Avaliação RA02

Resolução de Problemas Estruturados em Computação Gustavo Albiero, Gustavo Fraiz

Diferenças de uso de uma árvore de busca binária e uma árvore AVL:

O método de busca é mais eficiente em uma árvore AVL porque a árvore vai estar sempre balanceada facilitando a percussão da árvore visto que a cada operação, metade das possibilidades serão diminuídas.

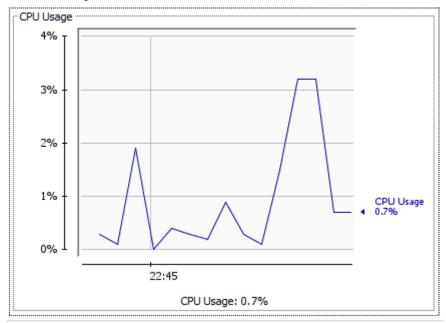
Porém a inserção e a remoção é prejudicada em relação a uma árvore de busca binária simples, a cada inserção ou remoção feita na árvore AVL, também é feito em conjunto outras operações de rotação dessa árvore, gerando uma perda de performance, enquanto na árvore BST é apenas a inserção ou remoção do elemento.

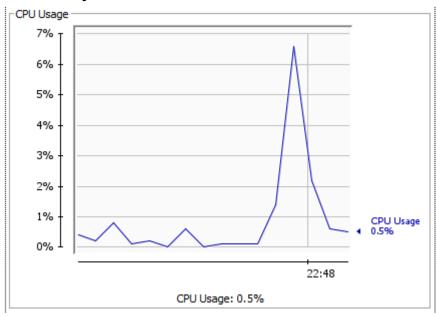
A árvore AVL, ao nosso olhar, é mais útil do que uma árvore de busca normal quando é preciso fazer muitas buscas dentro dos valores da árvore, em outros casos é melhor usar a árvore BST por ser mais fácil de implementar e não consumir muitos recursos na hora de inserir e remover.

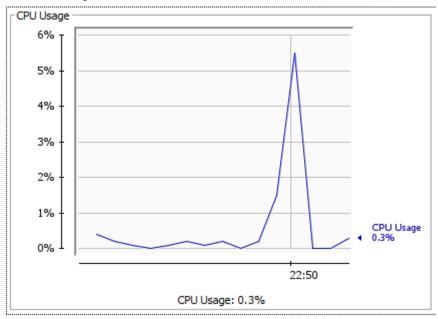
Testes:

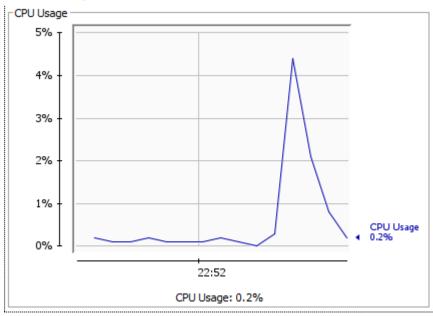
A ferramenta de monitoração usada foi o JConsole do kit de Desenvolvimento Java

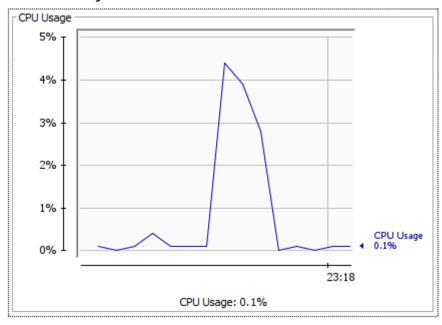
Inserções:

