVEIRA, VINICIUS SPANHOL
FERRARI**

TRABALHO 3 – Árvores Binárias Balanceadas

BARRA DO GARÇAS - MT

2021

INTRODUÇÃO

O objetivo deste trabalho é exercitar a implementação e manipulação de algoritmos de árvores binárias e balanceadas, visando verificar sua funcionalidade, e sua otimização em diferentes cenários onde se deve a grande quantidade de inserções de dados nas mesmas com a linguagem de programação C. Também foi trabalhado a manipulação de arquivos.

Foi utilizado o editor de textos Visual Studio Code para a programação, dos algoritmos responsáveis pelos dados que serão apresentados neste trabalho.

DESENVOLVIMENTO

O desenvolvimento consiste de 5 etapas onde 3 serão tratadas neste trabalho:

Etapa 1:

Implementamos o que foi pedido no arquivo “Etapa_01.c” onde foi necessário elaborar um algoritmo que gerasse um arquivo contendo um conjunto de valores inteiros, definidos aleatoriamente, não repetidos, no intervalo entre 1 e 100.000, sendo um valor por linha.

Etapa 2:

Nesta etapa devemos implementar as funções necessárias para a construção e manipulação de Árvore Binária (Clássica), Árvore AVL e Árvore Rubro-Negra. Por conta disso, criamos 3 algoritmos, um para cada árvore e realizamos a sua execução e chamadas de suas funções em um só algoritmo “Main.c”: que englobasse os resultados dos mesmos em um só lugar para facilitar a análise e coleta dos dados para avaliar sua otimização e eficiência.

Na implementação da árvore Rubro-Negra houveram dificuldades, então preferimos utilizar outro modelo de árvore Rubro-Negra que é equivalente.

Etapa 3:

Nesta etapa utilizamos nosso algoritmo “Main” para contabilizar o total de rotações executadas e avaliar a altura das árvores resultantes após a inserção dos valores informados pelo arquivo de entrada. O processo de avaliação constituiu-se em executar um mesmo arquivo de entrada para cada tipo de árvore binária (Binária Clássica, AVL e Rubro-Negra) e então comparar as diferenças em relação à altura e o número de rotações executadas em cada tipo de árvore.

Com o auxílio do algoritmo desenvolvido na etapa 1, foram gerados 3 arquivos de entradas com 5.000 números e 3 arquivos com 20.000 números e então utilizados nas árvores para a realização do cálculo e média entre os resultados individuais de cada árvore da maneira como foi descrito acima.

RESULTADOS

Tabela 1:

MÉDIA	Altura C/ 5000 Valores	Altura C/ 20000 Valores	Rotações C/ 5000 Valores	Rotações 20000 C/ Valores
CLÁSSICA	28	34	0	0
AVL	13	15	4704	23404
RUBRO-NEGRO	15	17	1950	7761

CONSIDERAÇÕES FINAIS

Ao fim das execuções e comparações dos valores finais fornecidos pelo algoritmo, nós acabamos por chegar nas seguintes conclusões:

O algoritmo da árvore AVL por realizar um grande número de rotações como pode ser visto ao analisar a tabela 1, gera uma árvore muito menor em altura e mais otimizada em tamanho do que as demais, porém seu número de rotações se torna extremamente exagerado, tendo que realizar em algumas situações mais rotações que o número de valores fornecidos ao algoritmo, o que tornaria sua execução muito pesada se executada com uma gama muito maior de dados.

Entretanto a árvore Rubro-Negra, consegue resultados muito semelhantes, com um número muito menor de rotações, o que otimiza mais ainda a execução, apesar de não ser menor tamanho é a mais balanceada (em aspectos gerais) de todas.

Já a árvore binária clássica por não realizar rotações se torna a mais simples de ser executada, porém se torna muito grande e nada otimizada em tamanho, sendo demorado sua inserção que em todas as execuções percorrerá um caminho muito grande dependendo da situação diferente da AVL e Rubro-Negro gerando assim uma árvore capaz de ter uma altura igual ao número de valores inseridos, tornando ela a pior na prática quando for necessário fazer uma busca ou uma exclusão.