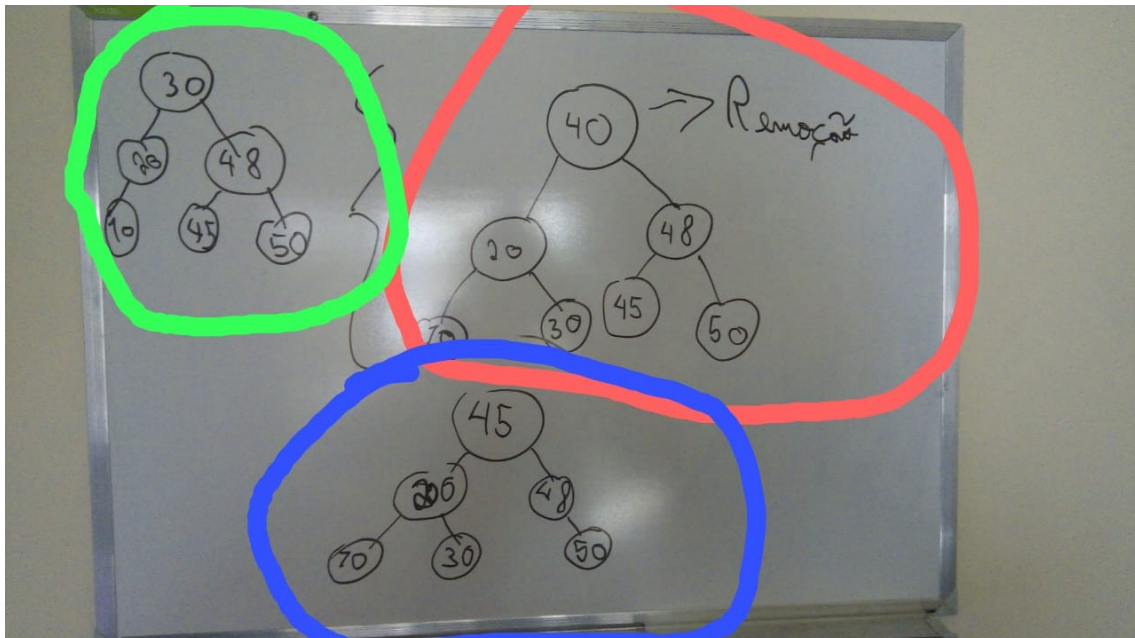


RELATÓRIO

A ideia central do trabalho seria desenvolver uma árvore AVL que recebe e processa instruções de inclusão/exclusão, imprimindo sua estrutura em arquivo. Para isso, considerando o tempo relativamente curto, fizemos pequenas reuniões para definir tarefas e prazos. Antes mesmo de implementar a árvore, focamos em tratar a entrada/saída padrão e ler os casos de teste, armazenando em um simples vetor de structs. Posteriormente, como forma de exercício, implementamos o trabalho utilizando a árvore BST, já que ainda não tínhamos tido a aula de AVL.

Na sequência, após a aula de AVL, reservamos alguns dias para estudar os algoritmos e então decidimos implementar o trabalho. Como já havíamos implementado com BST, bastava adaptar as estruturas e funções. Optamos pelo modo de leitura em ordem, utilizando o algoritmo do sucessor e o fator de balanceamento do nodo. Nesse momento surgiu nosso primeiro problema, já que as alturas impressas não conferiam com os casos de teste ofertados e o caso de teste 4 estava dando erro de segmentação. Como solução, começamos a desenhar a árvore em quadros e analisar como ela se comportava a cada nova instrução, também utilizamos um simulador de árvore AVL online para testes. Esses recursos foram muito importantes para identificar as rotações erradas no algoritmo e solucionar vários dos erros que estávamos encontrando.



Esboços do comportamento da AVL implementada.



Simulador de árvore AVL online utilizado nos testes.

Após muito esforço, desgaste e fracassos, tomamos uma escolha difícil: mudar a implementação. Passamos a utilizar o antecessor e a altura no nodo como fator de balanceamento da árvore. Para isso, tivemos que adaptar o código e suas respectivas funções. Decidimos dividir a função de exclusão em partes, para cada caso específico (nodo na folha, nodo com filho a esquerda ou direita, nodo com dois filhos). Além de desenvolver uma função de dupla rotação para facilitar balanceamento da árvore. Também foi mudado a estrutura do Nodo, onde substituímos o fator de balanceamento pela altura, o que acabou facilitando.

Como última tarefa, dividimos as funções da AVL em um arquivo separado com um cabeçalho por boas práticas, criamos o arquivo makefile para facilitar os testes e escrevemos este relatório a fim de informar nosso progresso com o trabalho.

O último desafio que encontramos foi a remoção errônea que nosso algoritmo estava fazendo para os testes 3 e 4. Após muitos esforços encontramos o problema, não era um problema de lógica mas sim no arquivo main.c, onde na hora que a função de remoção era chamada estávamos usando a função "exclui". Então o algoritmo estava removendo a raiz que era passada pra ele e não realizando uma busca de fato, com isso em vista ajustamos o código usando a função "removeNo".

Contudo, apesar dos desafios e problemas encontrados, conseguimos implementar a Árvore AVL. O processo foi fundamental para o entendimento do funcionamento do algoritmo, que significou um bom aprendizado na realização desse trabalho.