Nome: Rafael Thofehrn Castro
Gustavo Tochi Komura
Thiago Roscia Cerdeiro de Lima

Análise dos métodos

O trabalho foi implementado comparando dois métodos de balanceamento de árvores binárias de busca.

Para o primeiro método, a cada inserção ou remoção os balanceamentos já eram checados e devidamente arrumados com rotações para que se mantivessem entre -1 e 1. Neste caso, teremos sempre uma árvore com todos os nodos balanceados, característica de uma árvore AVL.

A cada inserção ou remoção, a árvore poderá continuar balanceada, necessitando de zero rotações, ficar desbalanceada do tipo esquerda-esquerda ou direita-direita, necessitando de uma rotação, ou ainda desbalanceada do tipo esquerda-direita ou direita-esquerda, necessitando de duas rotações. Em média, a cada operação (inserção ou remoção) temos 1 rotação, e para para n operações teremos n rotações (O(n)).

Para o segundo método, nenhuma rotação foi efetuada nas operações de inserção e remoção, criando uma árvore binária de busca qualquer. As rotações são efetuadas no final, ao executar o balanceamento pelo ponto médio, que consiste em particionar a raiz jogando para o topo o nodo intermediário, e em seguida, particionar os filhos da raiz recursivamente.

Neste caso, cálculo é feito em relação ao número de elementos da árvore não balanceada. O objetivo das rotações é levar um nodo folha até a raiz. Como a árvore é totalmente aleatória, não sabemos se para levar uma folha a raiz custará 1 rotação, ou n-1, caso a árvore foi criada a partir de um vetor ordenado. Imaginemos um caso médio, onde a árvore é completa, ou seja, o nível de todos os nodos folha é o mesmo. Neste caso, a altura da árvore é log2(n) e o número de rotações necessárias para levar uma folha até a raiz seria log2(n)-1. Como o algorítmo é efetuado para cada elemento da árvore, teremos um número de rotações aproximado de n(log2(n)): O(nlog2(n).

Pelas análises matemáticas, o algoritmo 1 é mais eficiente.

Realizamos testes executando um script que gerou centenas de árvores aleatórias, que foram balanceadas utilizando os dois métodos. Em média, o método 1 se sobressaiu em termos de desempenho, confirmando nossas conclusões previamente estabelecidas.

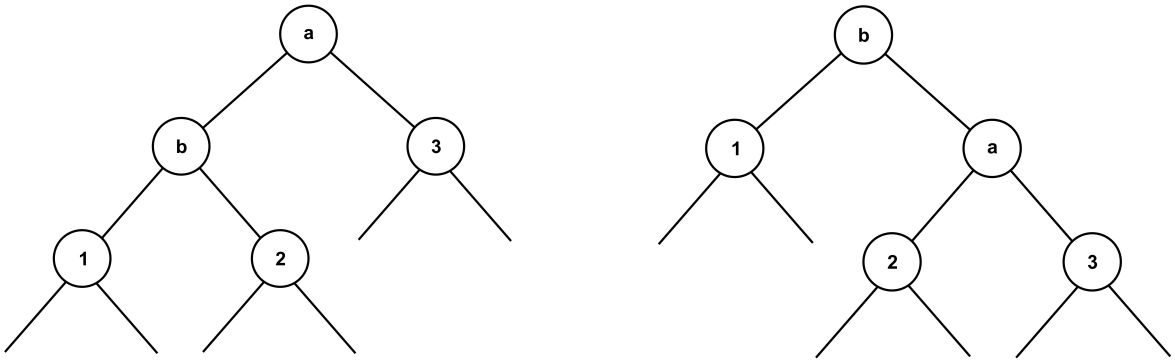
Cálculo dos balanceamentos

Artifício: Utilizamos módulo para pegarmos a maior altura entre 2 filhos de um nodo:

$$h_b = \frac{h_1 + h_2 + |b_b|}{2} + 1$$

onde b é o balanceamento do nodo b.

Rotação à direita, nodo 'a' com filho esquerdo 'b':



$$b_a = h_b - h_3 \quad \text{(I)}$$

$$b_b = h_1 - h_2 \quad \text{(II)}$$

$$h_b = \frac{h_1 + h_2 + |b_b|}{2} + 1 \quad \text{(III)}$$

Balanceamento final de a:

$$b_a' = h_2 - h_3$$

Substituindo (I), (II) e (III) na equação acima, obtemos:

$$b_a' = \frac{2b_a - b_b - |b_b| - 2}{2}$$

Balanceamento final de b:

$$h_a' = \frac{h_2 + h_3 + |b_a'|}{2} + 1 \quad \text{(IV)}$$

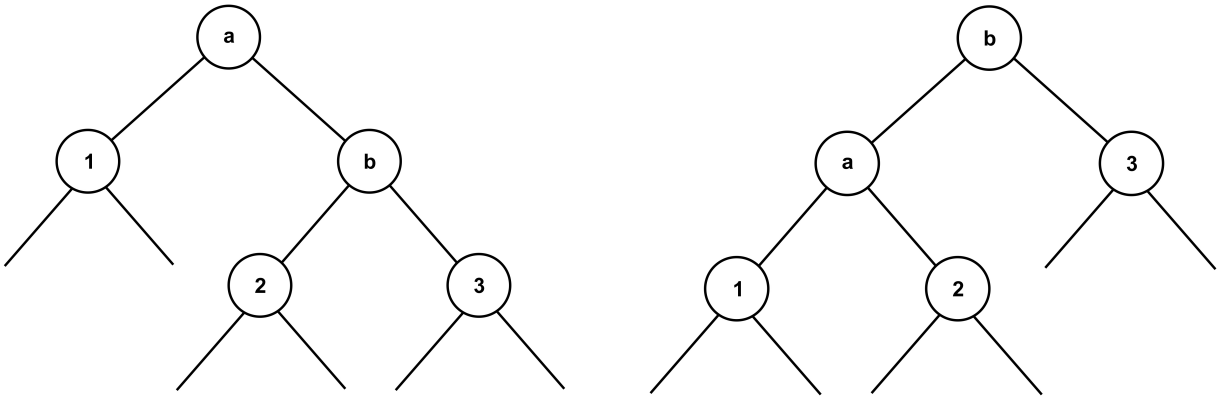
Substituindo (IV) em:

$$b_b' = h_1 - h_a'$$

Chegamos em:

$$b_b' = \frac{2b_b + b_a' - |b_a'| - 2}{2}$$

Rotação à esquerda, nodo 'a' com filho direito 'b':



$$b_a = h_1 - h_b \quad \text{(I)}$$

$$b_b = h_2 - h_3 \quad \text{(II)}$$

$$h_b = \frac{h_2 + h_3 + |b_b|}{2} + 1 \quad \text{(III)}$$

Balanceamento final de a:

$$b_a' = h_1 - h_2$$

Substituindo (I), (II) e (III) na equação acima, obtemos:

$$b_a' = \frac{2b_a - b_b + |b_b| + 2}{2}$$

Balanceamento final de b:

$$h_a' = \frac{h_1 + h_2 + |b_a'|}{2} + 1 \quad \text{(IV)}$$

Substituindo (IV) em:

$$b_b' = h_a' - h_3$$

Chegamos em:

$$b_b' = \frac{2\mathrm{b}_b + b_a' + |b_a'| + 2}{2}$$