CI1057: Algoritmos e Estruturas de Dados III
Árvores Binárias Balanceadas - Árvores AVL

Profa. Carmem Hara

Departamento de Informática/UFPR

21 de março de 2024

Relembrando as Operações de Inserção e Remoção em Árvores Binárias de Busca

- inserção dos valores: 50 70 30 10 20 80 100 90
- remoção do elemento 70

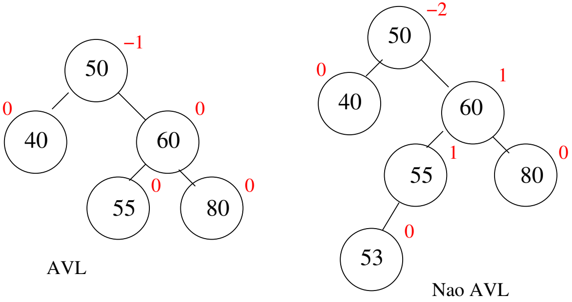
Árvore de Busca "Degenerada"

- inserção dos valores: 10 20 30 40 50 60 70 80 90
- remoção do elemento 70

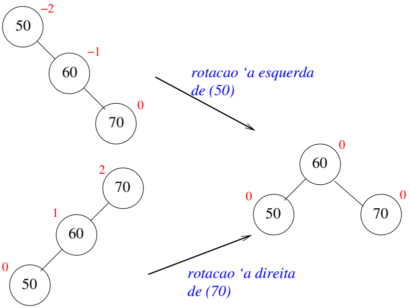
- Custo da busca: $O(n)$
- Em uma árvore balanceada: $O(lg(n))$

Árvore AVL

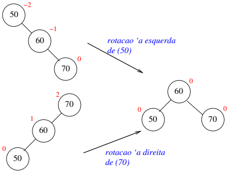
- Árvore de Adelson, Velskii e Landis
- Abordagem: manter a árvore balanceada após cada operação
- Balancamento de um nó n:
  $bal(n)$ = altura da subárv. esq. $-$ altura da subárv. direita
- Árvore AVL: todos os nós tem balanceamento -1, 0 ou 1



Técnica para manter o balanceamento: Rotações

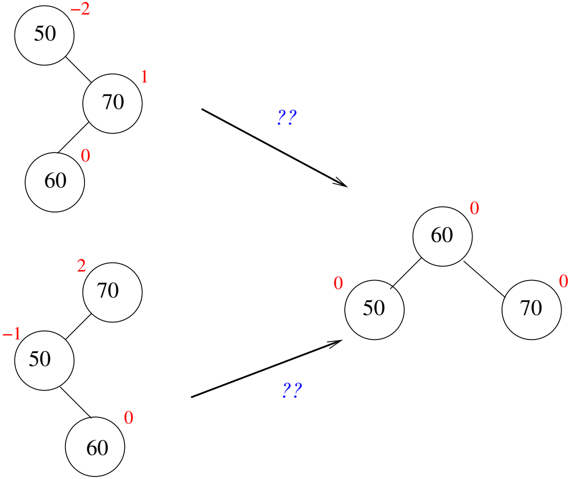


Rotações

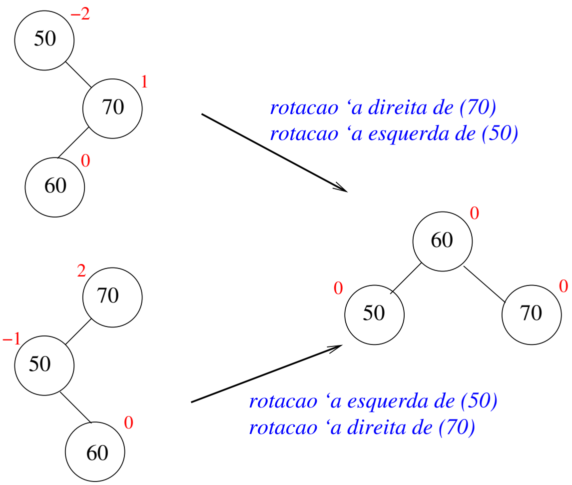


- a altura da árvore diminui de 1
- rotação à direita de *n*:
  - aumenta a altura da subárvore direita
  - o balanceamento todos os seus ancestrais de *n* (depois da rotação) diminui de 1
- rotação à esquerda de *n*:
  - aumenta a altura da subárvore esquerda
  - o balanceamento todos os seus ancestrais de *n* (depois da rotação) aumenta de 1

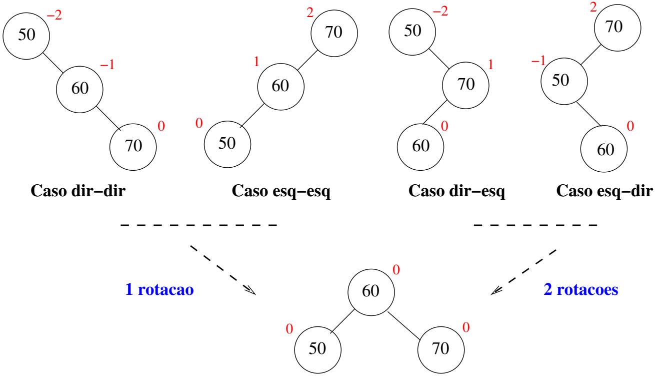
Técnica para manter o balanceamento: Rotações



