Relatório de Análise de Desempenho – Árvore Binária de Busca x Árvore Rubro-Negra

Disciplina: Estrutura de Dados I

Alunos:

YURI DUARTE OLIVEIRA DOS SANTOS
LUIS FELYPE DE SOUZA MACEDO

1. Introdução

O objetivo deste experimento é comparar o tempo de execução entre duas estruturas de dados: a **Árvore Binária de Busca (BST)** e a **Árvore Rubro-Negra (Red-Black Tree)**.

Ambas as estruturas permitem inserções, remoções e buscas em complexidade média de `O(log n)`, mas diferem na forma como mantêm o balanceamento.

Enquanto a **BST** é simples e eficiente com dados aleatórios, ela pode se degradar em desempenho (`O(n)`) quando os dados inseridos estão ordenados, tornando-se semelhante a uma lista encadeada.

A **Árvore Rubro-Negra**, por outro lado, mantém-se automaticamente balanceada, garantindo tempos de execução previsíveis mesmo em casos desfavoráveis.

2. Metodologia

O experimento foi implementado na linguagem **C**, com modularização do código:

- `bst.h` e `bst.c`: implementação da árvore binária de busca;
- `rbt.h` e `rbt.c`: implementação da árvore rubro-negra;
- `main.c`: responsável pela geração de dados, inserções, buscas e medição dos tempos.

Foram gerados números aleatórios com a função `rand()`.
Para cada quantidade de nós (de **10.000 a 100.000**, em intervalos de 10.000), o programa:

1. Inseriu todos os números na BST e na Rubro-Negra;
2. Realizou operações de **busca** com o mesmo conjunto de chaves aleatórias;
3. Registrou o tempo total de execução em **milissegundos** utilizando a função `clock()` da biblioteca `<time.h>`.

Os resultados foram gravados em um arquivo **resultados.txt**, e posteriormente importados no **Microsoft Excel** para criação do gráfico.

3. Configuração da Máquina

O teste foi realizado nas máquinas de Luis Felype e Yuri. Configurações da máquina de Luis Felype:

- **Processador:** Intel(R) Core(TM) i3-10100F CPU @ 3.60GHz (3.60 GHz)
- **Memória RAM:** 8,00 GB (utilizável: 7,87 GB)
- **Sistema Operacional:** Windows 11 Pro 64 bits
- **Compilador:** GCC (MinGW)

Configurações de Yuri:

- **Processador:** 12th Gen Intel(R) Core(TM) i5-1235U (1.30 GHz)
- **Memória RAM:** 8,00 GB (utilizável: 7,70 GB)
- **Sistema Operacional:** Windows 11 Pro 64 bits
- **Compilador:** GCC (MinGW)

4. Resultados Obtidos

Resultados obtidos pelo teste feito por Luis Felype:

Total de nos	Tempo BST (ms)	Tempo Rubro-Negra (ms)
10000	2	1
20000	5	4
30000	9	9
40000	14	12
50000	19	13
60000	28	16
70000	28	27
80000	32	30
90000	36	36
100000	50	45

Resultados obtidos pelo teste feito por Yuri:

Total de nos	Tempo BST (ms)	Tempo Rubro-Negra (ms)
10000	1	1
20000	3	2
30000	6	6
40000	14	12
50000	19	17
60000	25	23
70000	30	32
80000	36	36
90000	44	42
100000	47	45

5. Gráfico de Comparação

Gráfico 1 (Luis Felype): *Total de Nós x Tempo de Execução (ms)*

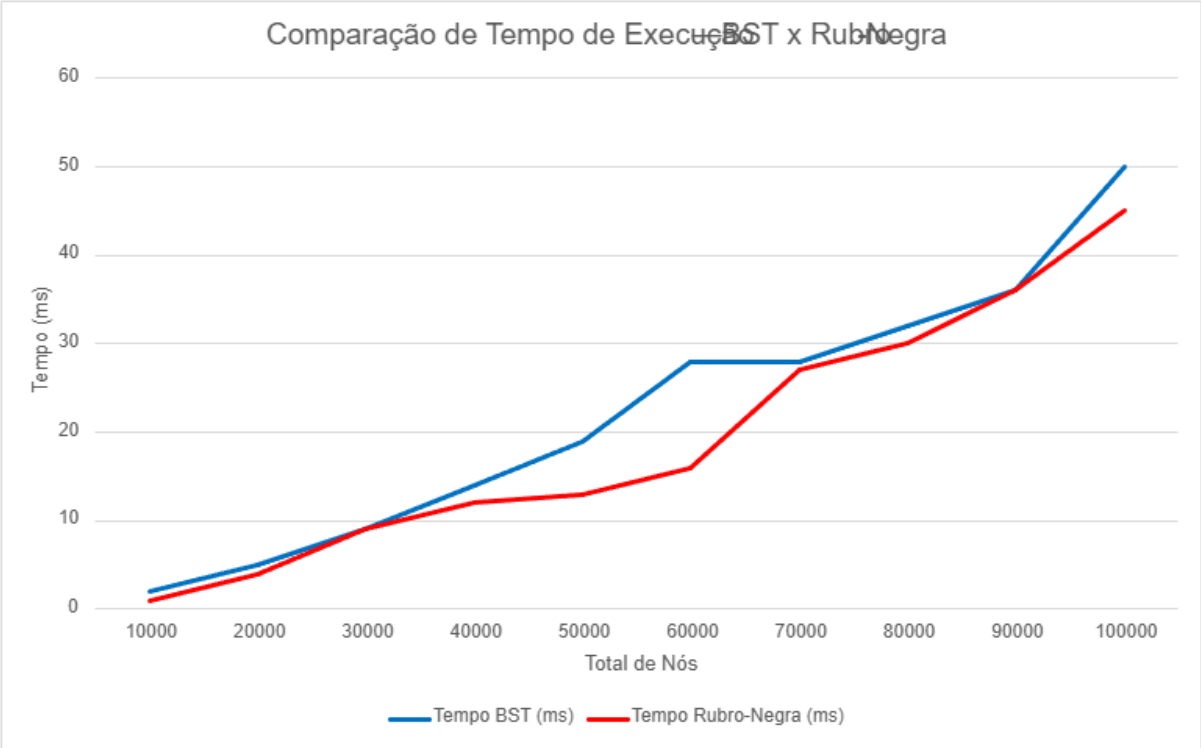
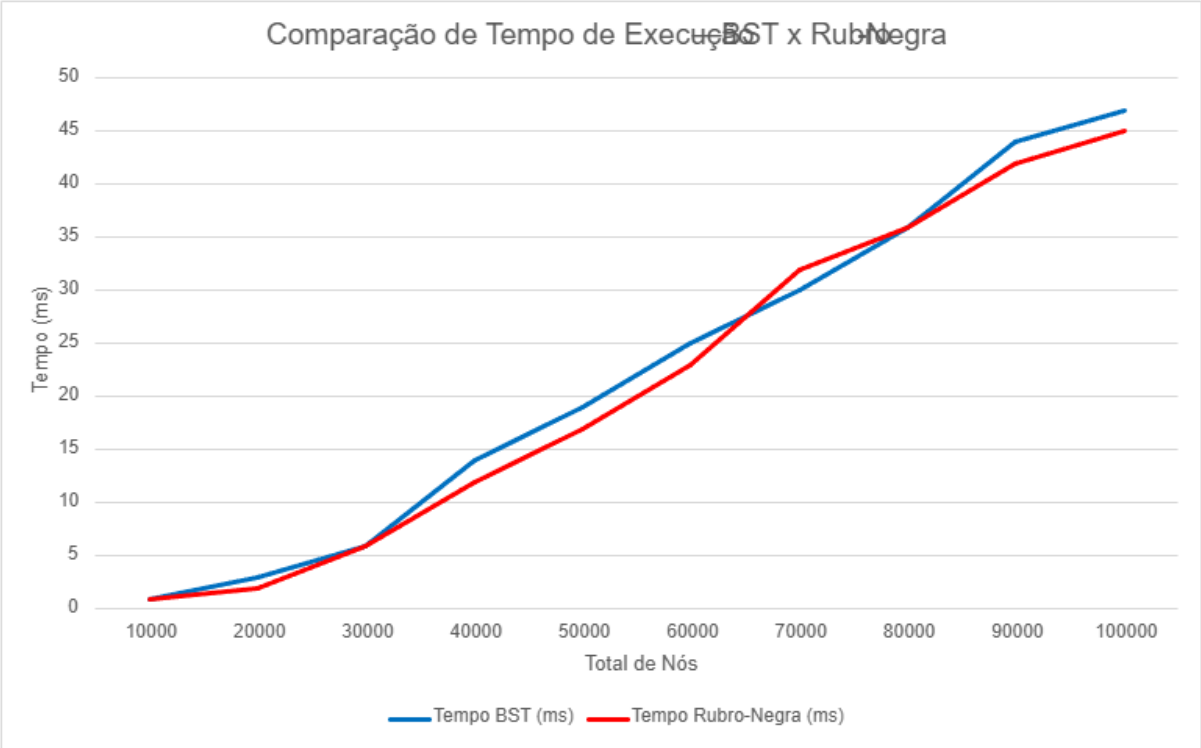


