



# Árvores AVL

---

Jarbas Joaci de Mesquita Sá Junior



# Árvores AVL

---

- Altura de uma árvore binária (AB): igual à profundidade, ou nível máximo, de suas folhas
- A eficiência da busca em árvore depende do seu balanceamento
- Algoritmos de inserção e remoção em ABB não garantem que a árvore gerada a cada passo seja balanceada
- Pergunta: Vale a pena balancear uma ABB de tempos em tempos?

# Árvore Binária de Busca Aleatória

- Nós externos: potenciais descendentes dos nós folha (não estão, de fato, na árvore)
- Uma árvore  $A$  com  $n$  chaves possui  $n+1$  nós externos
- Uma inserção em  $A$  é considerada 'aleatória' se
- ela tem probabilidade igual de acontecer em qualquer nos  $n+1$  intervalos
- Uma ABB aleatória com  $n$  chaves é uma árvore resultante de  $n$  inserções aleatórias sucessivas em uma árvore inicialmente vazia

# Árvore Binária de Busca Aleatória

- Para uma ABB 'aleatória', foi mostrado que o número esperado de comparações para recuperar um registro qualquer é cerca de  $1,39 \cdot \log_2(n)$ .
  - ou seja, 39% pior do que o custo do acesso em uma árvore perfeitamente balanceada
- Isto é: o balanceamento a cada operação aumenta o tempo e garante um desempenho melhor que numa árvore aleatória de, no máximo, 39% (o pior caso é muito raro!)
- Essa estratégia é aconselhável apenas se o número de buscas for muito maior do que o de inserções.
- A conservação do balanceamento pode ser mais simples se relaxarmos a condição de perfeitamente balanceada para balanceada apenas.



# Árvores AVL

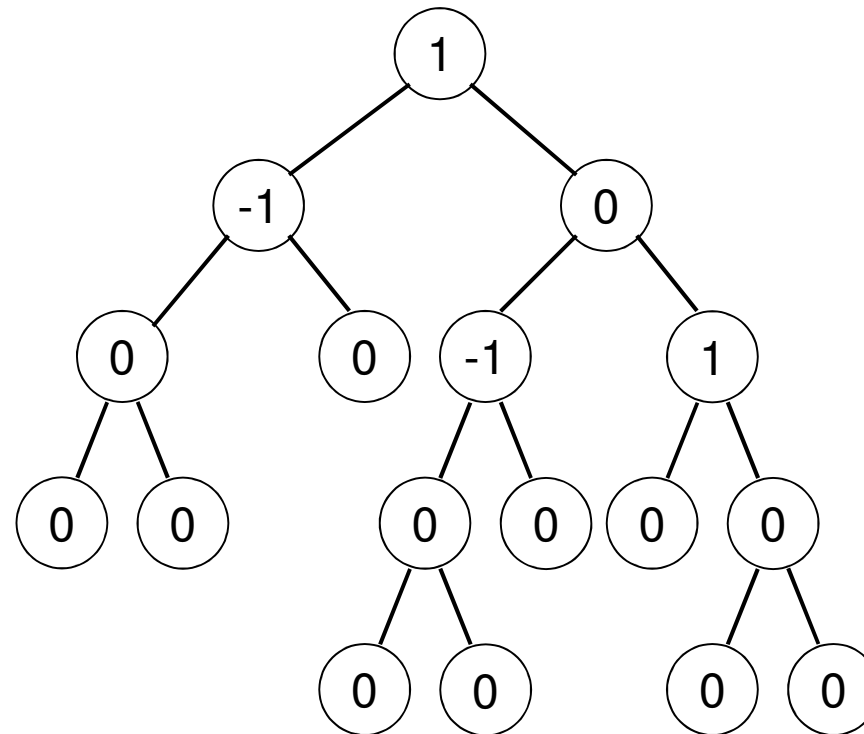
---

- Árvore AVL: ABB na qual as alturas das duas sub-árvores de todo nó nunca diferem em mais de 1. Ou seja, é uma ABB Balanceada.
- Seja o Fator de Balanceamento de um Nó (FB) a altura de sua sub-árvore direita menos a altura de sua sub-árvore esquerda
- Em uma árvore AVL todo nó tem FB igual a 1, -1 ou 0



