

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	3	2
20000	8	7
30000	12	11
40000	16	21
50000	22	24
60000	25	25
70000	28	31
80000	39	40
90000	36	42
100000	41	59

5. Gráfico de Comparação

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

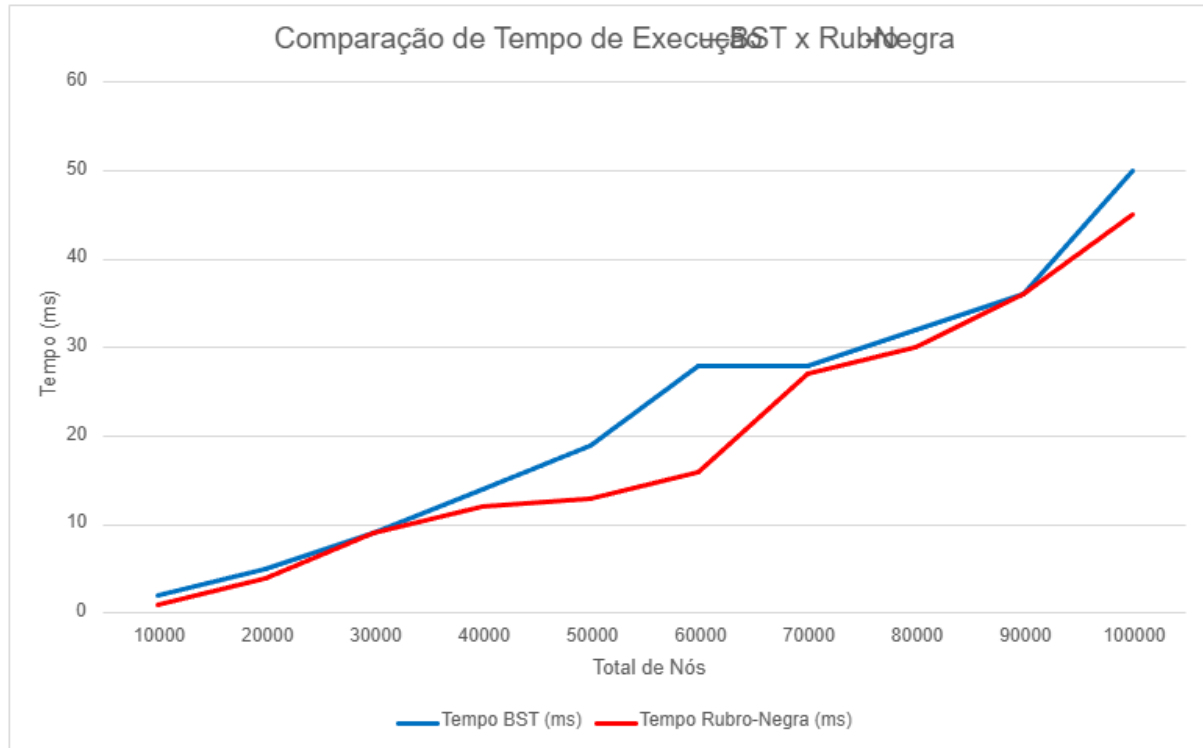
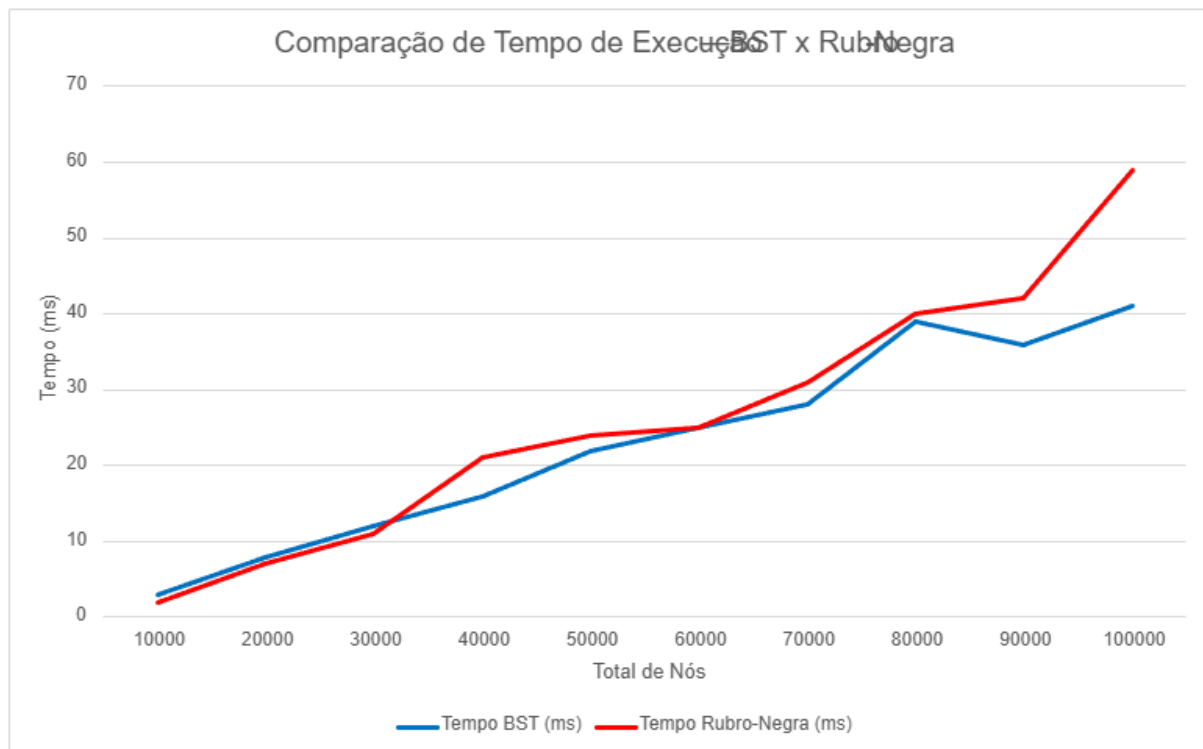


