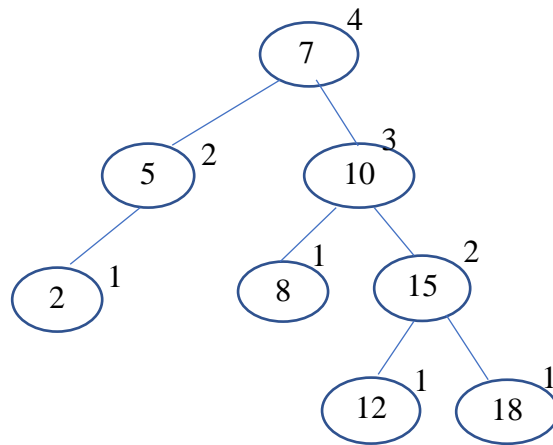


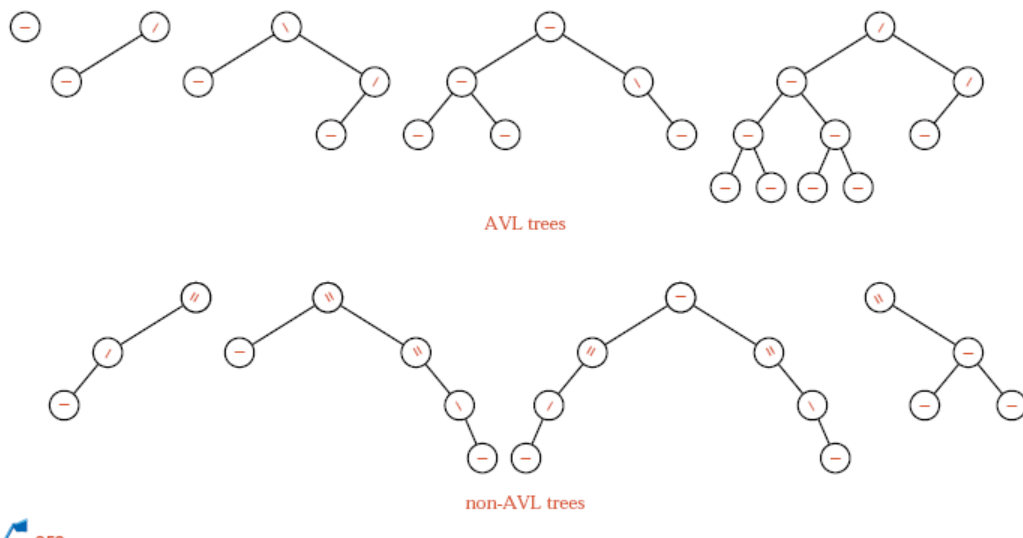
Tipo Abstrato de Dados (TAD) – Árvore AVL (Adelson-Velsky e Landi)



Uma árvore do tipo AVL é uma árvore de busca binária com a propriedade de que cada nó interno v da árvore T a altura dos filhos de v podem diferir em, no máximo, 1. A figura acima apresenta uma árvore do tipo AVL.

Uma árvore AVL difere de uma árvore de busca binária na estrutura do nó, pois ela mantém a informação da altura de cada nó e também nas operações de inserção e remoção onde ela atualiza e verifica a propriedade de balanceamento da árvore, e se esta propriedade for alterada as operações de rotação de sub-árvores são executadas para se manter a propriedade da árvore AVL.

A figura a seguir apresenta exemplos de árvores AVL e árvores não AVL.



A figura a seguir apresenta os passos para inserção de elementos na árvore AVL. Note que em nenhum momento a árvore deixou de ser AVL, o que pode não acontecer sempre.

