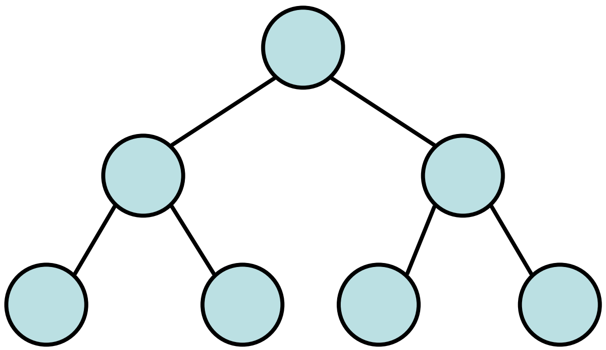
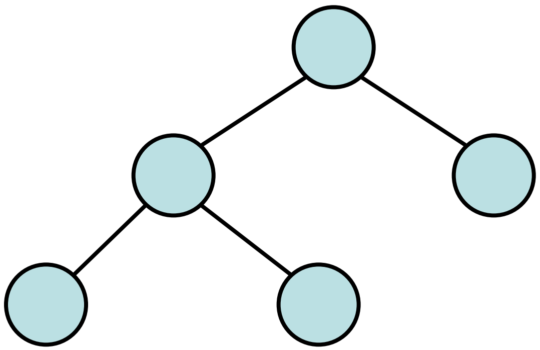
Árvores Balanceadas

- As árvores binárias de pesquisa são, em alguns casos, pouco recomendáveis para as operações básicas (inserção, remoção e busca)
- Árvores binárias de pesquisa degeneradas tornam as operações básicas lentas O(n)

Árvores Balanceadas

- Árvore binária completamente balanceada
  - Ocorre quando a árvore está cheia ou quase cheia com o nível n-1 completo

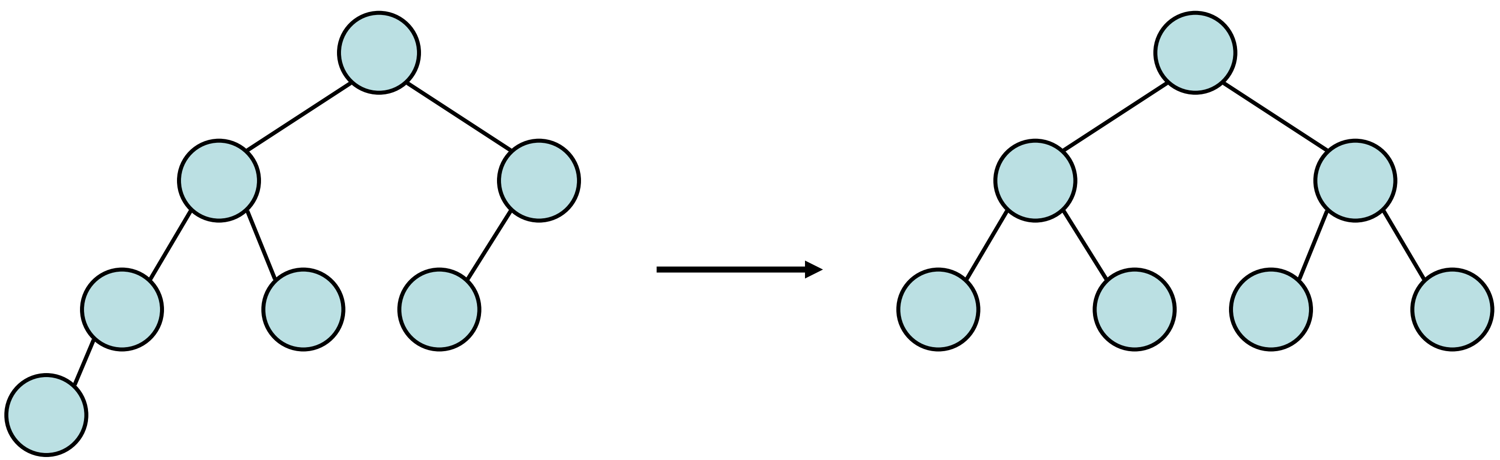




  - Uma árvore binária completa leva um tempo na ordem de O(log n) para operações de inserção, remoção e pesquisa. O que é, sem dúvida, muito bom

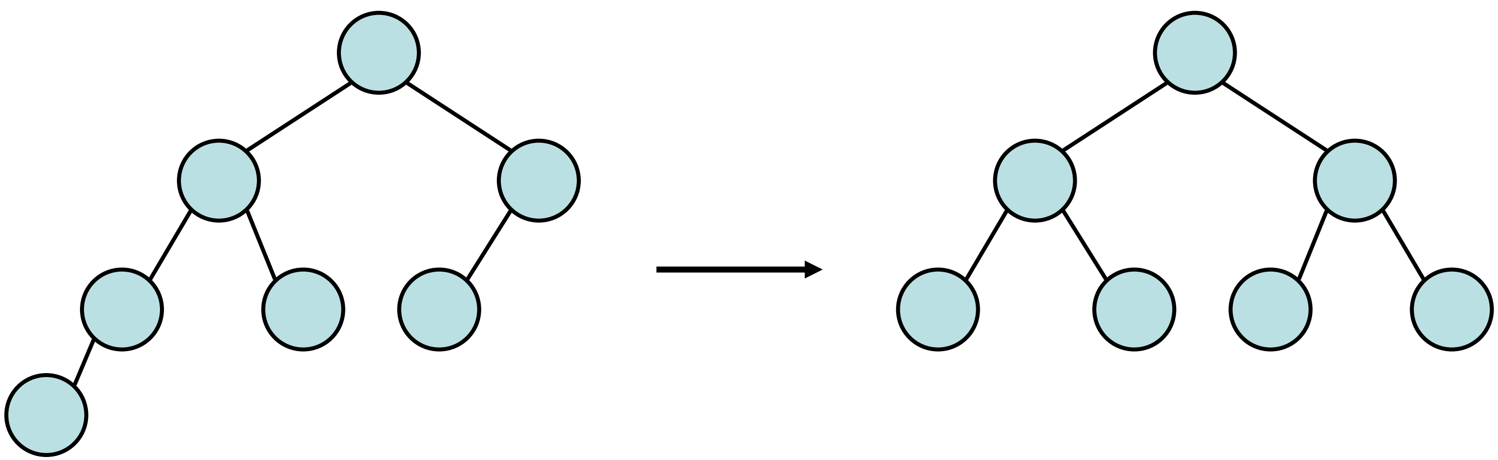
Árvores Balanceadas

- Árvore binária completamente balanceada
  - Após uma inserção ou remoção a árvore pode deixar de ser completa. A solução seria aplicar um algoritmo que tornasse a árvore novamente completa, porém o custo para realizar está operação seria de O(n)