# Árvores AVL - Exemplo

---





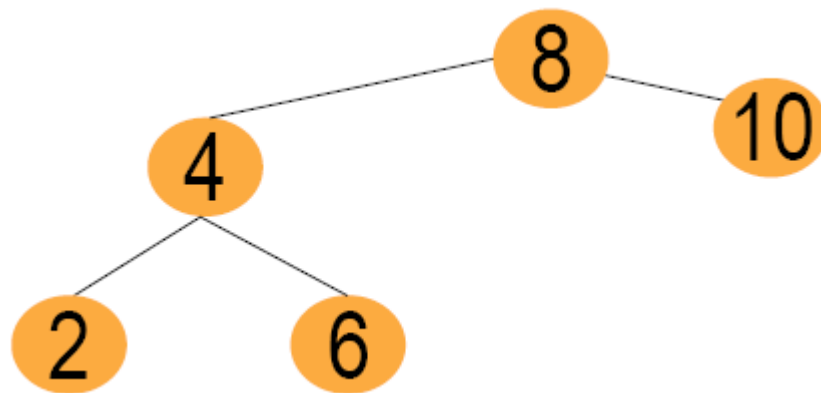
# Árvores AVL

---

- Para manter uma árvore balanceada é necessário aplicar uma transformação na árvore tal que:
  - O percurso in-ordem na árvore transformada seja igual ao da árvore original (isto é, a árvore transformada continua sendo uma ABB);
  - A árvore transformada fique balanceada (todos os nós com  $FB = -1, 0$  ou  $1$ ).

# Árvores AVL

- Numa AVL, a inserção é feita na posição adequada, e depois verifica-se se houve desbalanceamento.
- Veja o que pode acontecer a cada inserção:

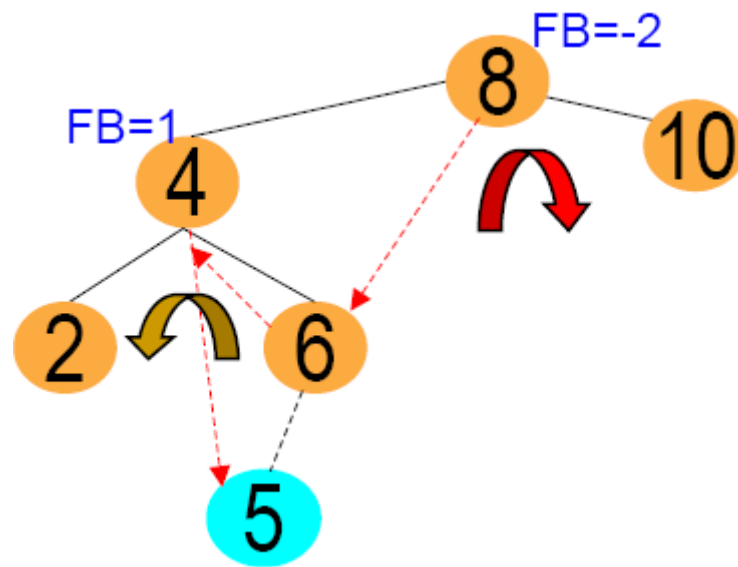


- As chaves 9 e 11 não violam o balanceamento (até melhoram!)
- As chaves 1, 3, 5 ou 7 violam.



# AVL: dois casos de rebalanceamento

- TIPO 1: a raiz de uma sub-árvore tem  $FB=2$  (ou  $-2$ ) e tem um filho com  $FB = -1$  (ou  $1$ ), i.e.  $FB$  com sinal oposto ao do pai.



**Inserção de 5!**

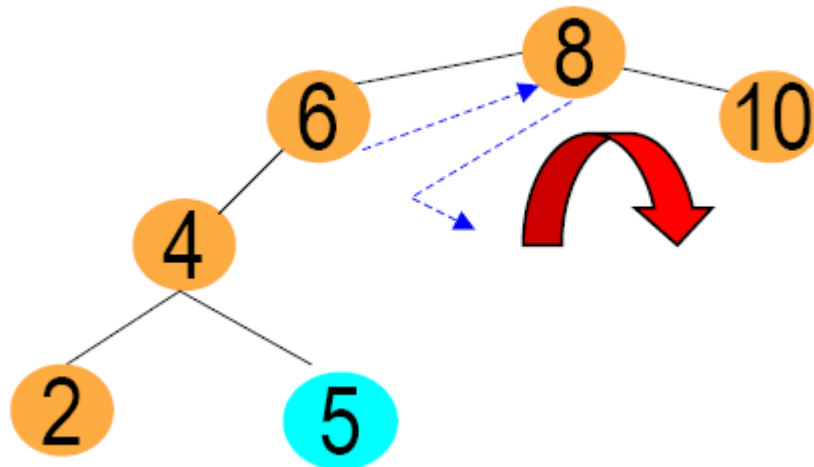
Como rebalancear?