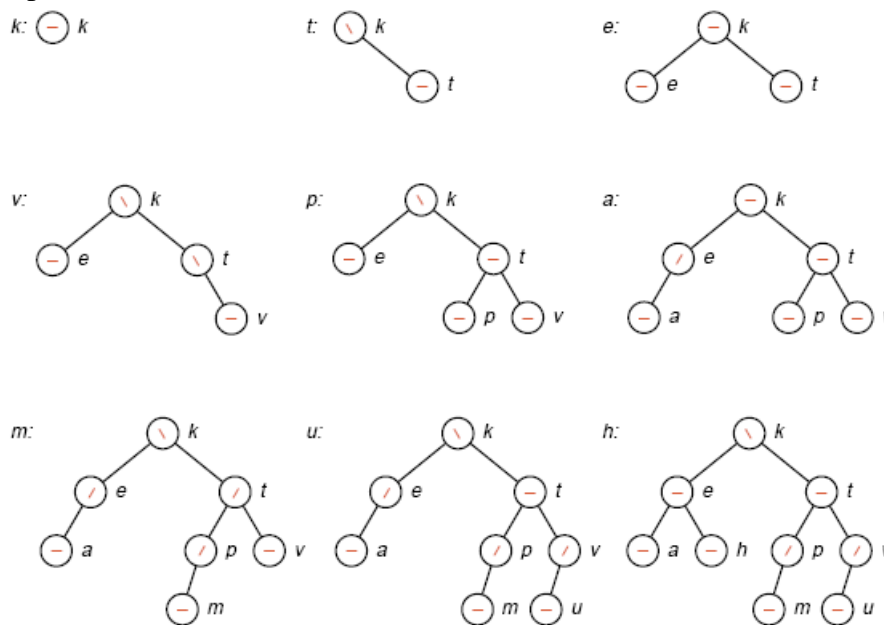


Figure 10.18. Simple insertions of nodes into an AVL tree

No caso da árvore ficar desbalanceada (a diferença da altura dos filhos é maior que 1), é necessário fazer rotações.

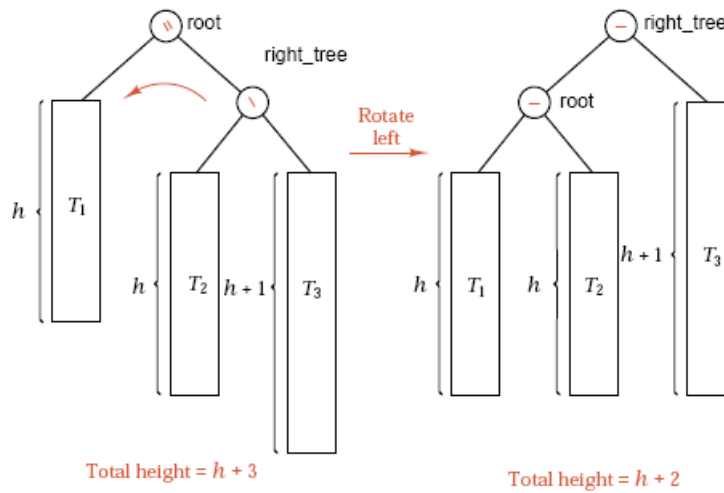
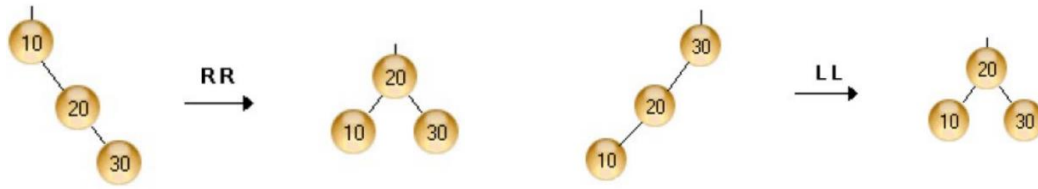
As operações de rotação podem ser simples ou duplas.

Na inserção utiliza-se um processo de balanceamento que pode ser de 4 tipos específicos:

- RR → caso Right-Right (rotação a esquerda)
- LL → caso Left-Left (rotação a direita)
- LR → caso Left-Right (rotação esquerda-direita)
- RL → caso Right-Left (rotação direita-esquerda)