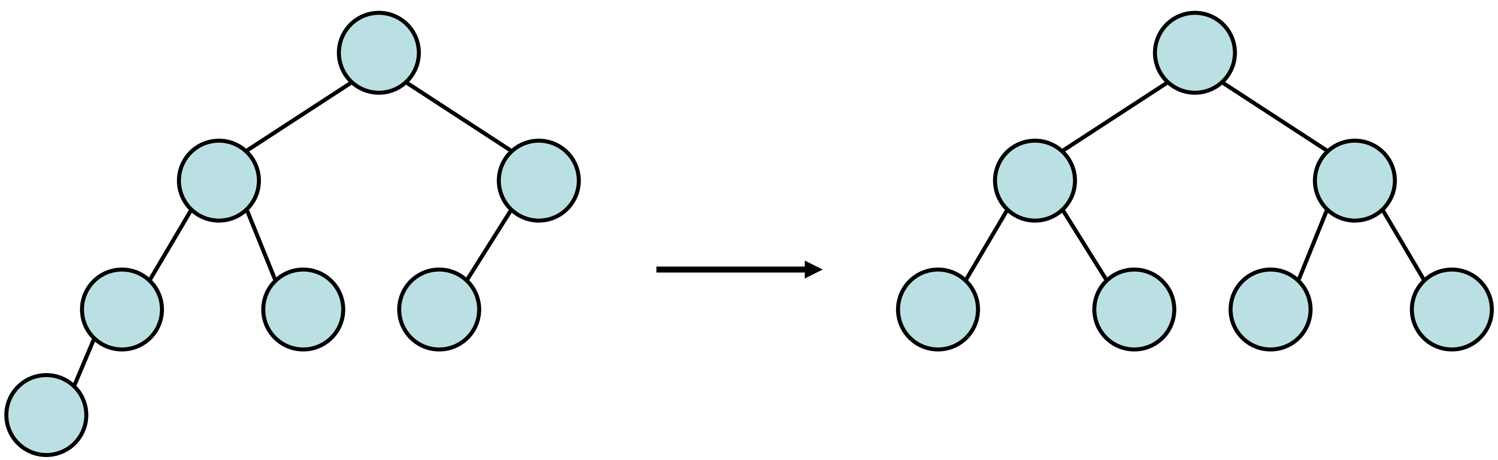
Árvores Balanceadas

- Árvore binária completamente balanceada
  - Todos os nós tiveram sua posição na estrutura alterados
  - Na maioria dos casos, utiliza-se árvores quase balanceadas



Critérios para definir balanceamento

- Vários são os critérios (métodos) para definir balanceamento. Alguns são:
  - Restrições imposta na diferença das alturas das subárvores de cada nó. Ex. AVL
  - Todos os nós folhas no mesmo nível



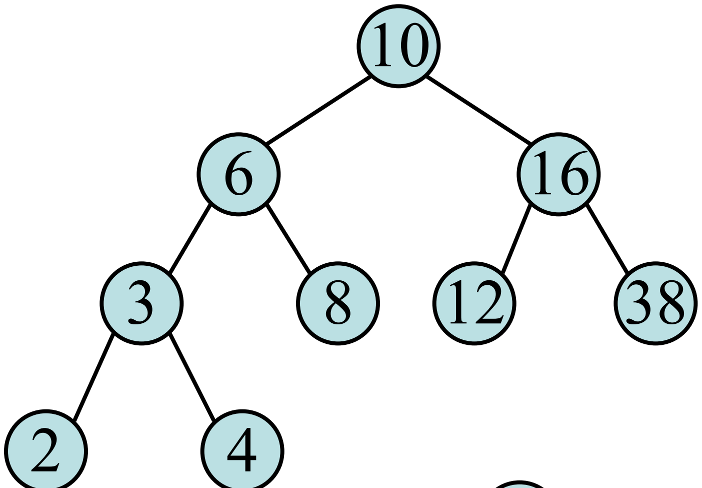
Árvores AVL

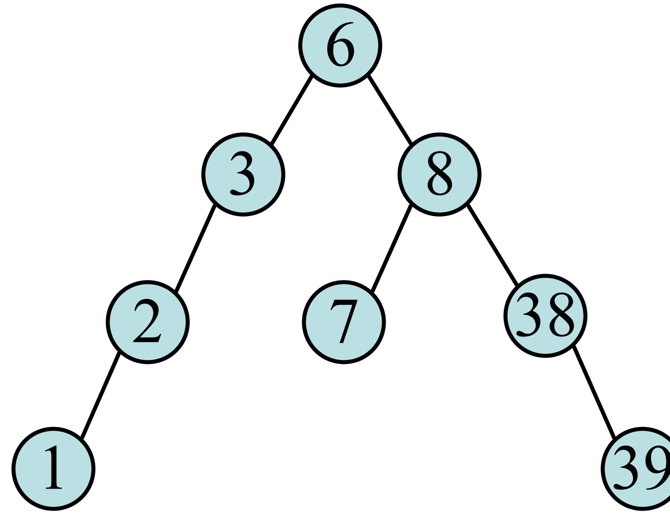
- Foram introduzidas por Adel`son-Vel´skii e landis em 1962
- São baseadas em árvore binárias de pesquisa
- A medida em que as operações de inserção e remoção são efetuadas a árvore é balanceada

