

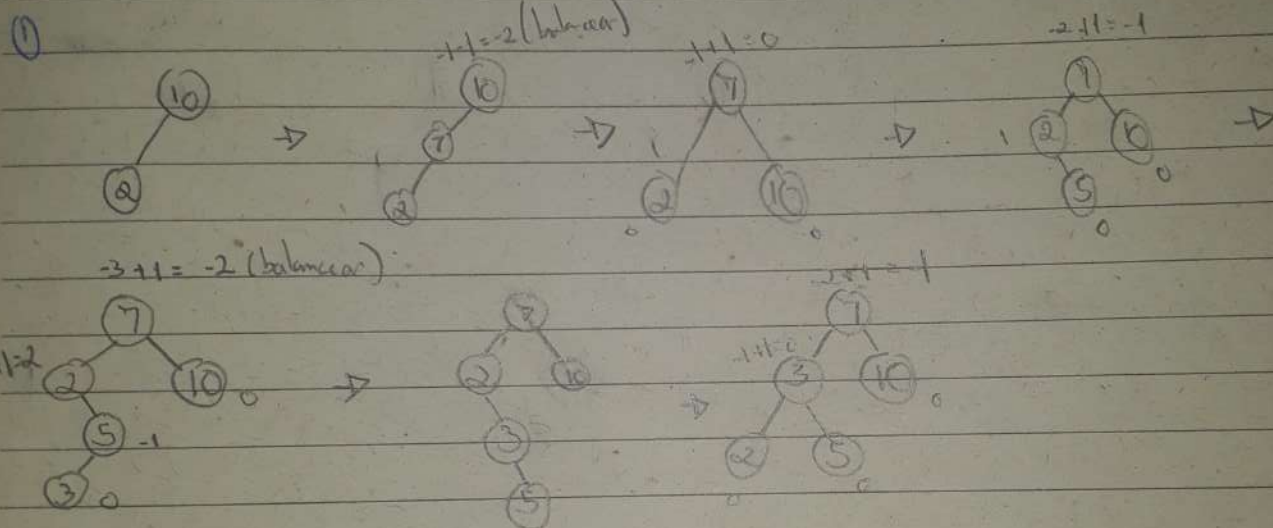
Exercício AEDS 2 - AVL - Frederico Dantés

A partir do conjunto de dados {10, 2, 7, 3, 5, 9, 16, 4, 11, 1, 6, 23, 14} faça:

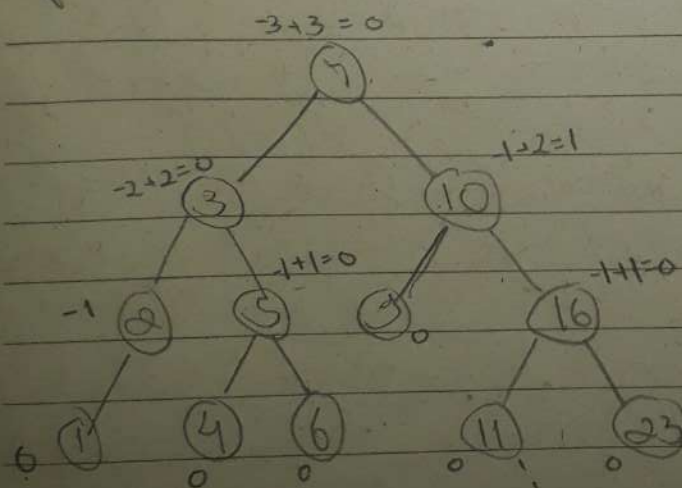
① construa a árvore AVL

② efetue a remoção de {10, 5, 4, 11, 6}

* Repetir passos com {78, 44, 67, 92, 22, 110, 16, 54, 23, 32, 37}



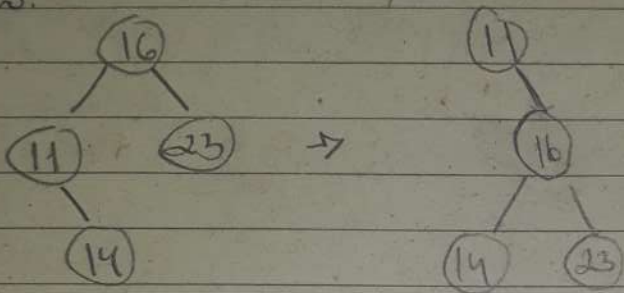
durante a inserção dos primeiros cinco valores, foi necessário balancear duas vezes a árvore AVL, durante a remoção dos dados {7, 3}, o que foi necessário. Os próximos sete dados inseridos, não houve necessidade de balanceamento, o que só foi necessário quando o último valor foi inserido {14}. Até a inserção do {14}, a árvore era:



