

Atividade Prática – AVL x Rubro Negra

Grupo: Tiago de Figueiredo Reis – 2023009225

Caio Mendes Ribeiro da Rosa – 2023002135

Os testes foram executados em um notebook Dell G3 15 com processador i7 9750H, 16GB RAM, sistema operacional Windows 11 e na IDE Clion, com compilador gcc 6.3.0.

A implementação das árvores está disponível no GitHub: [código](#)

1)

a) Realizando o benchmarking para inserção de 10000 valores ordenados crescentemente nas duas árvores, obteve-se os seguintes resultados.

AVL:

Entrada	Tempo (ms)	Quantidade de Rotações
10000	1	9986

Rubro:

Entrada	Tempo (ms)	Quantidade de Rotações
10000	2	9976

b) Realizando o benchmarking para remoção de 10000 valores ordenados crescentemente nas duas árvores, obteve-se os seguintes resultados.

AVL:

Entrada	Tempo (ms)	Quantidade de Rotações
10000	1	4988

Rubro:

Entrada	Tempo (ms)	Quantidade de Rotações
10000	1	4989

c) Realizando o benchmarking para a busca de 1000 valores ordenados crescentemente nas duas árvores, obteve-se os seguintes resultados.

AVL:

Entrada	Tempo (ms)
1000	1

Rubro:

Entrada	Tempo (ms)
---------	------------

1000	1
------	---

d) Na teoria, ambas as árvores apresentam complexidade assintótica igual a $O(\log n)$ para inserção, remoção e pesquisa, a AVL, por ser uma árvore com balanceamento mais rígido que a Rubro-Negra, deveria ser mais lenta na inserção e na remoção. Por outro lado, a AVL possui mais velocidade na busca de informações, pois como é uma árvore mais balanceada que a Rubro-Negra, sua altura é menor. Para uma melhor análise foram feitos alguns testes a mais em ambas as árvores.

AVL:

Entrada	inserção-Tempo(ms)	Inserção-Rotações
10000	2	9986
50000	8	49984
100000	14	99983
250000	34	249982
500000	70	499981
750000	112	749980
1000000	143	999980
5000000	763	4999977
10000000	1581	9999976
25000000	4077	24999975
50000000	8340	49999974
100000000	17363	99999973

Entrada	Remoção-Tempo(ms)	Remoção-Rotação
10000	2	4988
50000	4	24985
100000	9	49984
250000	21	124983
500000	50	249982
750000	70	374982
1000000	94	499981
5000000	510	2499979
10000000	1057	4999978
25000000	2793	12499977
50000000	5877	24999976
100000000	13241	49999975

Entrada-Busca	Busca-Tempo(ms)
5000	1
25000	5
50000	11
125000	23
250000	44
375000	68
500000	91

2500000	488
5000000	971
12500000	2419
25000000	5016
50000000	10310

Rubro-Negra:

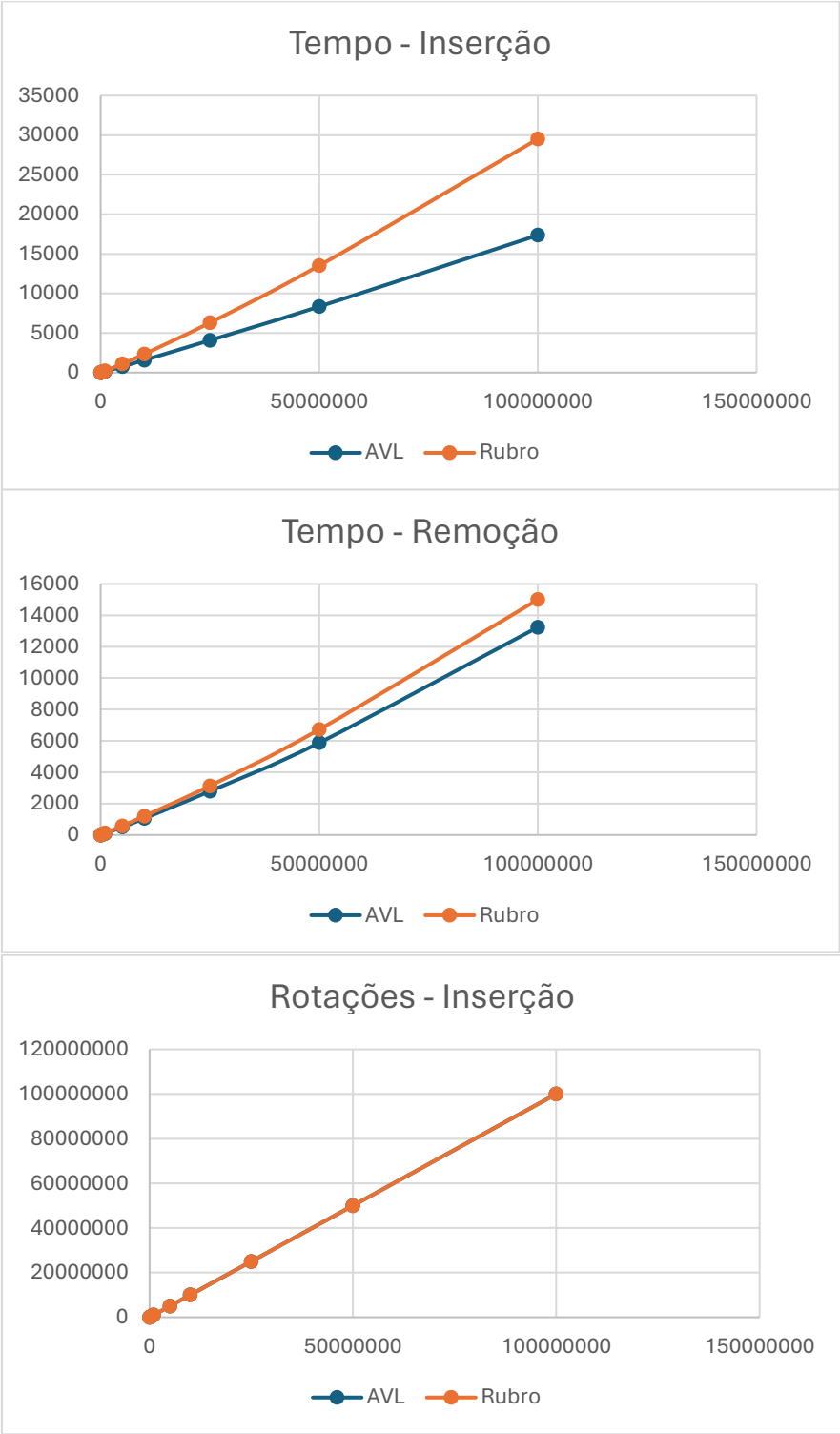
Entrada	Inserção-Tempo(ms)	Inserção-Rotações
10000	2	9976
50000	7	49971
100000	16	99969
250000	43	249967
500000	94	499965
750000	146	749964
1000000	200	999963
5000000	1119	4999958
10000000	2340	9999956
25000000	6304	24999954
50000000	13521	49999952
100000000	29517	99999950

Entrada	Remoção-Tempo(ms)	Remoção-Rotação
10000	1	4989
50000	5	24986
100000	10	49985
250000	26	124984
500000	51	249983
750000	78	374982
1000000	131	499982
5000000	581	2499980
10000000	1205	4999979
25000000	3133	12499977
50000000	6723	24999976
100000000	15003	49999975

Entrada-Busca	Busca-Tempo(ms)
5000	2
25000	5
50000	9
125000	22
250000	45
375000	67
500000	89
2500000	462
5000000	943
12500000	2426

25000000	5072
50000000	11199

Então foram gerados os seguintes gráficos.





Após realização dos testes, verificou-se que a complexidade assintótica das árvores estão corretas em ambas as métricas (tempo e quantidade de rotações), apresentando uma reta característica de $O(\log n)$, porém, em relação ao tempo a Rubro-Negra que deveria ser melhor na inserção e remoção em relação à AVL, apresentou um desempenho bem abaixo do esperado e, na busca, a teoria coincidiu com os testes, com a AVL apresentando uma melhor eficiência. Com relação à quantidade de rotações na inserção, a Rubro-Negra realmente apresentou uma quantidade um pouco menor em relação à AVL e, na remoção a Rubro-Negra apresenta uma quantidade maior em uma unidade ou igual em relação a AVL.