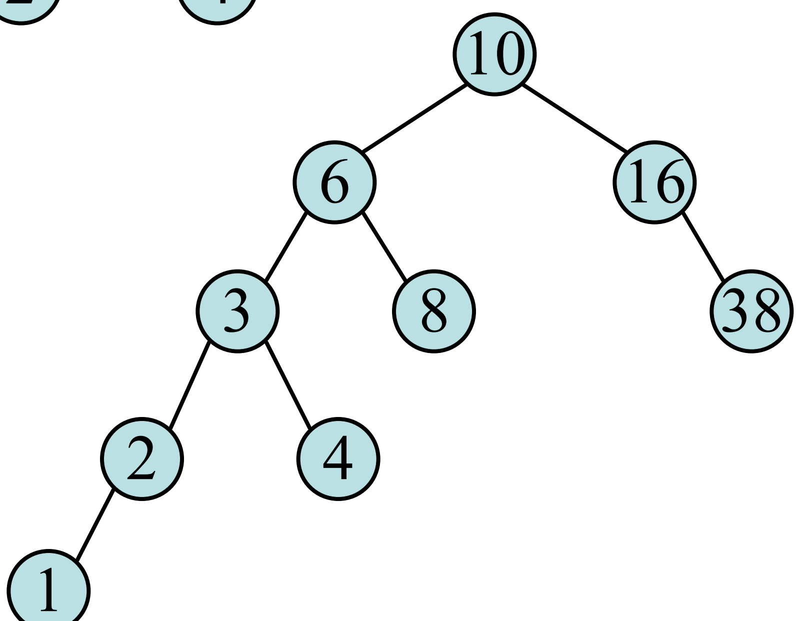
Árvores AVL

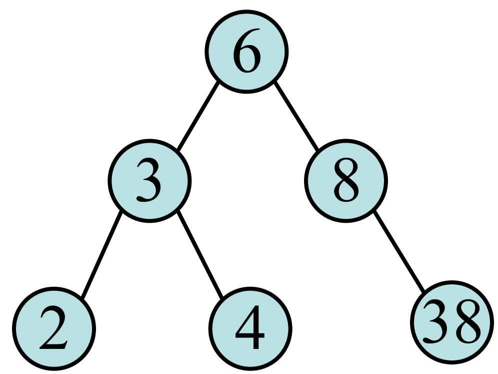
- Definição:
  - Uma árvore binária T é dita AVL quando, para qualquer nó v de T, a diferença entre a altura das subárvores esquerda $h_e(v)$ e direita $h_d(v)$ é no máximo em módulo igual a 1.

Árvores AVL









Árvores AVL

OBS.: se uma árvore T é dita AVL, então todas as suas subárvores também são AVL

Árvores AVL

- Balanceamento de um nó
  - O fator de balanceamento:
    - É dado pela altura da subárvores da esquerda $h_e(v)$ menos a altura da subárvore da direita $h_d(v)$.

$$\boxed{FB(v)= h_e(v) - h_d(v)}$$

Árvores AVL

- Nós balanceados
  - São aqueles onde os valores de FB são -1, 0 ou 1

- FB(v):
  - +1: subárvore esquerda mais alta que a direita
  - 0: subárvore esquerda igual a direita
  - -1: subárvore direita mais alta do que a esquerda

Árvores AVL

- Nós desregulados ou desbalanceados
  - São aqueles onde os valores de FB são diferentes de -1, 0 ou 1

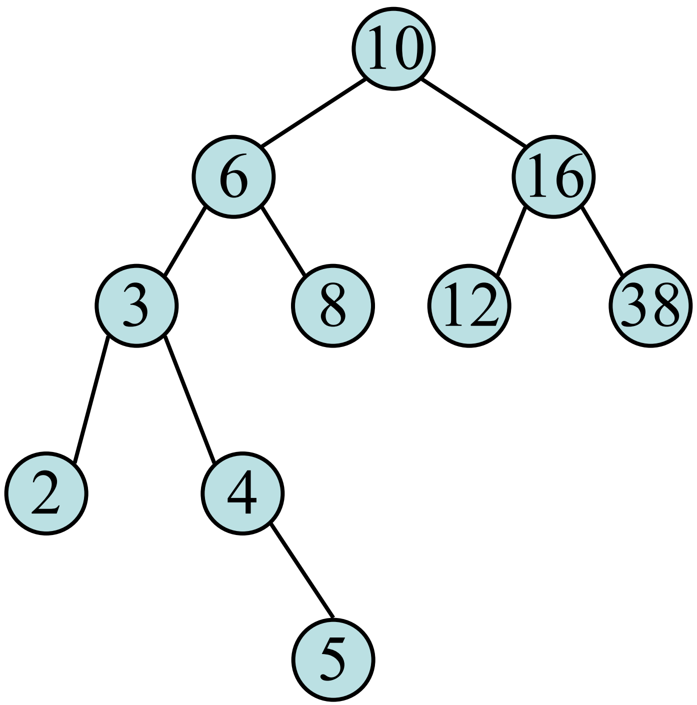
- FB(v):

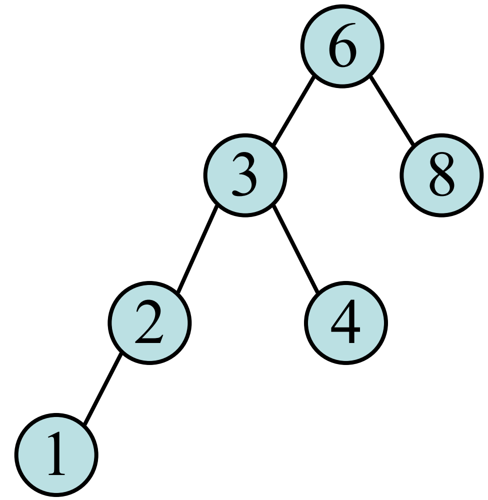
  >1: subárvore esquerda está desbalanceando o nó v