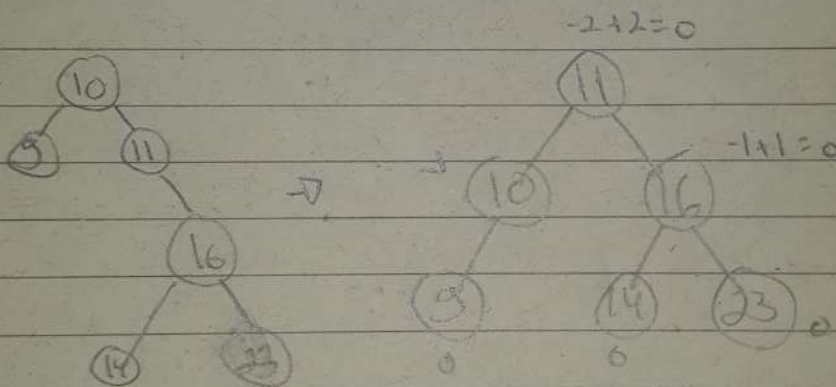
O valor de {14} só foi inserido no nó com o valor {10}, o que causou um desbalanço a favor em balanceamento.

Para o balanceamento da árvore, será necessário realizar uma rotação dupla no nó, no sentido anti-horário, primeiro com uma rotação no sentido horário no nó {16} e depois uma rotação simples no sentido anti-horário no nó {10}.

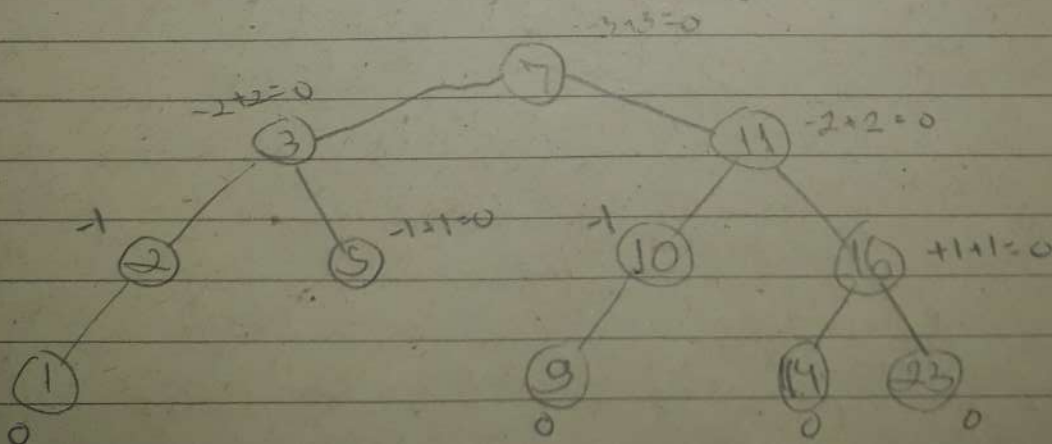
1ª rotação:



2ª rotação:



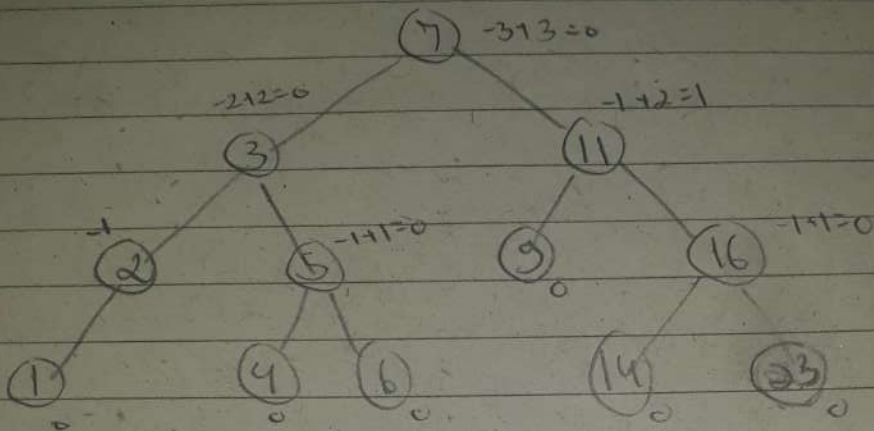
Após a terminação das inserções, a árvore fica assim:



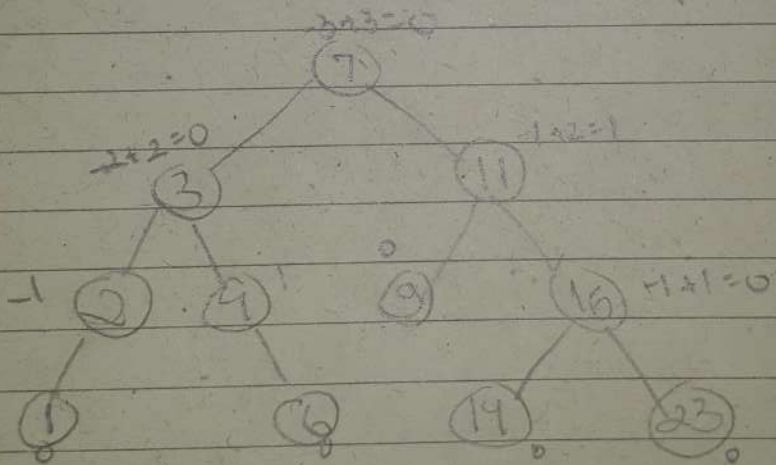
2

Após a inserção de todos os elementos e com a árvore balanceada, passamos a excluir. Iniciando com 9.

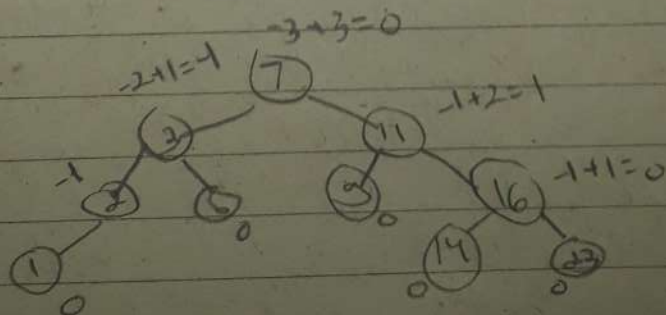
para o lugar do $\{10\}$, segundo a ordem que foi dada



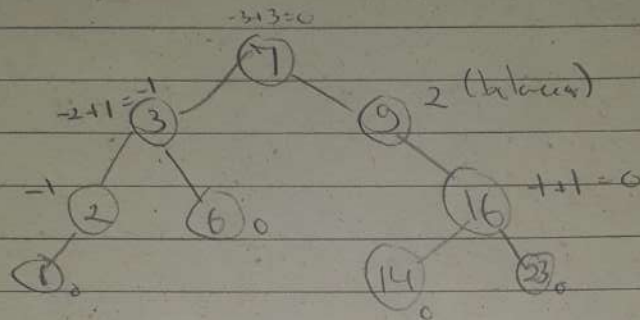
Passando para a exclusão do valor $\{5\}$, como ele tem filho direito demais, vamos fazer a rota esquerda até que a direita não tenha filho (nulo), esse valor então passará no lugar do $\{5\}$ que será excluído, neste caso será o $\{4\}$ a qual fará com que a árvore fique balanceada.



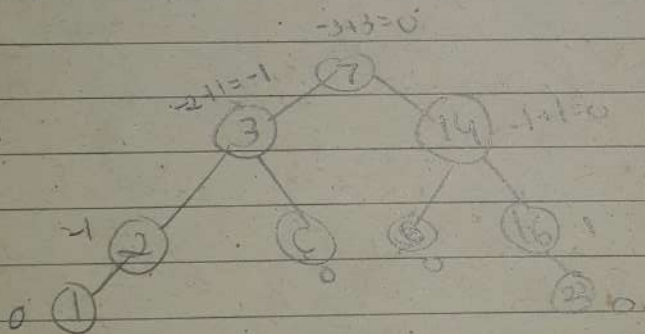
A próxima exclusão será o $\{4\}$, como ele tem filho direito, precisaremos o lado esquerdo, como nesse caso é nulo, basta substituir o $\{4\}$ pelo seu filho. Mantendo a AVL balanceada.