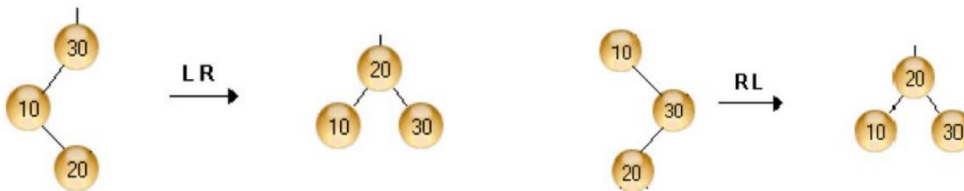
1. Rotação simples

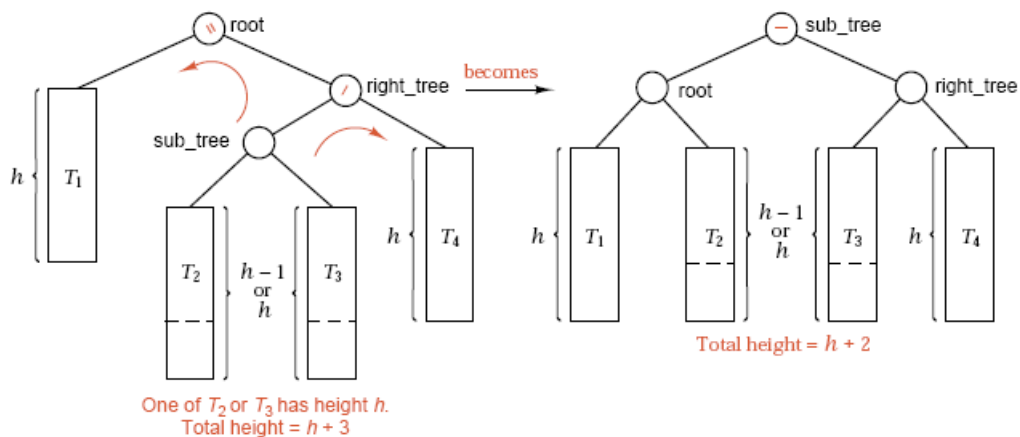
As rotações simples podem ser RR ou LL, tal como mostrado nas figuras abaixo.



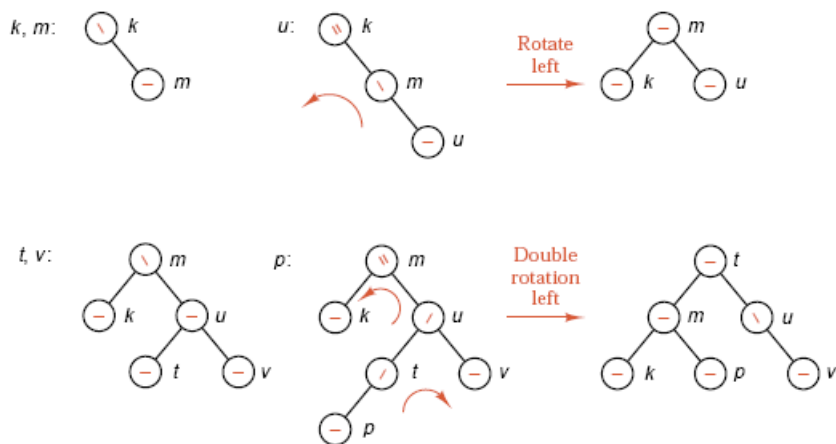
2. Rotação Dupla

As rotações duplas podem ser LR ou RL, tal como apresentado nas figuras abaixo.





Exemplos:



Exercícios:

1. Em cada um dos itens a seguir, insira as chaves, na ordem mostrada, com o resultado de cada inserção sendo uma árvore AVL.

- (a) A, Z, B, Y, C, X.
- (b) A, B, C, D, E, F.
- (c) M, T, E, A, Z, G, P.
- (d) A, Z, B, Y, C, X, D, W, E, V, F.
- (e) A, B, C, D, E, F, G, H, I, J, K, L.
- (f) A, V, L, T, R, E, I, S, O, K.

2. considere uma árvore de busca binária e uma árvore AVL de altura 4. Mostre como seria a estrutura destas duas árvores com o número mínimo de nós e com o número máximo de nós.

3. dada uma estrutura de árvore AVL descreva um algoritmo para encontrar o k-ésimo menor elemento na árvore em tempo $O(N)$, onde N é o número de elementos da árvore. Na árvore desenhada abaixo, o quarto menor elemento da árvore é o 8 e o sétimo menor elemento é o 15.