  <-1: subárvore direita está desbalanceando o nó v

Árvores AVL

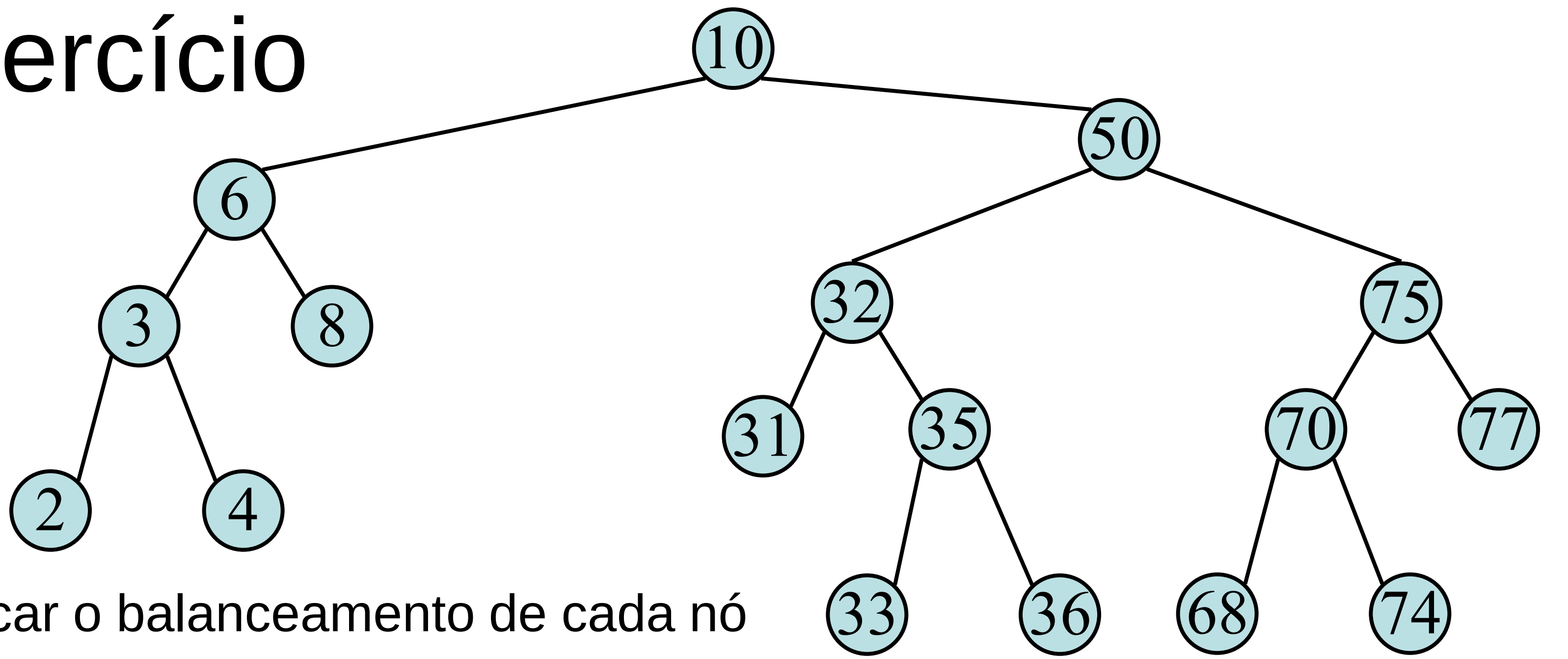
- Exemplos





Árvores AVL

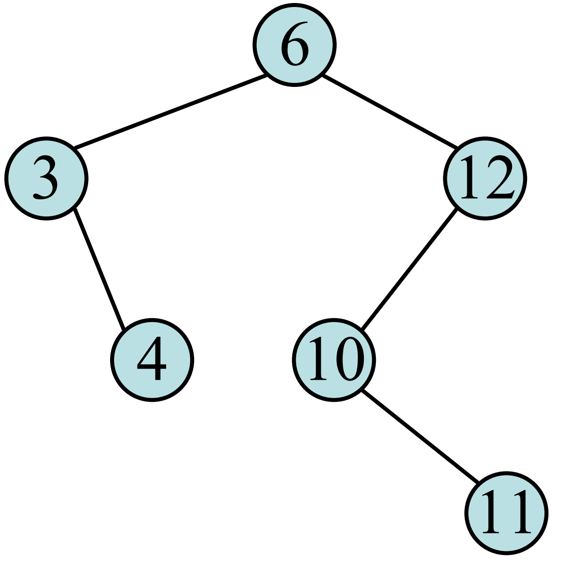
- Exercício



- Colocar o balanceamento de cada nó
- Dizer se a árvore é AVL
- Verificar quais as possíveis posições para a inserção de elementos e em quais posições de inserção, a árvore é AVL

Árvores AVL

- Exercício2



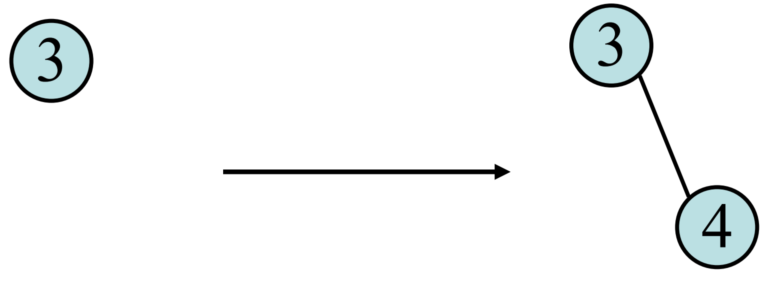
- Colocar o balanceamento de cada nó
- Dizer se a árvore é AVL
- Verificar quais as possíveis posições para a inserção de elementos e em quais posições de inserção, a árvore é AVL

Árvores AVL

- Verificando a ocorrência do desbalanceamento de um nó

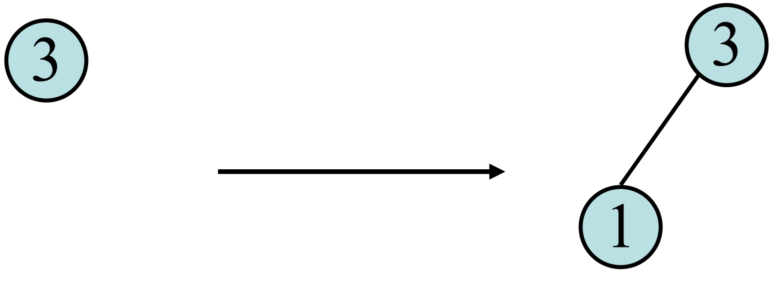
Árvores AVL

- Verificando a ocorrência do desbalanceamento de um nó
  - Quando Ocorre?
    - Se um nó tem FB(v)=0 e é feita uma inserção no lado direito, o FB=-1, ou seja, subtrai uma unidade (na remoção é invertido)



Árvores AVL

- Verificando a ocorrência do desbalanceamento de um nó
  - Quando Ocorre?
    - Se um nó tem FB(v)=0 e é feita uma inserção no lado esquerdo, o FB=1, ou seja, soma uma unidade(na remoção é invertido)



Árvores AVL

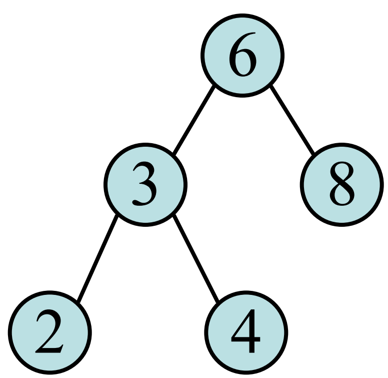
- Resumo

	ArvEsq	ArvDir
Inserção	+1	-1
Remoção	-1	+1

Árvores AVL

- Atualização do FB dos antecessores

	ArvEsq	ArvDir	Critério(atualiza FB antecessor e aplica regra abaixo)
Inserção	+1	-1	Se FB(Vantecessor)==0 pare
Remoção	-1	+1	Se FB(Vantecessor)!=0 pare



Árvores AVL

- Rebalanceando nós desregulados
  - Quando uma inserção ou remoção realizada em um nó altera o balanceamento da árvore, é necessário efetuar uma transformação na árvore, tal que:
    - O percurso em ordem fique inalterado em relação a árvore desbalanceada. Isto é, a árvore continua a ser uma árvore binária de pesquisa
    - A árvore transformada saiu de um estado de desbalanceamento para um estado de balanceamento

Árvores AVL

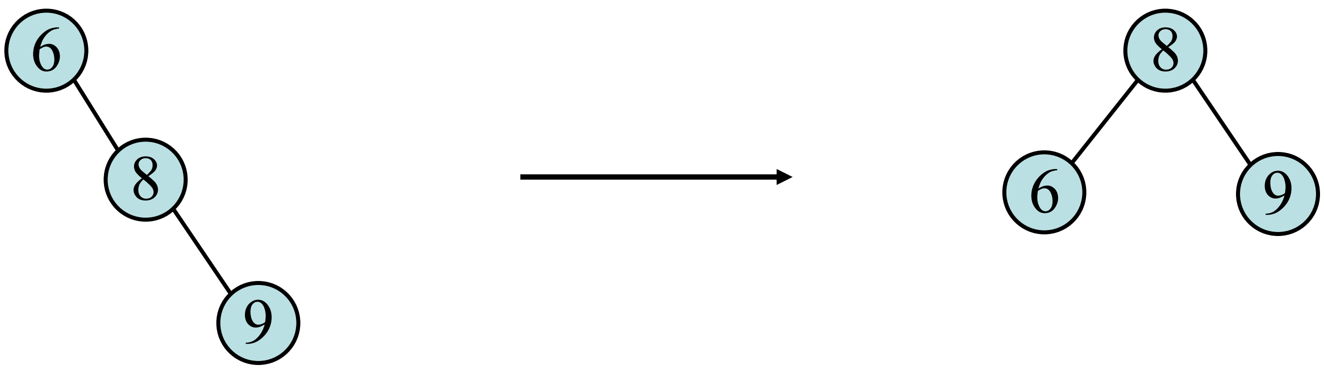
- Rotações
  - Operação que altera o balanceamento de uma árvore T, mantendo a seqüência de percurso em-ordem

Árvores AVL

- Rotações
  - Tipos de rotações
    - Esquerda Simples
    - Direita Simples
    - Esquerda Dupla
    - Direita Dupla

Árvores AVL

- Rotação Esquerda Simples (RES)



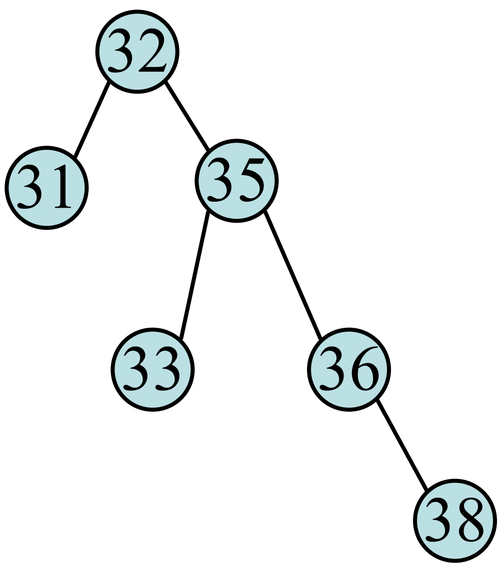
Percurso em ordem: 6, 8 e 9

Percurso em ordem: 6, 8 e 9

- Após a rotação a esquerda a árvore ficou balanceada e o percurso em-ordem permanece o mesmo