**Solução: 2 Rotações:**

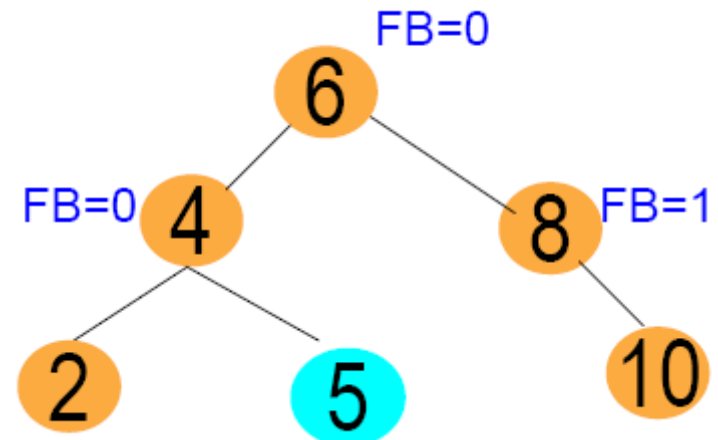
- 1) Rotação à esquerda de 4
- 2) Rotação à direita de 8

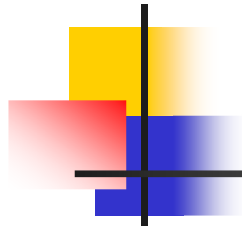
# AVL: rotação esquerda-direita

- Rotação à esquerda de 4



- Rotação à direita de 8





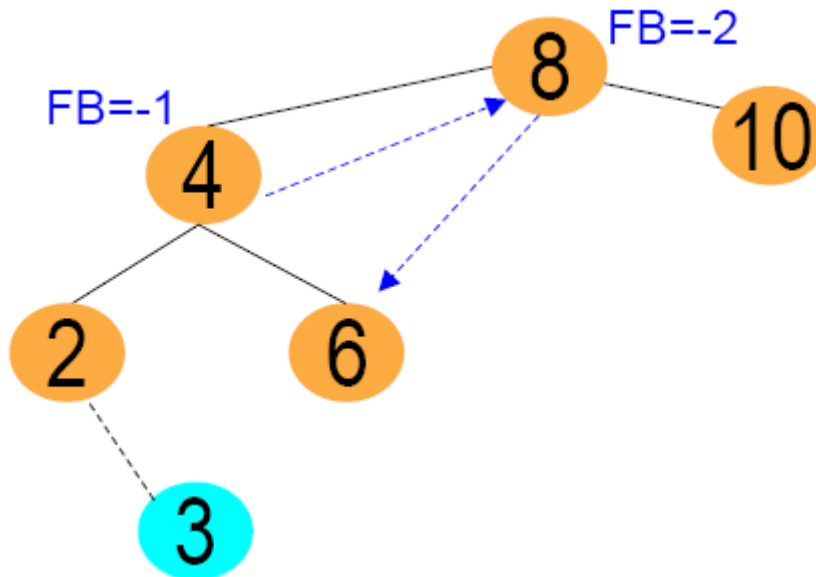
# AVL Tipo 1 - Resumo

---

- Requer uma rotação dupla: ESQUERDA-DIREITA ou DIREITA-ESQUERDA:
  1. Rotacionar o nó com  $FB = -1$  (ou  $1$ ) na direção apropriada, i.e., se  $FB$  negativo, para a direita; se positivo, para a esquerda.
  2. Rotacionar o nó com  $FB = -2$  (ou  $2$ ) na direção oposta.

