RELATÓRIO SOBRE O ALGORITMO EM C QUE IMPLEMENTA UMA ÁRVORE BINÁRIA DE BUSCA BALANCEADA

UNIVERSIDADE FEDERAL DO PARANÁ

Natã Abraão Serafini da Rocha GRR20200183

Victor Ribeiro Garcia GRR20203954

**Resumo.** Esse breve relatório descreve o processo criativo e as ideias e procedimentos usados para desenvolver um algoritmo em linguagem C que manipula dados e as estruturas de uma AVL (árvore binária de busca balanceada), com a intenção testar e avaliar as habilidades e conhecimentos dos alunos da matéria CI1057 – Algoritmos e estrutura de dados III.

**Objetivo do trabalho.**

O trabalho final deve apresentar um algoritmo que recebe uma entrada com a operação que deve ser realizada na AVL e qual o número deve-se usar, após a leitura de todas as entradas o programa deve entregar uma saída com os elementos da AVL e suas alturas, o algoritmo deve manter todas as propriedades da árvore binária de busca balanceada.

**Processo de criação e resultado.**

Uma árvore binaria de busca balanceada, também conhecida como AVL, é uma estrutura de dado que possuiu uma raiz e seus nós filhos, um a esquerda e outra à direita, não sendo necessário possuir ambos. Seus nodos filhos também podem ou não possuir filhos. Para qualquer nó na árvore inclusive a raiz os filhos da esquerda são menores e os da direita maiores que seu próprio valor. A nomenclatura de balanceada vem da característica de seu nós mais distantes da raiz possuírem entres si um fator de balanceamento entre -1 e 1, que é a diferença entre os as distâncias entre os nós e a raiz.

Para implementar de forma correta a AVL e manter suas propriedades foi criado a struct “no” que carrega em si as informações de cada nó, sendo elas a chave, os seus nós filhos e sua altura. O conjunto dos elementos de tipo “no” ligados uns aos outros por ponteiros que formam a AVL.

Na função main ficou estabelecido fazer a leitura do arquivo com as entradas, sendo elas do tipo inserção(i) ou exclusão(r) e o número a ser utilizado, e também verifica se o arquivo de entrada não está vazio.

Após ler as entradas e qual operação será realizada o algoritmo utilizará as funções necessárias para realizar a tarefa. A inclusão realizada na função “insere” é feita na “folha” da árvore e segue o padrão de alocar o filho menor a esquerda do nó pai e o maior a direta. Após inserir é checado o fator de balanceamento e caso necessário é feito a rotação para a árvore permanecer balanceada.

A exclusão é feita na função “exclui” e realiza a tarefa pelo sucessor, que busca o menor valor maior que o da chave a ser excluída e retira da árvore. Como realizado também na função de inserção é realizada o balanceamento caso necessário.

O balanceamento é o ato de realizar uma rotação ou à direita ou à esquerda dependendo de cada caso. As funções responsáveis por realizar esse comando são as “DirRot” e “EsqRot” que retornam o nó com seus dados atualizados após a rotação.

Foi preciso fazer algumas funções secundarias para realizar busca, verificar altura da árvore entre outras e assim garantir a saída esperada. O programa final recebe uma entrada no formato especificado e entrega a saída no formato esperado e seus dados corretos o que atende as especificações pedidas.