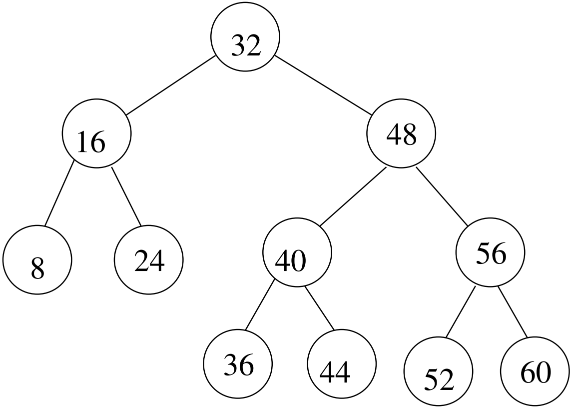
Técnica para manter o balanceamento: Rotações



Técnica para manter o balanceamento: Casos

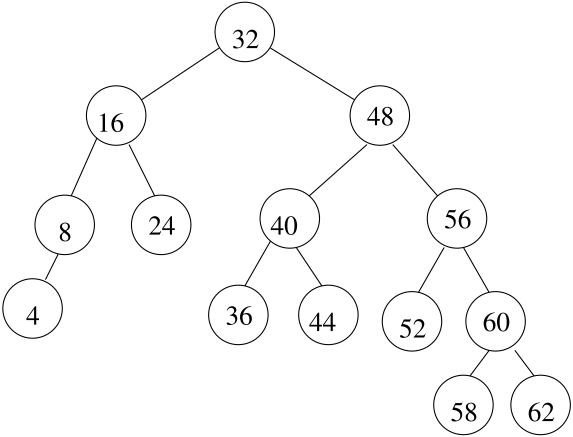


Inserção na AVL - Exemplo 1



Inserção de (46)

Inserção na AVL - Exemplo 2

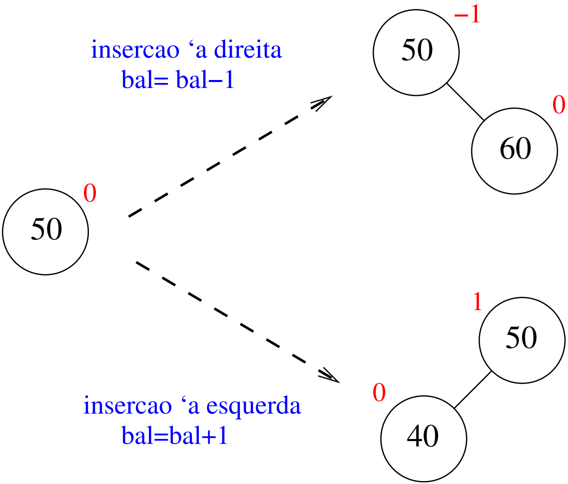


Inserção de (61)

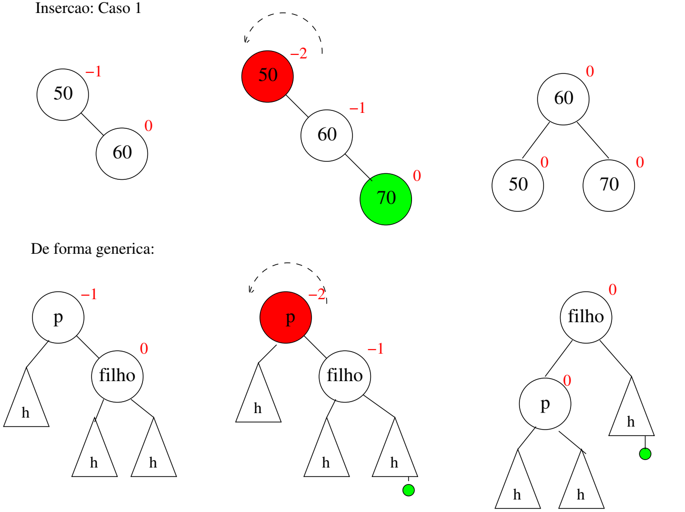
Inserção na AVL

```
1.  insere n na árvore binária
2.  altera o balanceamento dos ancestrais (a) de n:
        se chave(n) < chave(a)
            bal(a) = bal(a) + 1
        senao
            bal(a) = bal(a) - 1
        se o ancestral a ficar desbalanceado
        então balancear a
3.  se o balanceamento de a tornar-se zero
    então não alterar mais os balanceamentos dos seus ancestrais
```

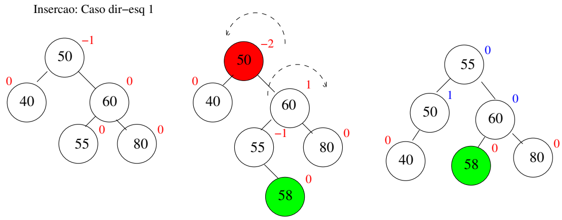
Inserção: sem necessidade de balanceamento