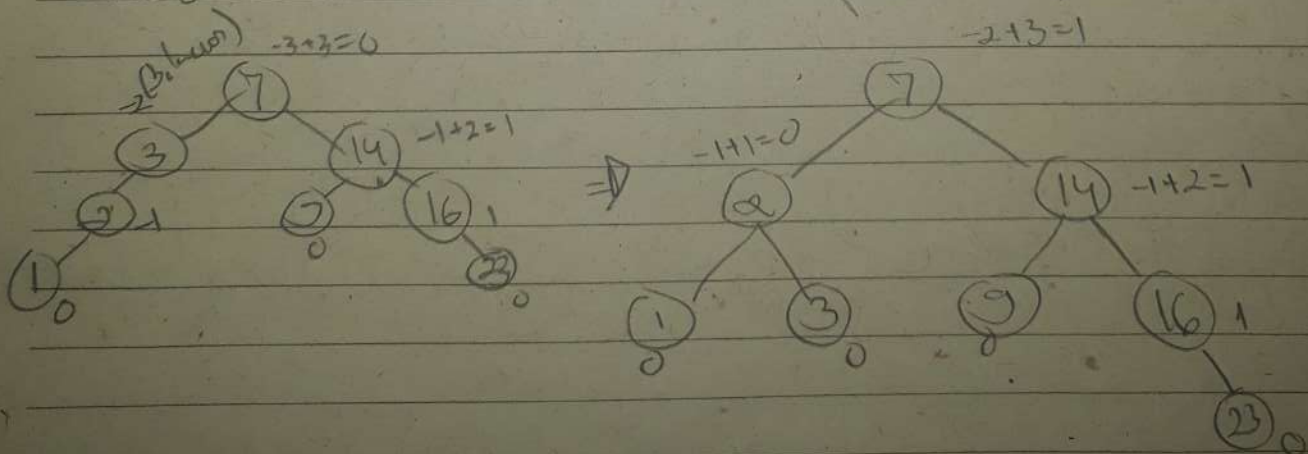
Quando para a próxima exclusão, tivermos o $\{11\}$, caso ele tem filho direito, passamos a esquerda e caso o $\{3\}$ não tem filho direito passamos ele pro lugar de $\{11\}$, para a árvore se ficar desbalanceada em 3 casos abaixo.



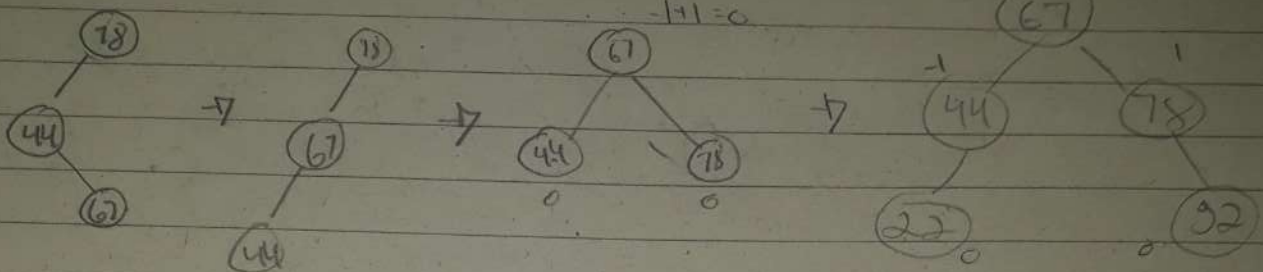
Para balancear passamos uma rotação a direita no nó $\{9\}$ e $\{11\}$, depois mudamos o nó entre $\{3\}$ e $\{14\}$, com uma rotação a esquerda sobre balanceado.



