

AVALIAÇÃO RA2

Resolução de Problemas Estruturados

Árvores Binárias

Integrantes:

Félix Augustus Motelevicz

Gustavo Sampietro Rinaldi

Explicação do Código:

Toda classe `ArvoreBinaria` possui uma raiz que é uma instância da classe `Node`, que por sua vez possui um valor, um nó para esquerda e um nó para direita.

O construtor da classe da `ArvoreBinaria` cria um nó e o atribui como sendo sua raiz.

O método `inserir` recebe um valor e checa se esse valor é maior ou menor que o nó atual, se for menor ele percorre o Nó da esquerda, já se for maior ou igual ele percorre o Nó da direita. Quando ele achar um Nó nulo, um novo Nó será criado com o valor passado como parâmetro.

O método `buscaBinaria` recebe um valor e procura se ele está na árvore. Ele faz isso percorrendo os nós, verificando se o valor é menor, maior ou igual à informação do nó. Se o valor é menor, ele percorre para esquerda, maior, para direita e se for igual ele retorna o nó atual. Caso ele não encontre esse valor, ele retornará nulo.

O método `remover` recebe um valor e antes de qualquer coisa checa se o valor está na árvore. Se ele achar, ele irá percorrer a árvore de forma similar aos métodos anteriores. Quando ele achar o valor, antes de remove-lo ele irá procurar o nó substituto, que será o menor valor maior que ele, só então ele remove o valor.

Na `ArvoreBinariaAVL` sempre após uma inserção ou uma remoção, a função de balanceamento é chamada. Essa função confere o Fator de Balanceamento para ver se uma rotação é necessária, e caso seja, a faz.

O método imprime serve para imprimir as Árvores de forma mais organizada.

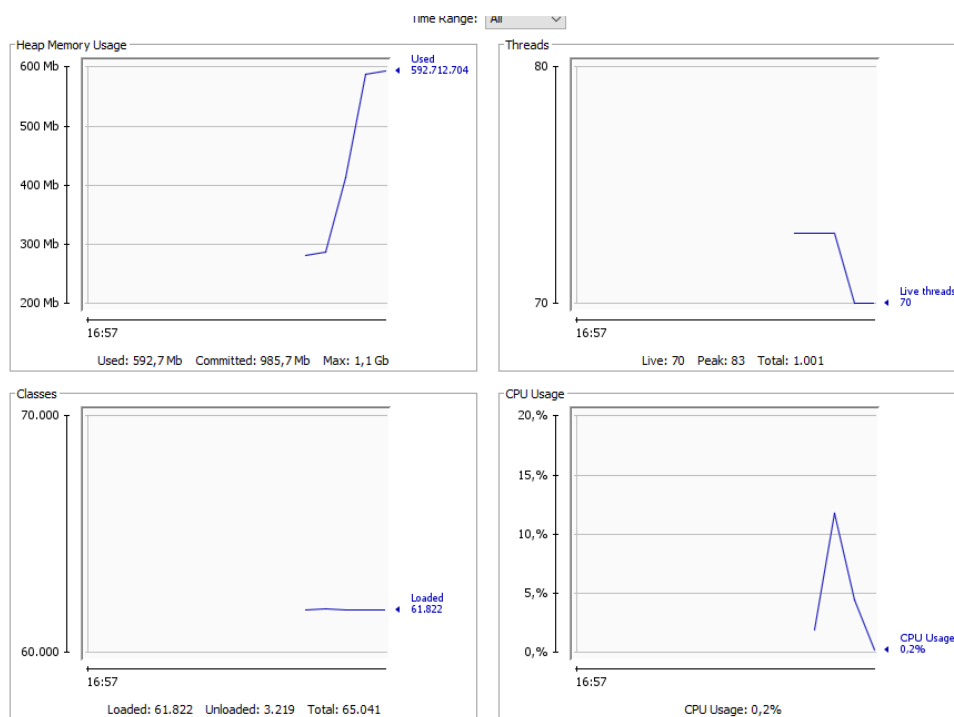
Análise de Desempenho :

Não foram utilizados métodos de Impressão.

Inserção, Busca e Remoção inclusos.

