

NOME: Caroline Souza Camargo

REGRA DE CÓDIGO LEGÍVEL: Stanford Style Guide

EXERCÍCIO: Trabalho Prático 3

Preencher uma árvore binária de busca e depois que parar de adicionar nodos dizer se é AVL ou não

LINK PARA O GIT:

<https://github.com/Caroline-Camargo/AlgoritimoseEstruturaDeDados/tree/master/TrabalhoPratico3>

LINK PARA O VÍDEO:

<https://drive.google.com/file/d/1SoLSh2ZO-wZr9PyG3Q3MyJIMqqWyWFT/view?usp=sharing>

O QUE FUNCIONA: Após vários testes com várias entradas e testando com o debugador e o Dr Memory o programa funcionou corretamente

O QUE NÃO FUNCIONA:

Durante o período da aula foi esquecido de alguns detalhes e alguns erros foram encontrados no código

- Na função de testar a altura tinha um erro que no momento de calcular a altura da subárvore à direita eu estava chamando uma recursão para a esquerda em vez de ser uma recursão para direita
- Me esqueci de testar o caso em que o nodo passado para função era um endereço inválido (NULL) na função do fator de balanceamento
- No momento de verificar se o fator de balanceamento estava fora do intervalo, ou seja, ser menor que -1 e maior que 1, coloquei um && ao invés de um || (Nunca ia entrar no if, pois não tem como um número ser menor que -1 e maior que 1 ao mesmo tempo)
- Não deu tempo de implementar a função de Clear durante o período da aula, apesar de não ser um requisito necessário, pois ao encerrar o programa o correto é dar free para todos nodos alocados durante a execução.

Todos problemas foram resolvidos

LISTA DE TESTES:

Teste 1: 100, 400, 500

Não é AVL



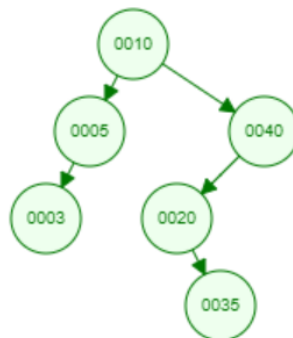
Teste 2: 400, 100, 500

É AVL



Teste 3: 10, 5, 40, 3, 20, 35

Não é AVL

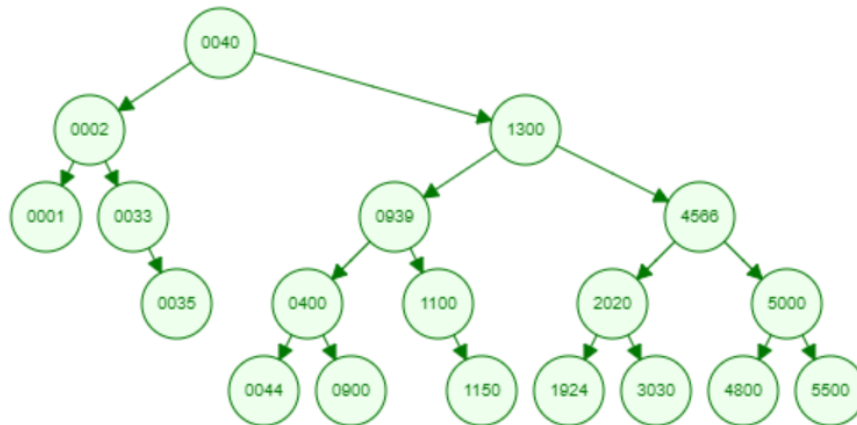


Teste 4: 10, 5, 40, 3, 2, 8, 20, 35, 70

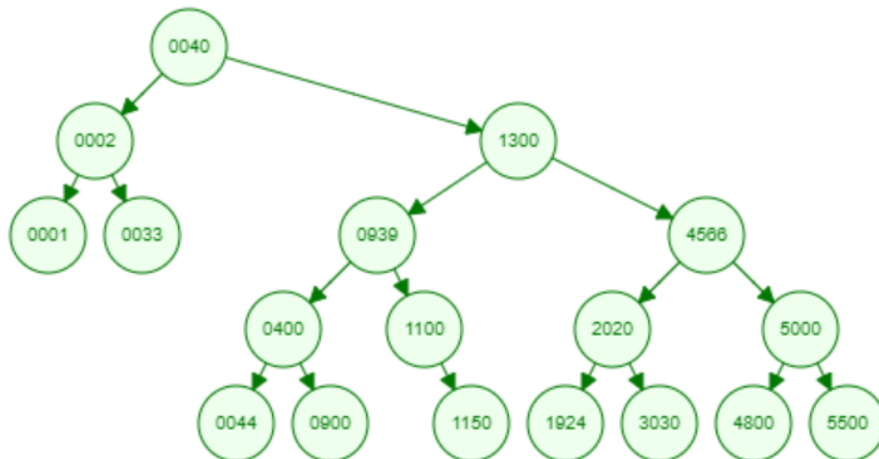
É AVL



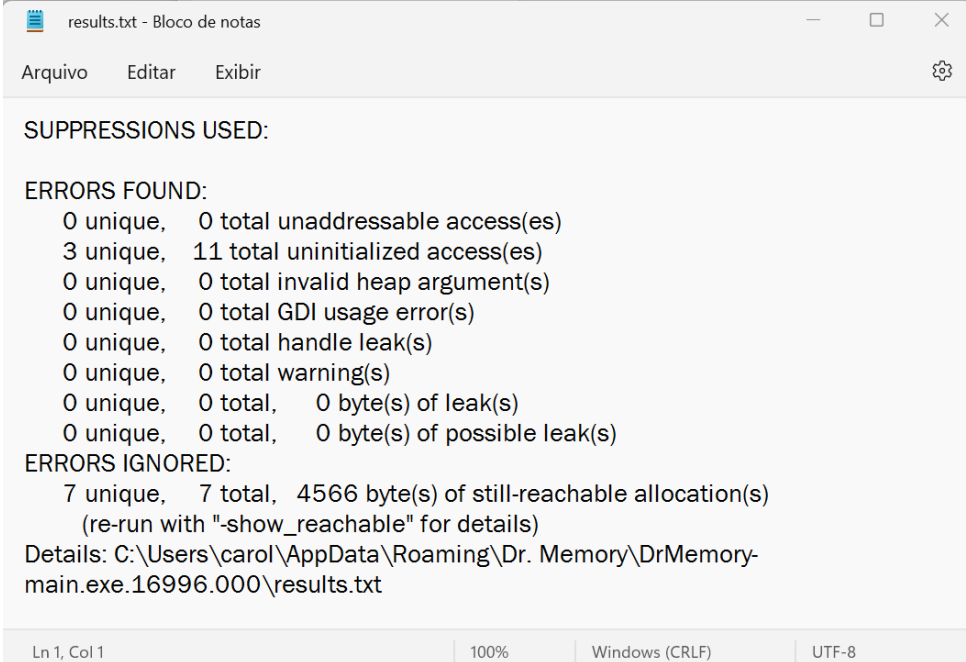
Teste 5: 40, 2, 1, 33, 1300, 939, 400, 1100, 400, 44, 900, 1100, 1150, 1300, 4566, 5000, 2020, 4800, 5500, 1924, 3030, 3030, 35
 É AVL



Teste 6: 40, 2, 1, 33, 1300, 939, 400, 1100, 400, 44, 900, 1100, 1150, 1300, 4566, 5000, 2020, 4800, 5500, 1924, 3030, 3030
 Não é AVL



Print do Dr Memory com o Exercício desenvolvido:



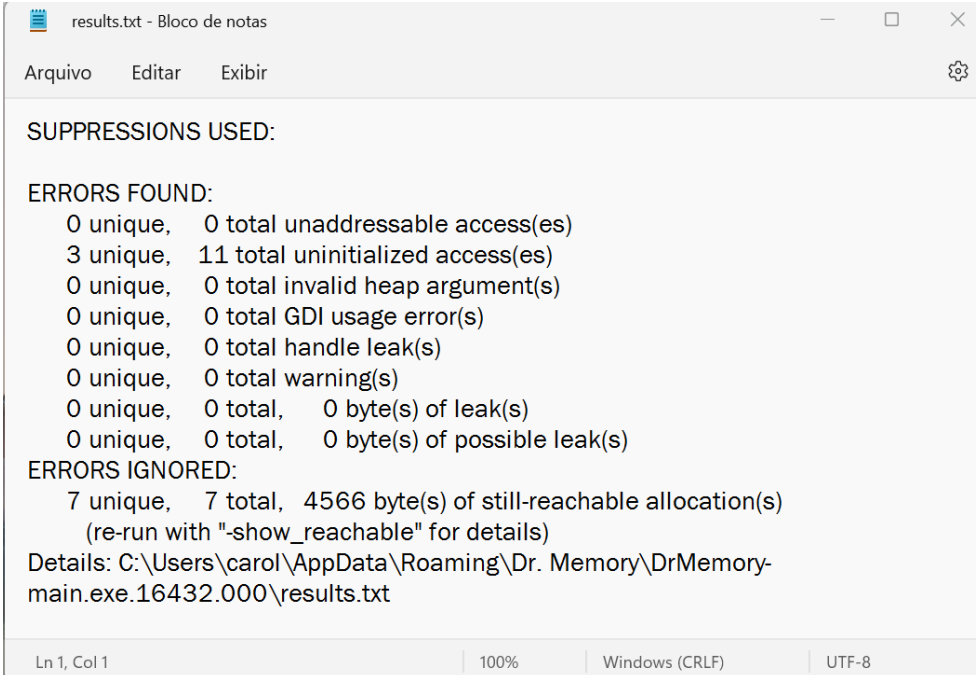
```
results.txt - Bloco de notas
Arquivo  Editar  Exibir

SUPPRESSIONS USED:

ERRORS FOUND:
    0 unique,  0 total unaddressable access(es)
    3 unique, 11 total uninitialized access(es)
    0 unique,  0 total invalid heap argument(s)
    0 unique,  0 total GDI usage error(s)
    0 unique,  0 total handle leak(s)
    0 unique,  0 total warning(s)
    0 unique,  0 total,  0 byte(s) of leak(s)
    0 unique,  0 total,  0 byte(s) of possible leak(s)
ERRORS IGNORED:
    7 unique,  7 total, 4566 byte(s) of still-reachable allocation(s)
    (re-run with "-show_reachable" for details)
Details: C:\Users\carol\AppData\Roaming\Dr. Memory\DrMemory-
main.exe.16996.000\results.txt

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8
```

Print do Dr Memory com um Olá mundo:



```
results.txt - Bloco de notas
Arquivo  Editar  Exibir

SUPPRESSIONS USED:

ERRORS FOUND:
    0 unique,  0 total unaddressable access(es)
    3 unique, 11 total uninitialized access(es)
    0 unique,  0 total invalid heap argument(s)
    0 unique,  0 total GDI usage error(s)
    0 unique,  0 total handle leak(s)
    0 unique,  0 total warning(s)
    0 unique,  0 total,  0 byte(s) of leak(s)
    0 unique,  0 total,  0 byte(s) of possible leak(s)
ERRORS IGNORED:
    7 unique,  7 total, 4566 byte(s) of still-reachable allocation(s)
    (re-run with "-show_reachable" for details)
Details: C:\Users\carol\AppData\Roaming\Dr. Memory\DrMemory-
main.exe.16432.000\results.txt

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8
```