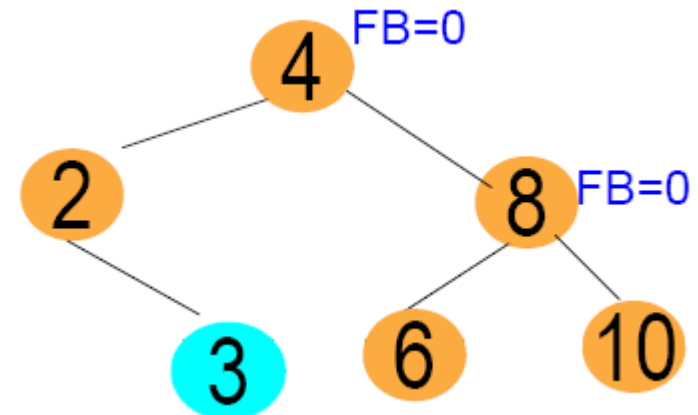
# AVL: dois casos de rebalanceamento

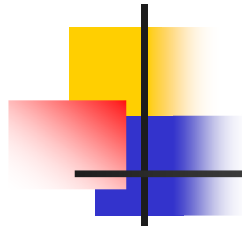
- Tipo 2: A raiz de uma subárvore tem  $FB = -2$  (ou 2) e tem filho com  $FB = -1$  (ou 1), i.e., com mesmo sinal do pai.



Inserção de 3

Solução: Uma Rotação simples à direita do nó 8

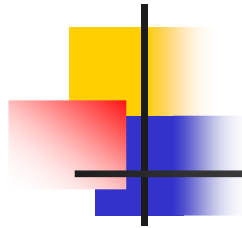




## AVL Tipo 2 - Resumo

---

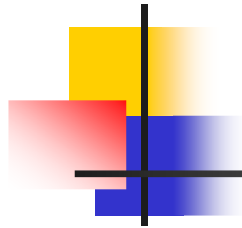
- Rotacionar uma única vez o nó de FB = -2 ou 2:
  - se negativo: à direita;
  - se positivo: à esquerda.



# Inserção em AVL: Resumo

---

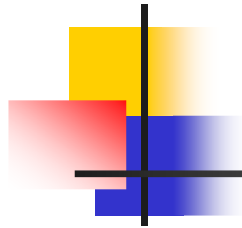
- A cada inserção, verificar se balanceamento foi conservado.
- Em caso negativo (se algum nó ficou com FB igual a 2 ou  $-2$ ), verificar qual caso se aplica (Tipo 1 ou Tipo 2).
- Efetuar as operações de rotação adequadas.



# Remoção em AVL

---

- Primeiro caso
  - Rotação simples em R ( $FB=2$  ou  $-2$ ) com filho com fator de balanceamento de mesmo sinal ( $1$  ou  $-1$ ) ou zero
  - Se R negativo, rotaciona-se para a direita; caso contrário, para a esquerda



# Remoção em AVL

---

- Segundo caso
  - Rotação dupla quando R ( $FB=2$  ou  $-2$ ) e seu filho ( $1$  ou  $-1$ ) têm fatores de balanceamento com sinais opostos
  - Rotaciona-se o filho para o lado do desbalanceamento do pai
  - Rotaciona-se R para o lado oposto do desbalanceamento



# Pseudo-Código: Rotações Simples

## ■ Rotação Simples a Esquerda

**p** aponta para o nó desbalanceado

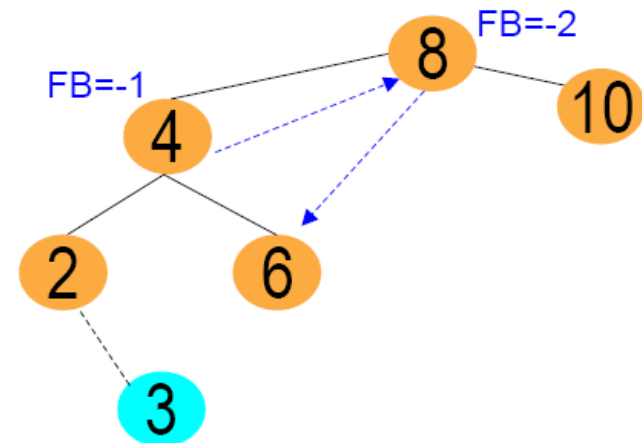
**q** = **right**(p);

hold = **left**(q);

**left**(q) = p;

**right**(p) = hold;

p = q;



## ■ Rotação Simples a Direita

**p** aponta para o nó desbalanceado

**q** = **left**(p);

hold = **right**(q);

**right**(q) = p;

**left**(p) = hold;

p = q;