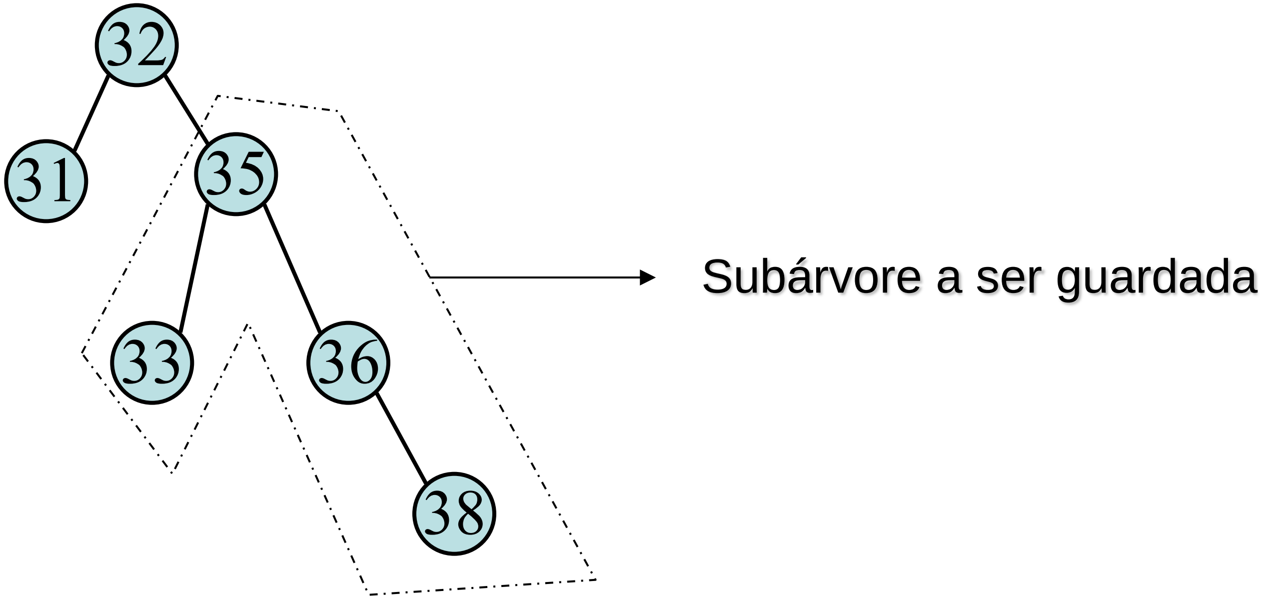
Árvores AVL

- Exemplo Rotação Esquerda Simples



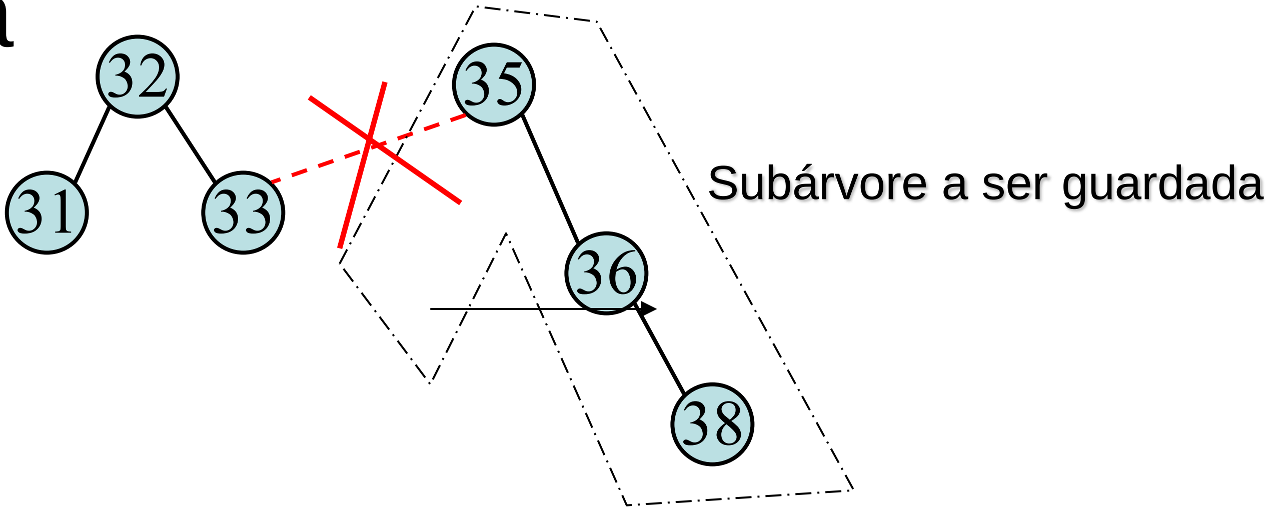
Árvores AVL

- Passos para efetuar a RES
  - Guarde a subárvore direita



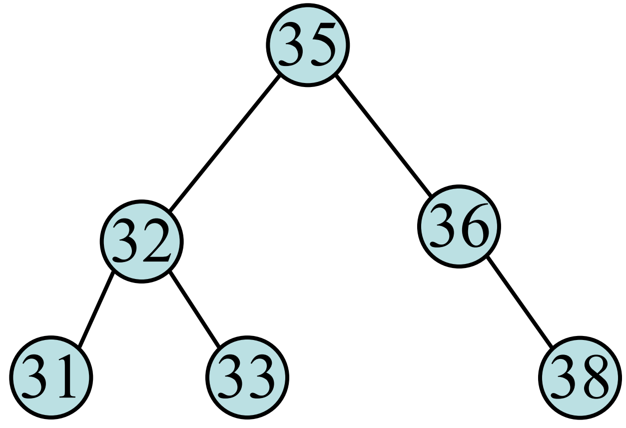
Árvores AVL

- Passos para efetuar a RES
  - Troque a subárvore guardada pela subárvore esquerda da árvore guardada



Árvores AVL

- Passos para efetuar a RES
  - Ponha na subárvore esquerda da subárvore guardada a árvore restante
  - verifique o balanceamento