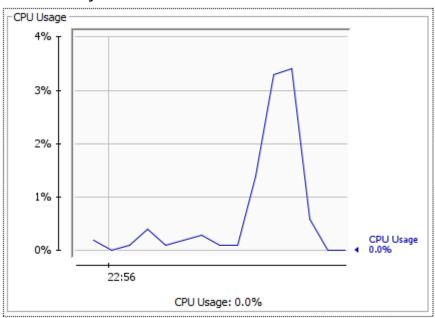


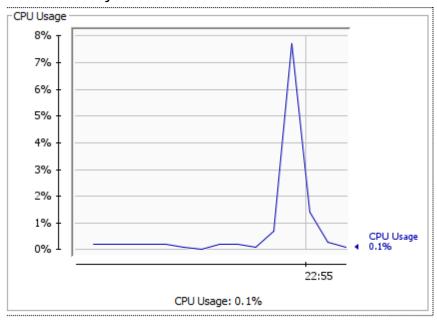


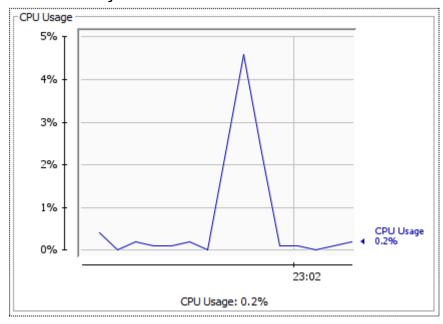


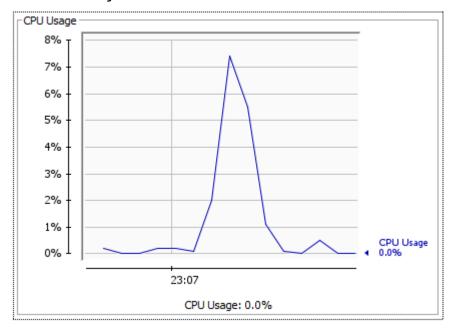


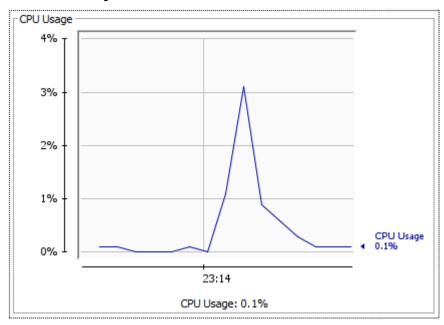






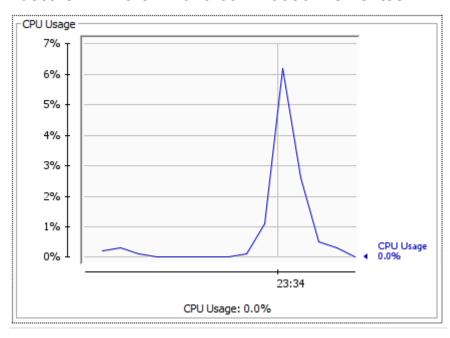




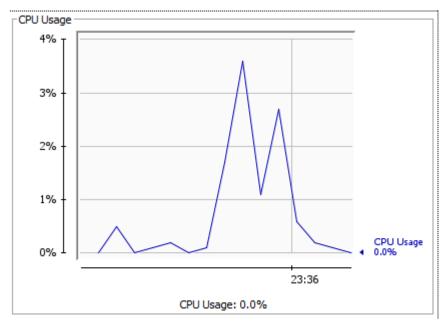


Busca:

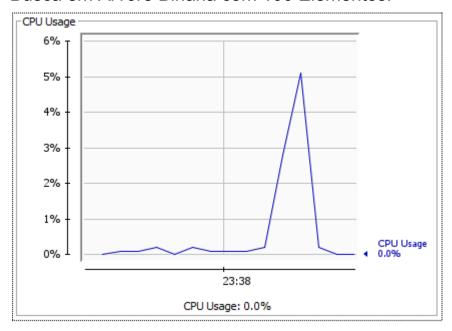
Busca em Árvore Binária com 20000 Elementos:



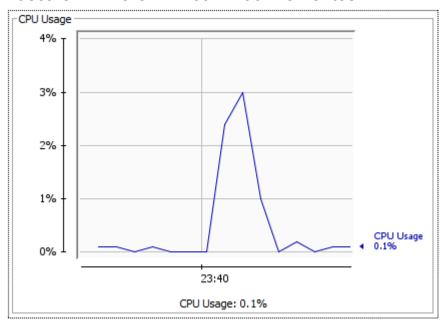
Busca em Árvore AVL com 20000 Elementos:



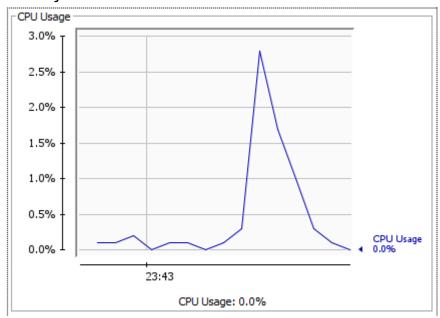
Busca em Árvore Binária com 100 Elementos:



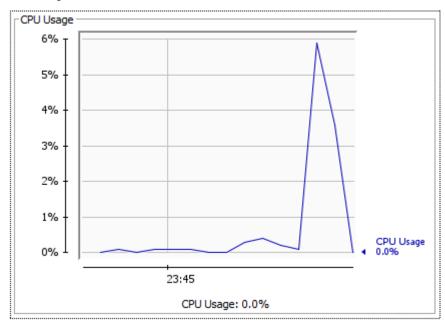
Busca em Árvore AVL com 100 Elementos:



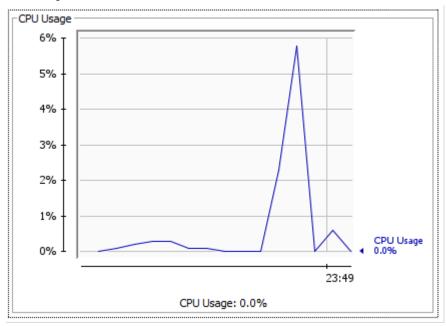
Remoção: Remoção de Árvore Binária com 100 elementos:



Remoção de Árvore AVL com 100 elementos:



Remoção de Árvore Binária com 20000 elementos:



Remoção de Árvore AVL com 20000 elementos:

