

Instituto Federal de Educação, Ciência e Tecnologia do maranhão
Algoritmos e Estruturas de Dados II
Franciele Alves da Silva (20231SI0012)

Relatório

Main:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "AVLvsRB.h"
#include <locale.h>

#define OPERACOES 5000000

int main() { setlocale(LC_ALL, "");

    ArvAVL *arvoreAVL = cria_ArvAVL();
    ArvRB *arvoreRB = cria_ArvRB();

    srand(time(NULL));

    clock_t inicio, fim;
    double tempoAVL, tempoRB;

    // Inserção
    printf("Inserindo elementos nas árvores AVL e RB...\n");
    inicio = clock();
    for (int i = 0; i < OPERACOES; i++) {
        int valor = rand() % OPERACOES;
        insere_ArvAVL(arvoreAVL, valor);
    }
    fim = clock();
    tempoAVL = ((double)(fim - inicio)) / CLOCKS_PER_SEC;

    inicio = clock();
    for (int i = 0; i < OPERACOES; i++) {
        int valor = rand() % OPERACOES;
        insere_ArvRB(arvoreRB, valor);
    }
    fim = clock();
    tempoRB = ((double)(fim - inicio)) / CLOCKS_PER_SEC;

    printf("Tempo de inserção (AVL): %.6f segundos\n", tempoAVL);
    printf("Tempo de inserção (RB): %.6f segundos\n", tempoRB);

    // Busca
    printf("\nBuscando elementos nas árvores AVL e RB...\n");
    int encontradosAVL = 0, encontradosRB = 0;
    inicio = clock();
    for (int i = 0; i < OPERACOES; i++) {
        int valor = rand() % OPERACOES;
        encontradosAVL += consulta_ArvAVL(arvoreAVL, valor);
    }
    fim = clock();
    tempoAVL = ((double)(fim - inicio)) / CLOCKS_PER_SEC;

    inicio = clock();
    for (int i = 0; i < OPERACOES; i++) {
        int valor = rand() % OPERACOES;
        Node *resultado = busca_ArvRB(arvoreRB, valor);
        if (resultado != NULL)
            encontradosRB++;
    }
    fim = clock();
    tempoRB = ((double)(fim - inicio)) / CLOCKS_PER_SEC;

    printf("Tempo de busca (AVL): %.6f segundos\n", tempoAVL);
    printf("Tempo de busca (RB): %.6f segundos\n", tempoRB);

    printf("\nTotal de elementos encontrados (AVL): %d\n", encontradosAVL);
    printf("Total de elementos encontrados (RB): %d\n", encontradosRB);

    // Liberar memória
    liberar_ArvAVL(arvoreAVL);
    remove_ArvRB(arvoreRB);

    return 0;
}
```

Árvore AVL

Tempo gasto para n = 1.000.000: 0,376000 segundos

Tempo gasto para n = 5.000.000: 1,978000 segundos

Tempo de busca gasto para n = 1.000.000: 0,221000 segundos

Tempo de busca gasto para n = 5.000.000: 1,179000 segundos

Árvore Red-Black

Tempo gasto para n = 1.000.000: 1,070000 segundos

Tempo gasto para n = 5.000.000: 14,722000 segundos

Tempo de busca gasto para n = 1.000.000: 0,220000 segundos

Tempo de busca gasto para n = 5.000.000: 1,324000 segundos

Conclusões:

Tempo de inserção:

- ★ A árvore AVL teve um desempenho melhor no tempo de inserção em comparação com a árvore Red-Black. O que indica que a estrutura da árvore AVL é mais eficiente para inserções em grandes conjuntos de dados.

Tempo de busca:

- ★ As duas árvores apresentaram tempos de busca semelhantes, com a árvore AVL tendo uma leve vantagem no caso de 5.000.000 de operações. Isso sugere que a eficiência das duas estruturas é comparável para operações de busca.

Escalabilidade:

- ★ A árvore AVL mostrou-se mais escalável com o aumento do número de operações, mantendo um tempo de execução razoável. Enquanto isso, a árvore Red-Black mostrou um aumento significativo no tempo de execução à medida que o número de operações aumentava, indicando uma escalabilidade inferior.