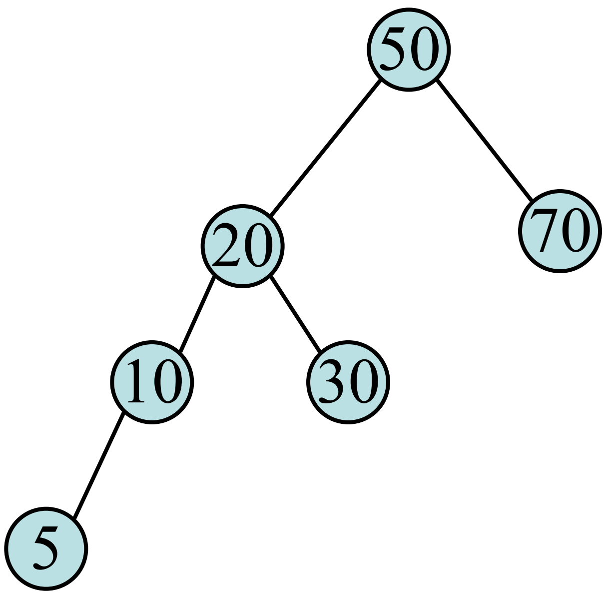


Árvores AVL

- Rotação Simples a Direita(RSD)
  - A rotação a direita simples é simétrica a rotação esquerda simples
  - Os quatro passos realizados na rotação esquerda simples se aplicam da mesma forma à rotação direita simples

Árvores AVL

- Rotação Simples a Direita(RSD)
  - Exemplo

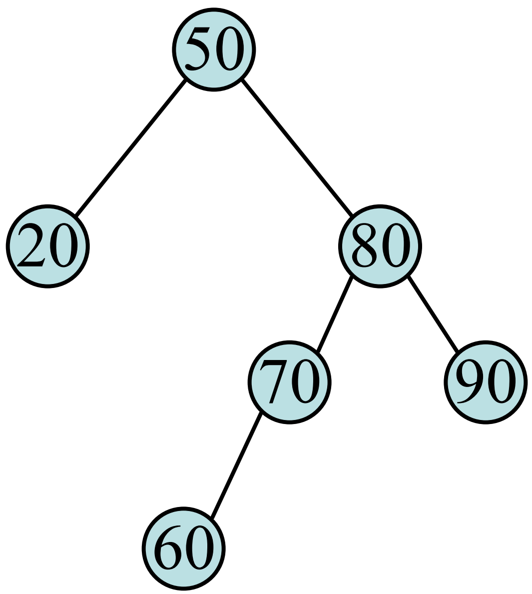


Árvores AVL

- Rotação Dupla a Esquerda(RDE)
  - Passos:
    - Efetua-se uma rotação simples direita na subárvore direita do nó desbalanceado
    - Realiza-se uma rotação simples esquerda no nó desbalanceado

Árvores AVL

- Rotação Dupla a Esquerda(RDE)
  - Exemplo:

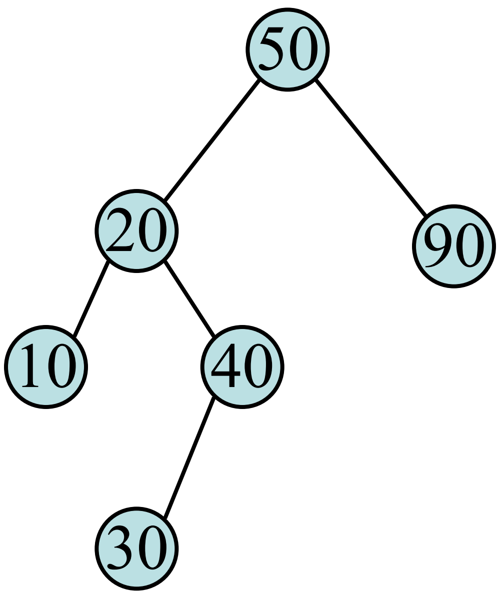


Árvores AVL

- Rotação Dupla a Direita(RDD)
  - É simétrica a rotação esquerda dupla
  - Efetuar uma rotação simples esquerda na subárvore esquerda do nó desbalanceado
  - Realizar uma rotação simples direita no nó desregulado

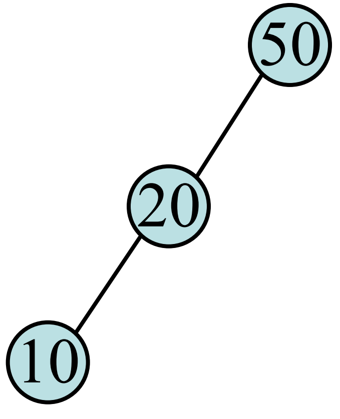
Árvores AVL

- Rotação Dupla a Direita(RDD)
  - Exemplo:



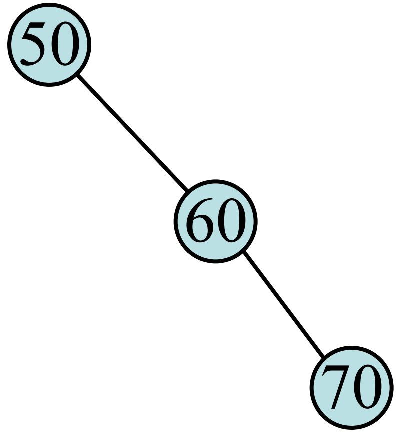
Árvores AVL

- Quando fazer Rotações
  - Quando uma árvore ou subárvore tem um fator de balanceamento FB=2, deve-se fazer uma rotação a direita



Árvores AVL

- Quando fazer Rotações
  - Quando uma árvore ou subárvore tem um fator de balanceamento FB=-2, deve-se fazer uma rotação a esquerda

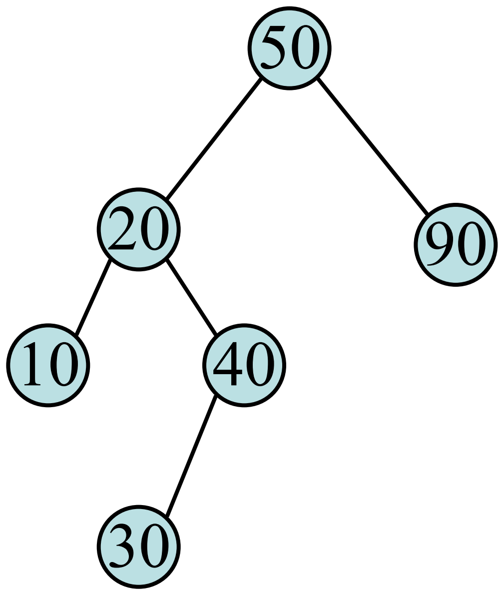


Árvores AVL

- Quando fazer Rotações
  - Quando uma árvore ou subárvore tem um fator de balanceamento FB=2 e sua subárvore esquerda tem um FB>=0, faz-se uma rotação direita simples. Caso o FB<0 na subárvore esquerda do nó desregulado uma rotação dupla direita é necessária.

