

Re-balanceamento de Árvore - AVL

Árvore AVL:

Em 1962 os matemáticos Russos G. M. Adelson-Velskii e E. M. Landis sugeriram uma definição para “near balance” e descreveram procedimentos para inserção e eliminação de nós nessas árvores: os algoritmos de balanceamento de árvores são chamados de **algoritmos AVL** e as árvores são chamadas de **árvores AVL**.

Uma árvore AVL é uma árvore binária de busca (ABB) construída de tal modo que a altura de sua subárvore direita difere da altura da subárvore esquerda de no máximo 1.

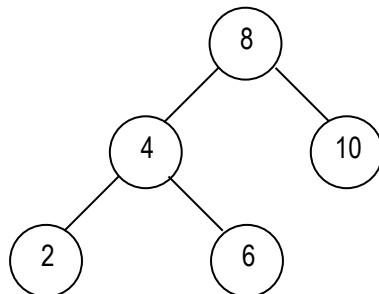
Inserção AVL:

O que pode acontecer quando um novo nó é inserido numa árvore balanceada?

Dada uma raiz **r** com subárvores L (left) e R (right), e supondo que a inserção deve ser feita na subárvore da esquerda. Podemos distinguir 3 casos:

1. Se $hL = hR$, então L e R ficam com alturas diferentes mas continuam balanceadas.
2. Se $hL < hR$, então L e R ficam com alturas iguais e o balanceamento foi melhorado.
3. Se $hL > hR$, então L fica ainda maior e o balanceamento foi violado.

Na árvore abaixo:



- Nós 9 ou 11 podem ser inseridos sem re-balanceamento. O nó 10 passa a ter uma subárvore e a subárvore com raiz 8 vai ficar melhor balanceada!
- Inserção dos nós 3, 5 ou 7 requerem que a árvore seja re-balanceada!

Fator de balanceamento (FB) de um nó na AVL

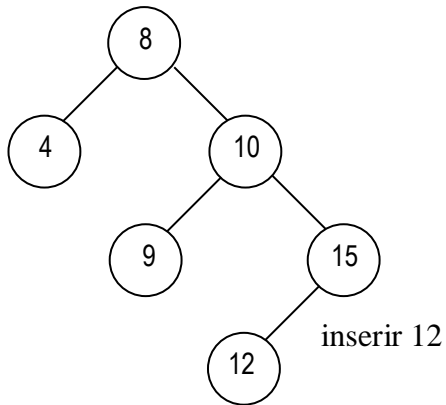
É a altura da subárvore direita do nó menos a altura da subárvore esquerda do nó.

Re-balanceamento:

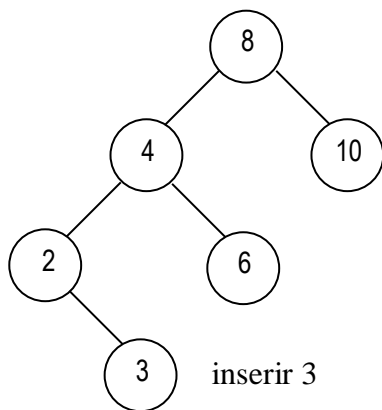
Os problemas podem ser mapeados para dois casos:

Tipo 1: o nó raiz de uma subárvore tem **FB 2 (ou -2)** e tem um filho com **FB 1 (-1)** o qual tem o mesmo sinal que o FB do nó pai. Exemplos:

Exemplo 1:



Exemplo 2:



- solução: rotação simples sobre o nó de $FB = 2$ (-2).
- rotações são feitas à esquerda quando FB positivo e à direita quando FB negativo.

Algoritmo para rotação à direita sobre o nó p

```

procedure rotacao_direita(var p:tree);
var q,temp:tree;
begin
    q:=p^.esq;
    temp:=q^.dir;
    q^.dir:=p;
    p^.esq:=temp;
    p:=q;
end;

```

Algoritmo para rotação à esquerda sobre o nó p

```

procedure rotacao_esquerda(var p:tree);
var q,temp:tree;
begin
  q:=p^.dir;
  temp:=q^.esq;
  q^.esq:=p;
  p^.dir:=temp;
  p:=q;
end;

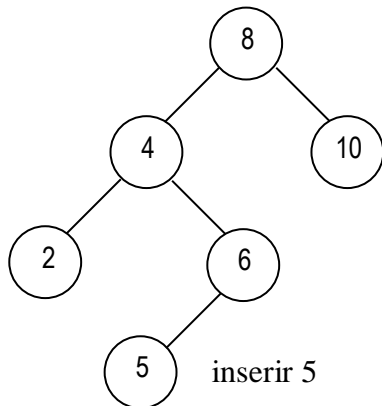
```

Tipo 2: o nó raiz de uma subárvore tem $FB = 2$ (ou -2) e tem um filho com $FB = -1$ (1) o qual tem o sinal oposto ao FB do nó pai.

Exemplo: Caso $(-2) (1)$

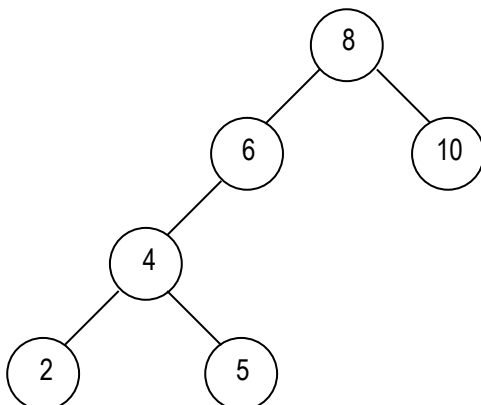
FB do nó que contém 8: -2

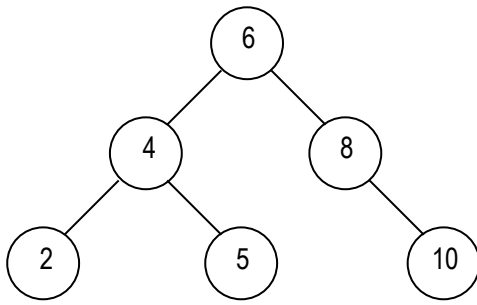
FB do nó que contém 4: 1



- solução: duas rotações.
1. primeiro roda-se o nó com $FB = 1$ (-1) na direção apropriada.
 2. depois roda-se o nó que tinha $FB = -2$ (2) na direção oposta.

Rotação de 4 à esquerda.



Rotação de 8 à direita.

- Elaborar um caso (2) (-1) e executar o re-balanceamento.
- Como seria uma inserção fazendo na árvore fazendo a cada inserção um re-balanceamento (se necessário)?