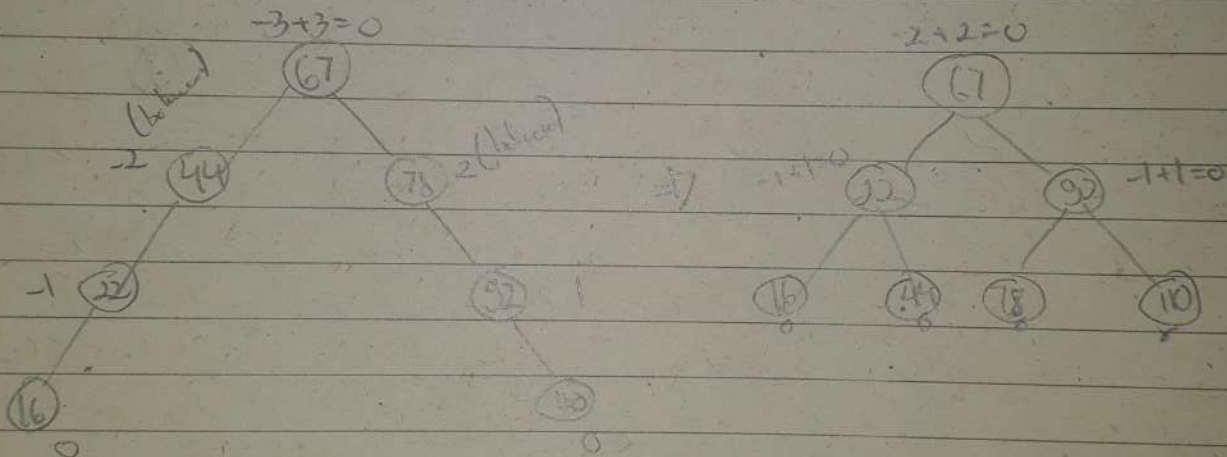
Para finalizar só mudamos o valor $\{6\}$ e com isso o nó $\{3\}$ ficará desbalanceado, sendo necessária uma rotação no nó $\{3\}$ para balancear.



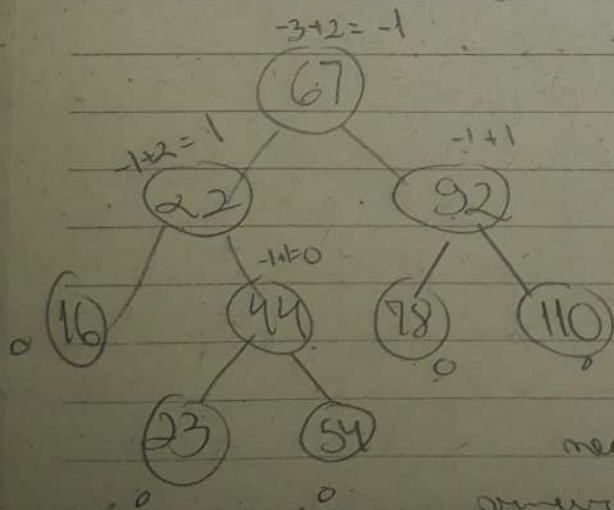
Fazer o mesmo com outros valores:



Aí o momento a inserção dos novos primeiros valores a árvore permanece balanceada. Ao inserir os dois próximos valores {16} e {110} a mesma via para um novo nível à direita - esquerda a que vai desbalancear a árvore nos nós {44} e {78}.

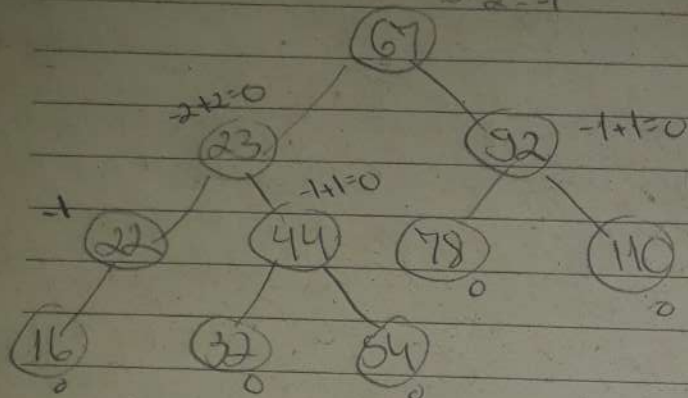


As inserções dos próximos valores {54} e {23} a árvore continuará balanceada.



