**Relatório de Análise e Comparação de Algoritmos**

**Alunos: Caíke Azevedo, Eduardo Neves e Vinícius Zaia**

**Classes:**

1. AVLNode:

- Esta classe é um nó que se destina a ser usado na Árvore AVL.

- Cada nó possui um dado, uma altura e referências aos nós filho esquerdo e direito.

2. AVLTree:

- Esta classe implementa uma árvore AVL, uma árvore binária de busca auto balanceada.

- Contém funções para inserção, remoção e busca.

- Contém funções para rotação à esquerda e à direita, que são essenciais para manter a árvore balanceada.

3. BinaryTree:

- Esta classe implementa uma árvore binária de busca simples.

- Contém funções para inserção, remoção e busca.

4. Node:

- É a classe do nó da BinaryTree.

- Cada nó tem um dado e referências para os nós filho esquerdo e direito.

5. TreeComparison:

- Esta é a classe principal que compara o desempenho entre a BinaryTree e a AVLTree.

- Contém opções para inserir, remover, imprimir e comparar as operações de inserção, busca e remoção entre as duas árvores.

**Resultados Obtidos:**

Ao rodar o programa, a classe `TreeComparison` (classe main) oferece várias opções. A opção mais interessante para fins de comparação é a primeira ("Comparar árvores"), que compara as operações de inserção, busca e remoção entre a BinaryTree e a AVLTree para diferentes tamanhos de entradas.

O desempenho de ambas as árvores é medido em nanosegundos é impresso no console. A comparação é feita usando tamanhos de entrada de 100, 500, 1000, 10000 e 20000 elementos (como pedido pelo professor).

Os resultados são apresentados no próprio console, segue exemplo:



---

**Análise Crítica:**

- As árvores AVL sempre garantem que a árvore permaneça balanceada, enquanto a árvore binária de busca simples (BinaryTree) não. Portanto, em cenários onde os dados são inseridos em uma ordem que levaria a árvore binária de busca a se tornar desequilibrada (por exemplo, dados já ordenados), a AVLTree terá um desempenho muito superior.

- Na BinaryTree, a inserção é feita de forma direta, enquanto na AVLTree, após a inserção, são realizadas verificações e possivelmente rotações para garantir o balanceamento. Assim, a inserção na AVLTree pode ser ligeiramente mais lenta, mas isso é compensado pelo fato de que as operações subsequentes (como busca e remoção) são mais rápidas em uma árvore balanceada.

- Semelhante à inserção, após a remoção em uma AVLTree, verificações e possíveis rotações são realizadas para manter o balanceamento. Na BinaryTree, a remoção é direta.