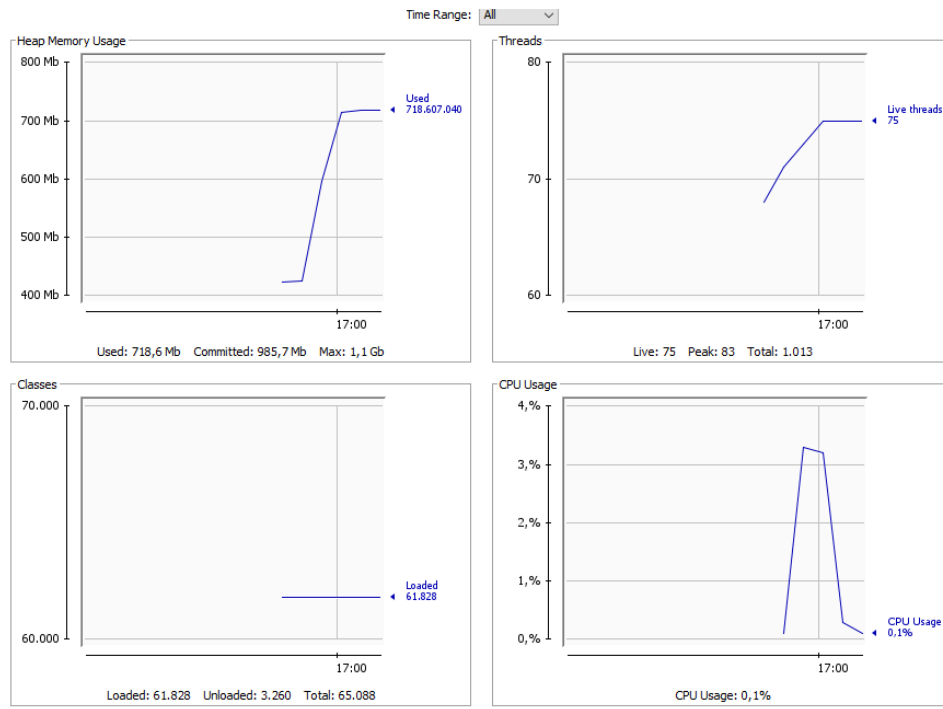
ArvoreBinaria com 100 números:

2ms



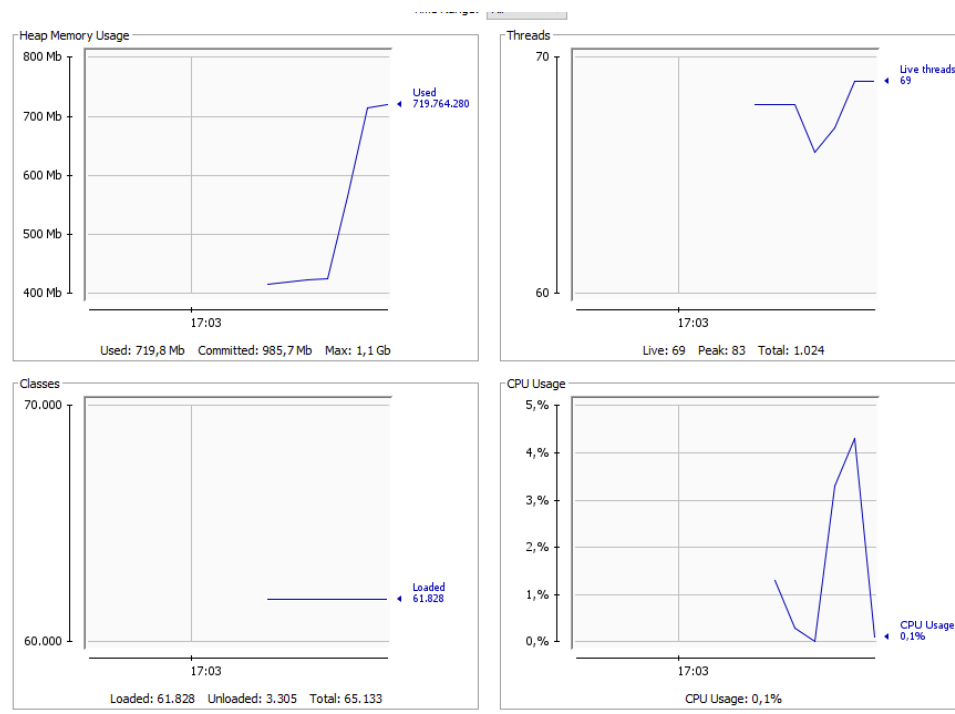
ArvoreBinariaAVL com 100 números:

3ms



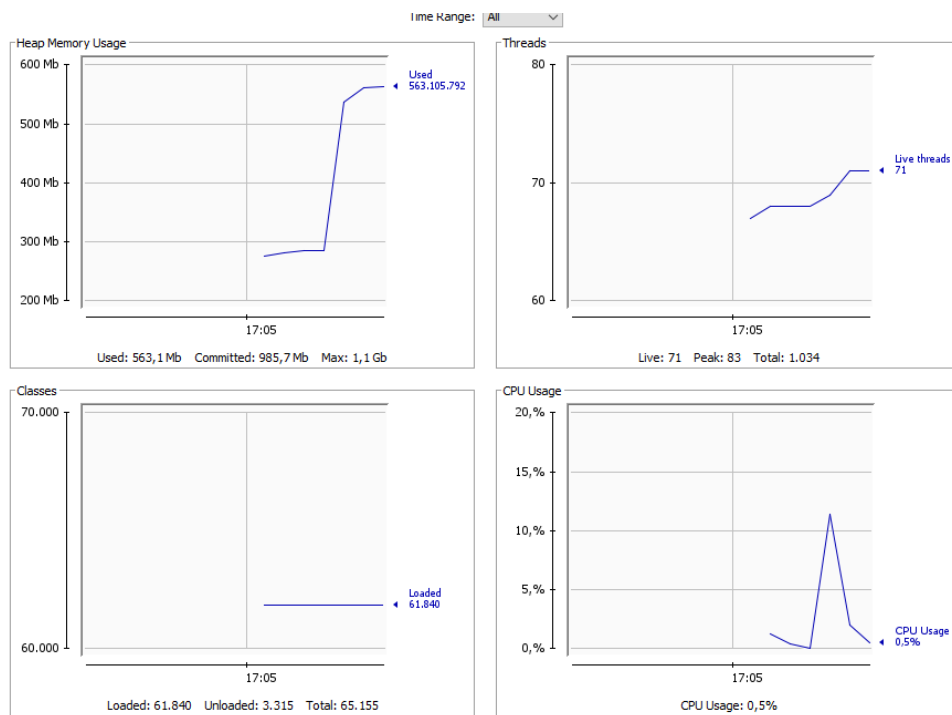
ArvoreBinaria com 500 números:

6ms



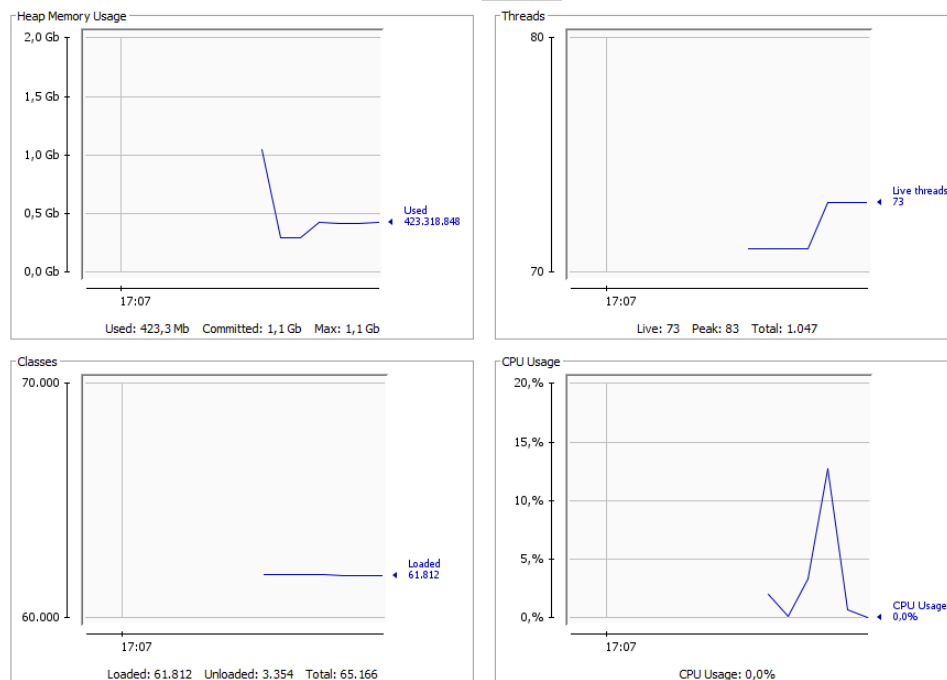
ArvoreBinariaAVL com 500 números:

6ms



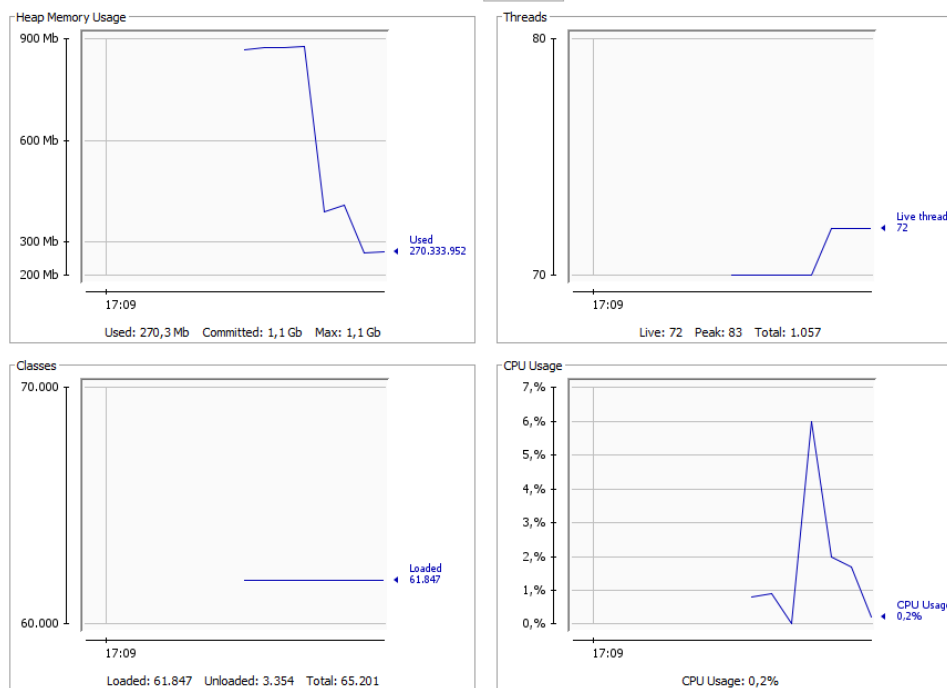
ArvoreBinaria com 1000 números:

4ms



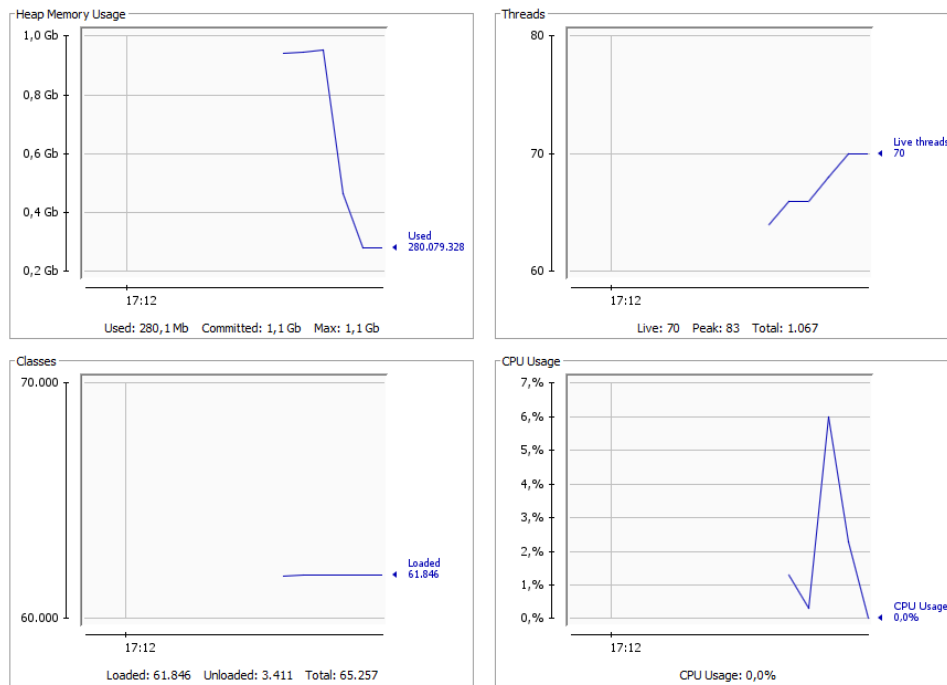
ArvoreBinariaAVL com 1000 números:

14ms



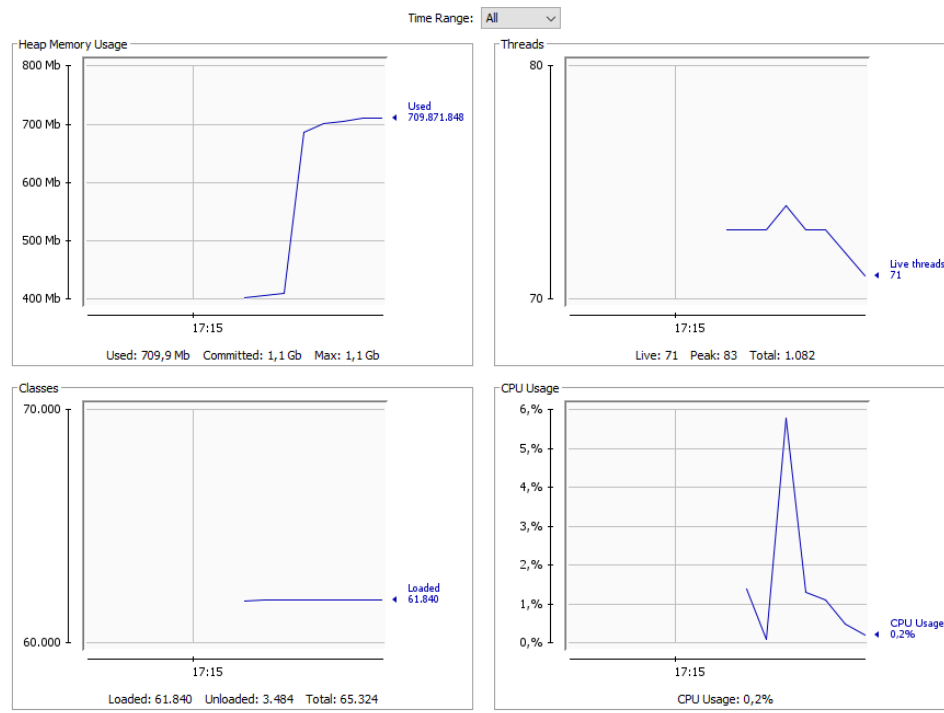
ArvoreBinaria com 10000 números:

10ms



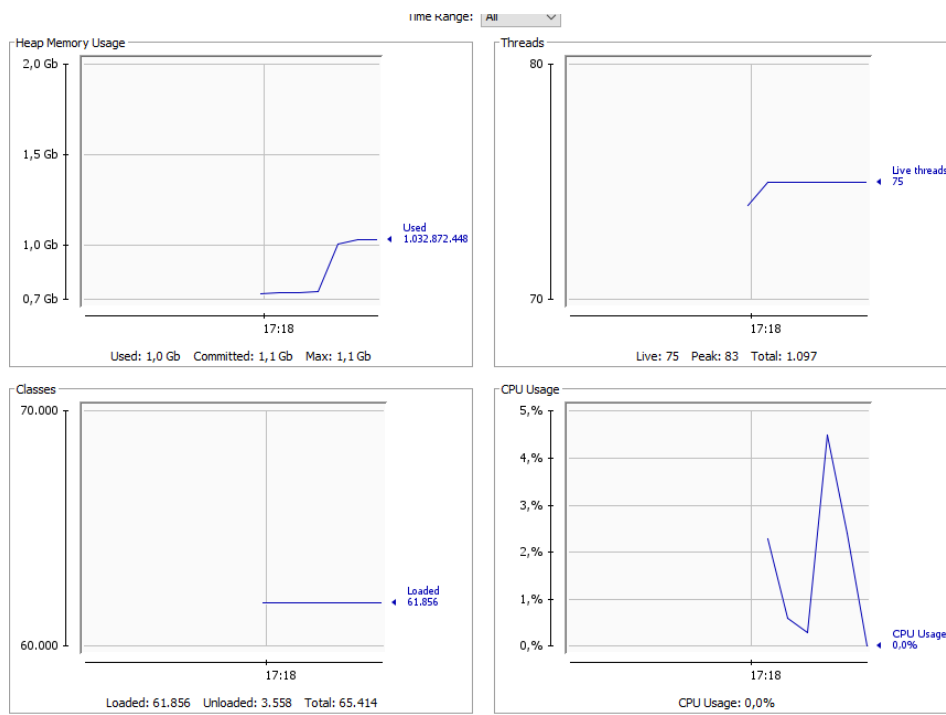
ArvoreBinariaAVL com 10000 números:

1568ms



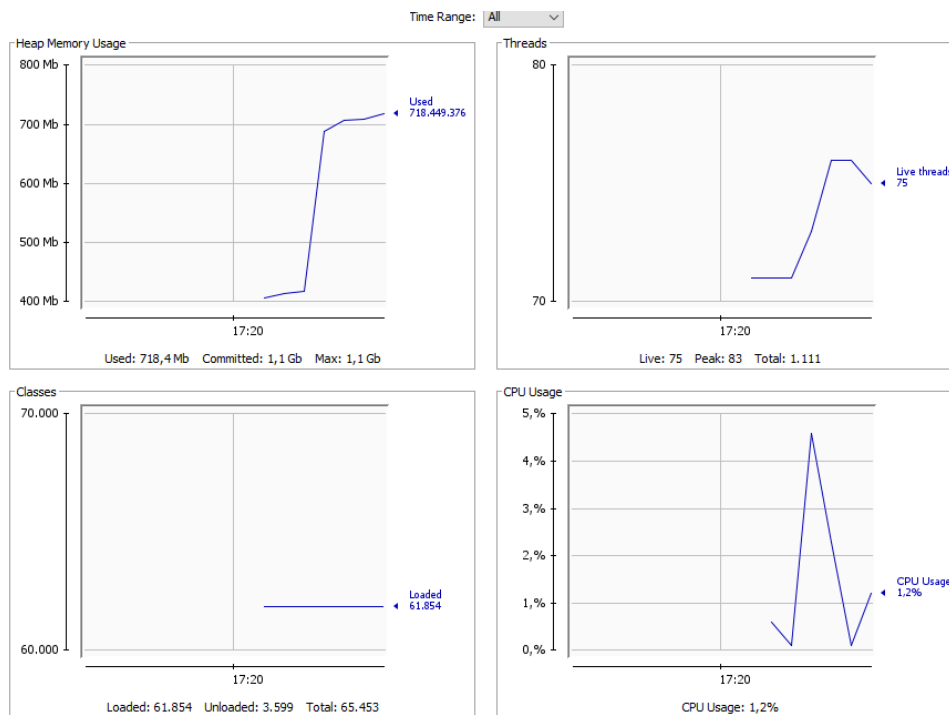
ArvoreBinaria com 20000 números:

43ms



ArvoreBinariaAVL com 20000 números:

6635ms



Conclusão:

A ArvoreBinaria é mais rápida na hora de inserção, pois não precisa ficar se reorganizando durante a ela, diferente da ArvoreBinariaAVL, que demora muito mais para terminar a inserção. Mas outras operações como busca e remoção, tendem a ser mais rápidas na ArvoreBinariaAVL, já que ela está organizada de forma balanceada, fazendo com que valores que normalmente estariam nos últimos nós, sejam mais fáceis de encontrar.