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 <locale.h>
#include "ArvB.h"

void testeArvoreB(int m, int numOperacoes) {
    NoArvB *raiz = criarNo(m, true);
    clock_t inicio, fim;
    double tempoInsercao;

    inicio = clock();
    for (int i = 0; i < numOperacoes; i++) {
        int chave = rand();
        insert(&raiz, chave);
    }
    fim = clock();
    tempoInsercao = ((double) (fim - inicio)) / CLOCKS_PER_SEC;

    int alturaArvore = altura(raiz);

    printf("Árvore B (m = %d):\n", m);
    printf("Número de operações: %d\n", numOperacoes);
    printf("Tempo de inserção: %.2f segundos\n", tempoInsercao);
    printf("Altura da árvore: %d\n\n", alturaArvore);
}

int main() { setlocale(LC_ALL, "Portuguese");
    srand(time(NULL));

    printf("Teste com 1.000.000 de operações:\n");
    testeArvoreB(5, 1000000);
    testeArvoreB(10, 1000000);

    printf("Teste com 5.000.000 de operações:\n");
    testeArvoreB(5, 5000000);
    testeArvoreB(10, 5000000);

    return 0;
}
```

Resultados

Resultados da árvore B:

Para m = 5:
Tempo de inserção gasto para n = 1.000.000: 0,54 segundos
Tempo de inserção gasto para n = 5.000.000: 4,57 segundos
Tempo de busca gasto para n = 1.000.000: 0,287000 segundos
Tempo de busca gasto para n = 5.000.000: 1,474000 segundos

Para m = 10:
Tempo de inserção gasto para n = 1.000.000: 0,33 segundos

Tempo de inserção gasto para n = 5.000.000: 3,25 segundos
Tempo de busca gasto para n = 1.000.000: 0,261000 segundos
Tempo de busca gasto para n = 5.000.000: 1,377000 segundos

Resultados das árvores AVL e RB da atividade V:
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

Qual a altura das árvores AVL, RB e B geradas?
Árvore B

- Para m = 5:
    - Altura quando n = 1.000.000: 8
    - Altura quando n = 5.000.000: 9
- Para m = 10:
    - Altura quando n = 1.000.000: 6
    - Altura quando n = 5.000.000: 6

Árvore AVL

- Altura quando n = 1.000.000: 17
- Altura quando n = 5.000.000: 18

Árvore RB

- Altura quando n = 1.000.000: 26
- Altura quando n = 5.000.000: 30

Conclusão
A árvore B com m = 10 é mais eficiente para inserção do que com m = 5. E, quando se trata das buscas, a árvore B com m = 10 também é ligeiramente mais eficiente para buscas do que com m = 5.
Já a árvore AVL, tem um bom desempenho de inserção, sendo mais rápida que a árvore B para m = 5, mas um pouco mais lenta que a árvore B para m = 10. E é mais rápida para busca em comparação com a árvore B.

Enquanto a árvore RB é significativamente mais lenta para inserção em comparação com as árvores AVL e B. Porém, ela é comparável à árvore AVL em termos de tempo de busca, mas um pouco mais lenta para 5.000.000 de operações.
Em termos de altura, a árvore B com m = 10 tem uma altura significativamente menor, o que contribui para sua eficiência, a árvore AVL mantém uma altura relativamente baixa, contribuindo para um bom desempenho em inserção e busca e a árvore RB tem a maior altura entre as três, o que pode explicar o tempo de inserção mais lento.
No fim, escolher uma estrutura de dados adequada vai depender muito do contexto e de requisitos específicos. Se a inserção rápida for crítica, a árvore B com m = 10 é recomendada. Se a busca rápida for mais importante, a árvore AVL seria a melhor escolha. A árvore Red-Black, apesar de ser mais lenta para inserções, pode ser preferida em cenários onde a manutenção automática da altura balanceada é crucial e o tempo de inserção não é um fator limitante.