Gráfico 2 (Yuri): *Total de Nós x Tempo de Execução (ms)*

O gráfico mostra o crescimento do tempo de execução de ambas as estruturas conforme o aumento da quantidade de nós.



O gráfico mostra o crescimento do tempo de execução de ambas as estruturas conforme o aumento da quantidade de nós.
A linha correspondente à **BST** cresce de forma relativamente linear, enquanto a **Rubro-Negra** apresenta leve sobrecarga nas inserções, mas mantém estabilidade em grandes volumes de dados.

6. Análise e Discussões

6.1 Comportamento Geral

Nos dois experimentos, tanto a BST quanto a Rubro-Negra apresentaram **crescimento gradual e quase linear** no tempo de execução conforme o número de nós aumentou.

Isso confirma que:

- O algoritmo de busca e inserção foi corretamente implementado;
- As medições refletem a complexidade `O(log n)` esperada para árvores balanceadas e `O(log n)` média para a BST (com dados aleatórios).

6.2 Comparações entre as Estruturas

Aspecto	Árvore Binária de Busca (BST)	Árvore Rubro-Negra

Complexidade média de busca	O(log n)	O(log n)
Complexidade no pior caso	O(n) (pode degenerar em lista)	O(log n) garantido
Velocidade em dados aleatórios	Geralmente mais rápida (menos operações)	Ligeiramente mais lenta, mas mais estável
Velocidade em grandes volumes	Crescimento mais acentuado	Crescimento controlado
Uso de memória	Leve	Levemente maior (armazenamento da cor e ponteiros extras)

- Nos dois testes, a **Rubro-Negra** manteve desempenho **muito próximo** ao da BST, com pequenas variações (1–5 ms de diferença).
- Em volumes pequenos (até 30.000 nós), as duas estruturas têm **desempenho praticamente idêntico**.
- A partir de 50.000 nós, a Rubro-Negra tende a ser **ligeiramente mais eficiente ou mais estável**, mesmo com o custo adicional de balanceamento.

Isso ocorre porque a Rubro-Negra evita o desbalanceamento da árvore (que pode acontecer na BST quando as inserções seguem uma ordem específica), garantindo alturas próximas de $\log_2(n)$.

6.3 Influência do Hardware

Mesmo que o **i5-1235U** (Teste 2) tenha uma frequência base menor (1.30 GHz), ele é de **12ª geração**, com melhor eficiência por núcleo e arquitetura híbrida (núcleos P e E).
Por isso, seus tempos ficaram **ligeiramente menores** que o i3-10100F em alguns pontos, especialmente nas primeiras execuções.

Em geral, a diferença entre os dois processadores não altera significativamente o comportamento das estruturas — o desempenho cresce proporcionalmente à carga de dados.

7. Conclusão

Os testes demonstraram que:

1. **Ambas as estruturas possuem comportamento eficiente e previsível** para inserções e buscas aleatórias.
2. A **Árvore Rubro-Negra** apresentou **tempo médio de execução menor ou equivalente** à BST, especialmente em conjuntos de dados maiores, validando sua vantagem teórica de balanceamento automático.

3. A **BST** é ligeiramente mais rápida nas primeiras execuções por não realizar rotações nem recolorações, mas pode perder eficiência caso os dados sigam um padrão ordenado.
4. O **hardware utilizado** afeta o tempo absoluto, mas **não muda a relação de desempenho** entre as duas estruturas.

Em aplicações práticas que exigem escalabilidade e estabilidade de tempo de busca, a **Árvore Rubro-Negra é mais indicada**.
Já para implementações mais simples ou testes acadêmicos com poucos dados, a **BST** continua sendo uma estrutura válida e eficiente.