Porém ao adicionar mais o próximo valor, {32} isso vai gerar um desbalanceamento no nó {22}, e por se tratar de um desbalanceamento à esquerda, será necessário uma rotação dupla para esquerda primeiro à direita e depois à esquerda.

$$2+2=-1$$

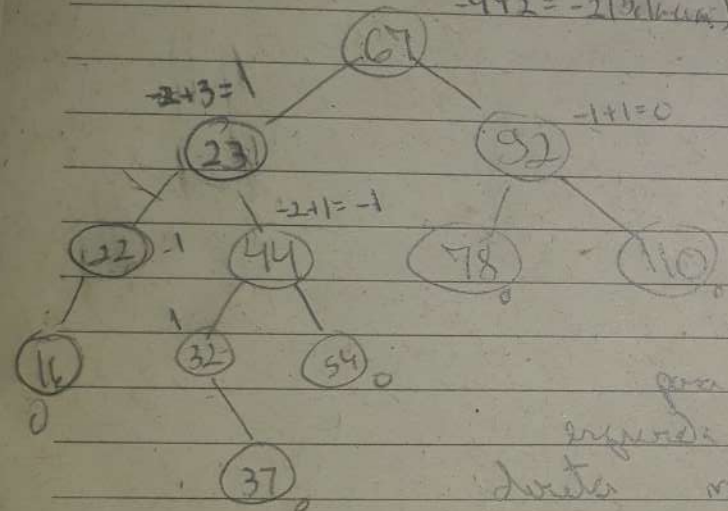


As operações para a primeira e última inserção, a rota SLL.

Porém isto virá com um desbalanceamento na raiz.

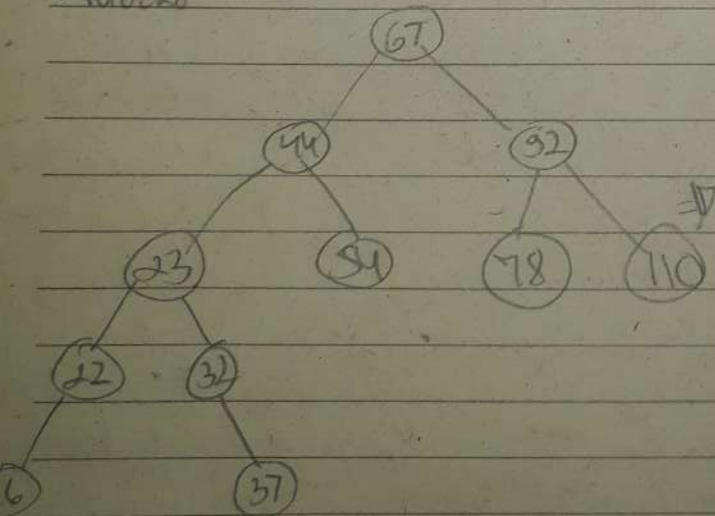


$$-4+2=-2 \text{ (L/R ou RL)}$$

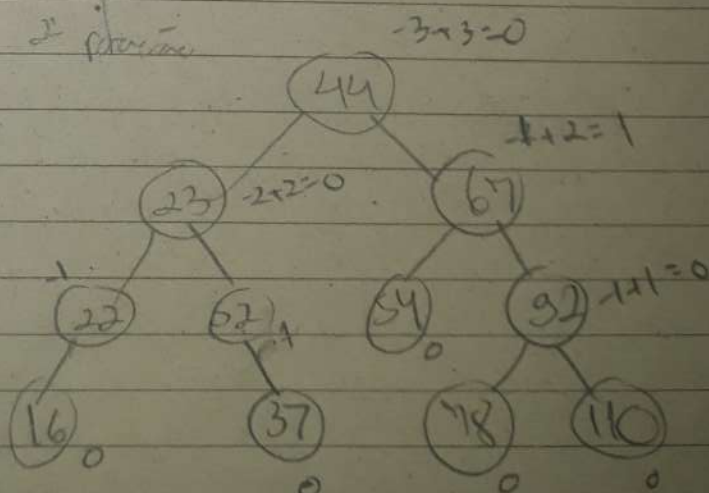


Como o desbalanceamento está na raiz, se tentarmos fazer uma rotação simples a direita ela continuará desbalanceada, ou seja, é necessário uma rotação dupla para a direita balancear, simples a esquerda no nó 23 e uma simples a direita no nó 67.

1ª rotação



2ª rotação



Reflexão sobre os benefícios do uso da AVL.

A árvore AVL tem como base a árvore binária, porém contendo com o incremento do balanceamento que é realizado quando necessário de forma a manter a estrutura equilibrada. Dando o uso a AVL possui um custo computacional maior por utilizar mais operações para alcançar o balanceamento. Entretanto essa particularidade tem vantagens também que seria uma maior facilidade de localizar o dado na estrutura e mantê-lo (insere ou remove) com uso de custo máximo que se teria com essa estrutura seria de $n \log n + 1$.