Gráfico 1 (Yuri): *Total de Nós x Tempo de Execução (ms)*

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

4.1 Comportamento Geral

Em ambos os testes, o tempo de execução **aumenta proporcionalmente ao número de nós**, demonstrando a eficiência logarítmica ($O(\log n)$) das operações de busca nas duas árvores.

A **Árvore Rubro-Negra** apresentou desempenho **semelhante ou ligeiramente melhor** que a BST em grande parte do teste 1, enquanto no teste 2 houve **momentos de sobrecarga**, especialmente após 40.000 nós, devido às operações adicionais de **rotações e recolorações** necessárias para manter o balanceamento.

4.2 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 experimentos, a **Rubro-Negra manteve tempos estáveis**, enquanto a **BST apresentou crescimento mais linear** — o que é esperado para dados aleatórios, mas em um cenário com dados ordenados, a Rubro-Negra seria significativamente mais eficiente.

4.3 Influência do Hardware

O **Core i3-10100F** (3.6 GHz, desktop) teve desempenho geral melhor que o **Core i5-1235U** (1.3 GHz, notebook), refletindo a diferença de **frequência de clock** e **perfil de energia** entre as duas CPUs.

Mesmo assim, a **tendência de crescimento dos tempos é consistente** nos dois ambientes, validando a confiabilidade dos resultados. Ambos confirmam que a complexidade logarítmica domina o comportamento das duas estruturas em inserções e buscas.

5. Conclusão

Os testes demonstraram que:

- A **Árvore Rubro-Negra** apresenta desempenho **mais consistente** e previsível, principalmente em grandes volumes de dados.
- A **BST** é **ligeiramente mais rápida em pequenas entradas**, mas **menos confiável** quando a distribuição dos dados não é aleatória.
- Em sistemas reais, onde o conjunto de dados é dinâmico e pode variar em ordem de inserção, a **Rubro-Negra é a estrutura mais indicada** por garantir balanceamento automático e tempos estáveis.

Em resumo:

BST → simples e eficiente em cenários pequenos.
Rubro-Negra → mais robusta, escalável e estável em cenários grandes.