Árvores AVL

- EX.:



Árvores AVL

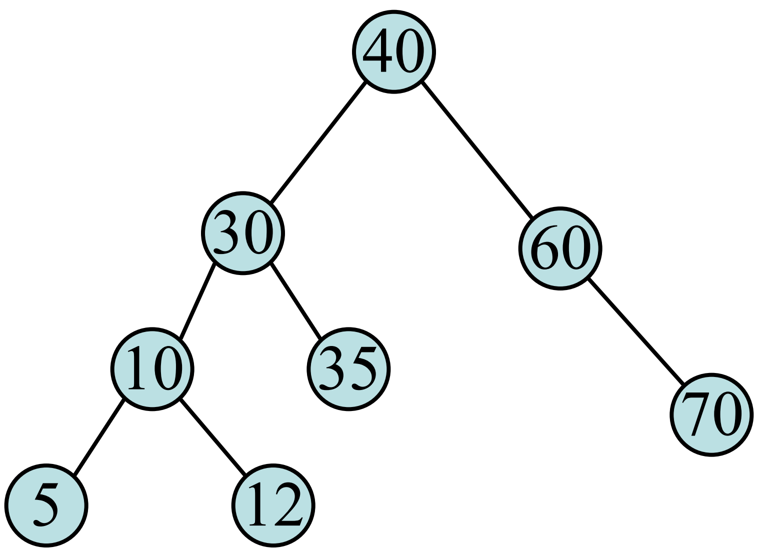
- Quando fazer Rotações
  - Quando uma árvore ou subárvore tem um fator de balanceamento FB=-2 e sua subárvore direita tem um FB<=0, faz-se uma rotação esquerda simples. Caso o FB>0 na subárvore direita do nó desbalanceado uma rotação dupla esquerda é necessária.

Árvores AVL

- Inserção de elementos
  - Procedimentos: percorrer a árvore até o ponto de inserção (usando a operação de busca)
  - Inserir o novo elemento
  - Balancear a árvore (quando necessário fazer rotações)

Árvores AVL

- Exemplo
  - Inserir na árvore AVL abaixo os seguintes elementos: 3,33,11 e 9



Árvores AVL

- Exemplo
  - Inserir na árvore AVL inicialmente vazia os seguintes elementos: 10,20,30,40,50,25,60,70,80 e 90

Árvores AVL

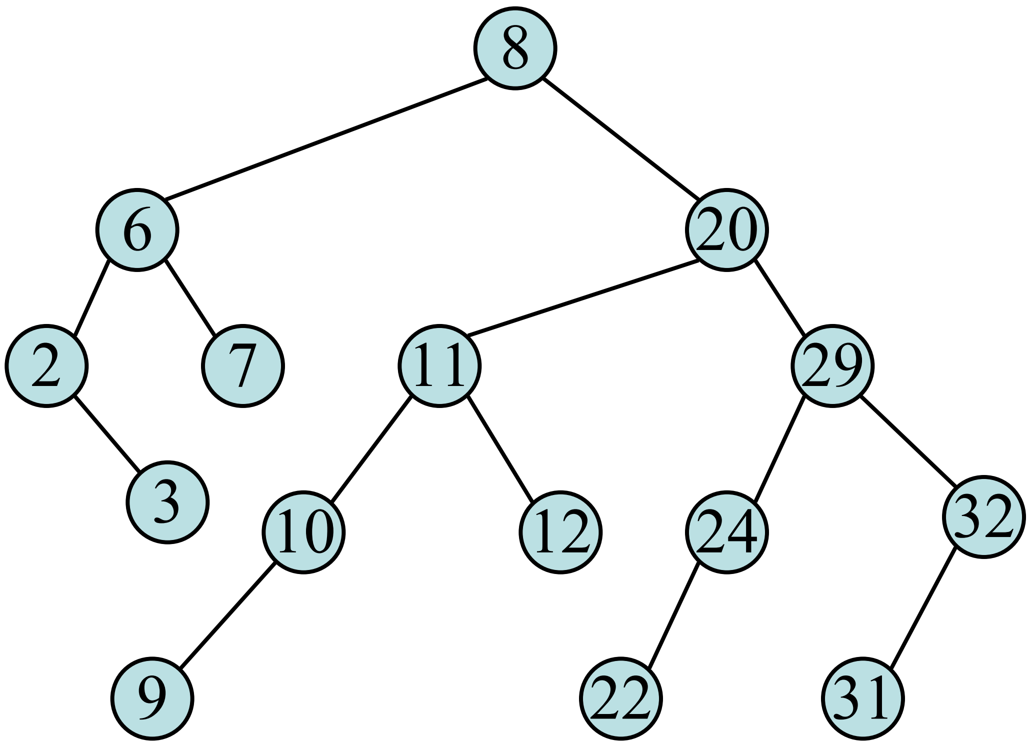
- Exemplo
  - Inserir na árvore AVL inicialmente vazia os seguintes elementos: 10,20,30,40,50,25,60,70,80 e 90

Árvores AVL

- Remoção de Elementos
  - Procedimentos
    - Percorrer a árvore até o nó a ser removido (usando a operação de busca)
    - Retirar o elemento (igual a árvore binária de pesquisa)
    - Balancear a árvore (quando necessário fazer rotação)

Árvores AVL

- Exemplo: remover 22,31,12,7 e 20



Árvores AVL

- Ex2: remover: 40,25,50,10,35,30,20,70 e 60