

# Binary Search Tree vs. AVL Tree

Resultados da minha implementação dos algoritmos da binary Search tree e AVL tree, nesse relatório irei referenciá-las como BST e AVL respectivamente.

A máquina em que os testes foram realizados tem como processador uma i5-6600 3.30GHz e 16GB DDR4, os resultados de tempo de execução foram obtidos utilizando os métodos `currentTimeMillis()`, o uso de CPU foi obtido através do JConsole.

Abaixo tem a tabela com os resultados, inserção refere-se a inserção de um único valor de escolha do usuário, inicialização representa a inserção em massa de valores aleatórios, todos valores aleatórios foram gerados no range de 0 a 1000.

|           | inicializar        | inserção      | remoção       | busca         |
|-----------|--------------------|---------------|---------------|---------------|
| BST 100   | 1ms / <1% CPU      | 0ms / <1% CPU | 0ms / <1% CPU | 0ms / <1% CPU |
| AVL 100   | 2ms / <1% CPU      | 0ms / <1% CPU | 0ms / <1% CPU | 0ms / <1% CPU |
| BST 500   | 1ms / <1% CPU      | 0ms / <1% CPU | 0ms / <1% CPU | 0ms / <1% CPU |
| AVL 500   | 9ms / <1% CPU      | 0ms / <1% CPU | 0ms / <1% CPU | 0ms / <1% CPU |
| BST 1000  | 1ms / <1% CPU      | 0ms / <1% CPU | 0ms / <1% CPU | 0ms / <1% CPU |
| AVL 1000  | 39ms / <1% CPU     | 0ms / <1% CPU | 0ms / <1% CPU | 0ms / <1% CPU |
| BST 10000 | 4ms / <1% CPU      | 0ms / <1% CPU | 0ms / <1% CPU | 0ms / <1% CPU |
| AVL 10000 | 6577ms / <25% CPU  | 2ms / <1% CPU | 2ms / <1% CPU | 0ms / <1% CPU |
| BST 20000 | 8ms / <1% CPU      | 0ms / <1% CPU | 0ms / <1% CPU | 0ms / <1% CPU |
| AVL 20000 | 26205ms / <26% CPU | 3ms / <1% CPU | 3ms / <1% CPU | 0ms / <1% CPU |

AVL melhor para busca já que fica com altura menor que a BST, que chegava nas dezenas nos últimos testes, a BST não tem necessidade de fazer rebalanceamentos portanto teve uma performance melhor, principalmente nas inserções em massa.

Enquanto a implementação, na AVL o rebalanceamento é feito após cada inserção, o que torna a inserção de grandes quantidades de dados ineficiente. Uma forma de melhorar seria fazer o rebalanceamento após todos os dados serem inseridos.