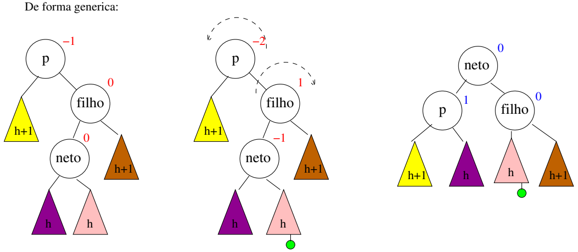


Inserção com balanceamento: Caso dir−dir

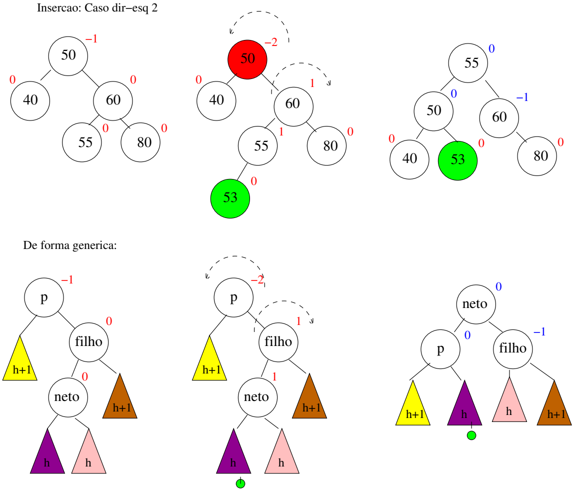


Inserção com balanceamento: Caso dir-esq - Exemplo 1





Inserção com balanceamento: Caso dir-esq - Exemplo 2



Balanceamento quando *bal*(*a*) é -2

```
f = direita(a)
se bal(f) == -1
    rotacaoEsquerda(a)
    bal(a) = bal(f) = 0
senao se bal(f) == 1
    neto = esquerda(f)
    rotacaoDireita(f)
    rotacaoEsquerda(a)
    se bal(neto) == 1
        bal(a)= 0
        bal(f) = -1;
    senao se bal(neto) == -1
        bal(a) = 1;
        bal(f) = 0;
    bal(neto) = 0;
```

Inserção com balanceamento

Como são os balanceamentos nos casos:

- esq-esq ?
- esq-dir ?

Balanceamento quando bal(a) é +2

```
f = esquerda(a)
se bal(f) == 1
     rotacaoDireita(a)
     bal(a) = bal(f) = 0
senão se bal(f) == -1
     neto = direita(f)
     rotacaoEsquerda(f)
     rotacaoDireita(a)
     se bal(neto) == 1
          bal(a) = -1;
          bal(f) = 0;
     senão se bal(neto) == -1
          bal(a) = 0;
          bal(f) = 1;
     bal(neto) = 0
```

Quando parar a atualização de balanceamentos

- Quando o balanceamento de um ancestral *a* tornar-se zero
- A altura de *a* não muda após a inserção:
  - $bal(a) == 1 \quad \longrightarrow \quad bal(a) == 0$
    inserção na subárvore direita e ela passou a ter a mesma altura que a subárvore esquerda
  - $bal(a) == -1 \quad \longrightarrow \quad bal(a) == 0$
    inserção na subárvore esquerda e ela passou a ter a mesma altura que a subárvore direita
  - $bal(a) == 1 \quad longrightarrow \quad bal(a) == 2$
    inserção na subárvore esquerda e depois do balanceamento a altura final da subárvore esquerda passou a ser a mesma que a altura original, antes da inserção
  - $bal(a) == -1 \quad longrightarrow \quad bal(a) == -2$
    inserção na subárvore direita e depois do balanceamento a altura final da subárvore direita passou a ser a mesma que a altura original, antes da inserção

Implementação da inserção na árvore AVL

- `Insere( valorDaChave, nodoAtual, mudouAltura )` retorna apontador para a subárvore com o `valorDaChave` inserido, caso ela ainda não exista na árvore
- `mudouAltura`:
  `true`: se mudou a altura da subárvore com raiz no `nodoAtual` `false`: caso contrário

```c
Avl Insere(ItemAvl valor, Avl p, int *mudouAltura){
  if( p == NULL ){
    *mudouAltura = TRUE;
    return criaNo( valor, NULL, NULL );
  }
  if( valor == p->item){
    *mudouAltura = FALSE;
    return p;
  }
  if( valor < p->item ){
    p->esq = Insere( valor, p->esq, mudouAltura );
    if( *mudouAltura ){
      p->bal++;
      if( p->bal == 2)
         balanceia( &p );
      if( p->bal == 0 )
         *mudouAltura = FALSE;
    }
  } else {
    p->dir = Insere( valor, p->dir, mudouAltura );
    if( *mudouAltura ){
      p->bal--;
      if( p->bal == -2)
        balanceia( &p );
      if( p->bal == 0 )
        *mudouAltura = FALSE;
    }
  }
  return p;
}
```