

# Algoritmos e Estrutura de Dados II (AE23CP-3CP)

## Aula #12 - Árvores AVL

Prof<sup>a</sup> Luciene de Oliveira Marin  
lucienemarin@utfpr.edu.br

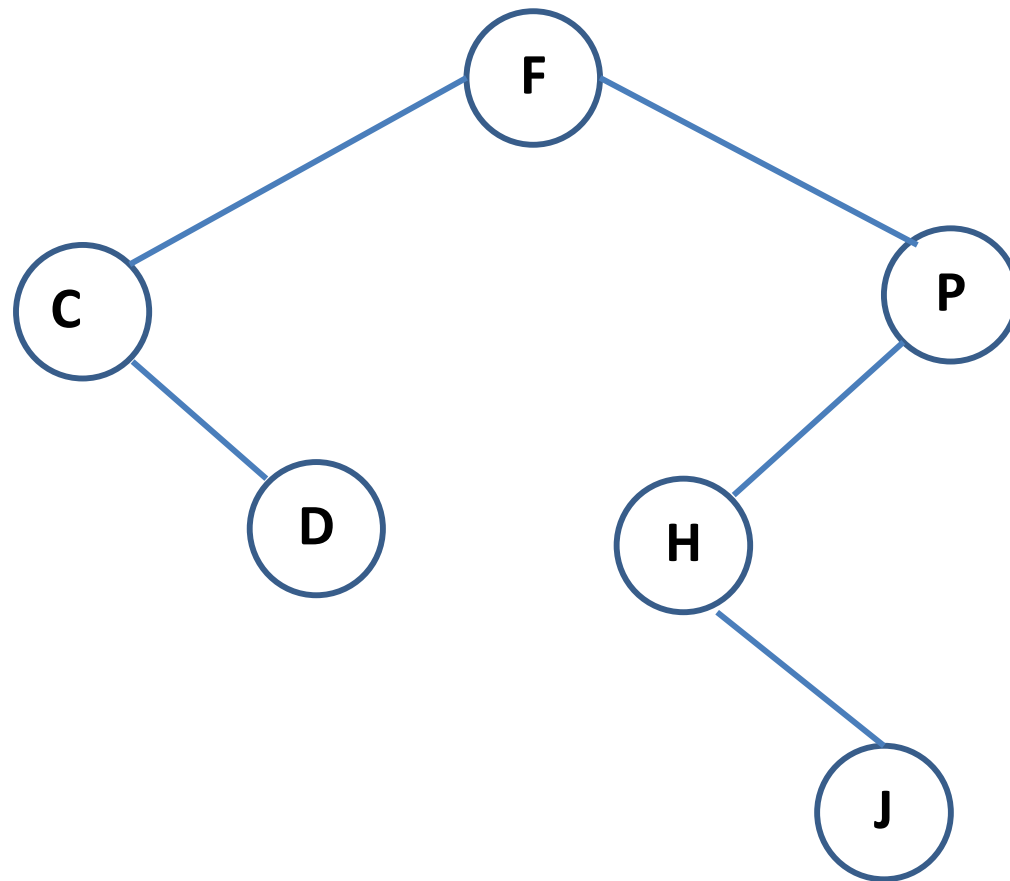
# Árvores AVL

# Árvores AVL - Definição

- Adelson-Velskii e E. M. Landis (1962).
- Constitui uma **árvore binária de pesquisa** que é **balanceada** com relação às suas subárvores.
- Uma árvore AVL é dita balanceada quando a diferença entre as alturas das suas subárvores diferem no máximo em uma unidade ou
  - “Para cada nó interno  $v$  de uma árvore  $T$ , as alturas dos filhos de  $v$  podem variar em no máximo 1”.

# Exemplo 1

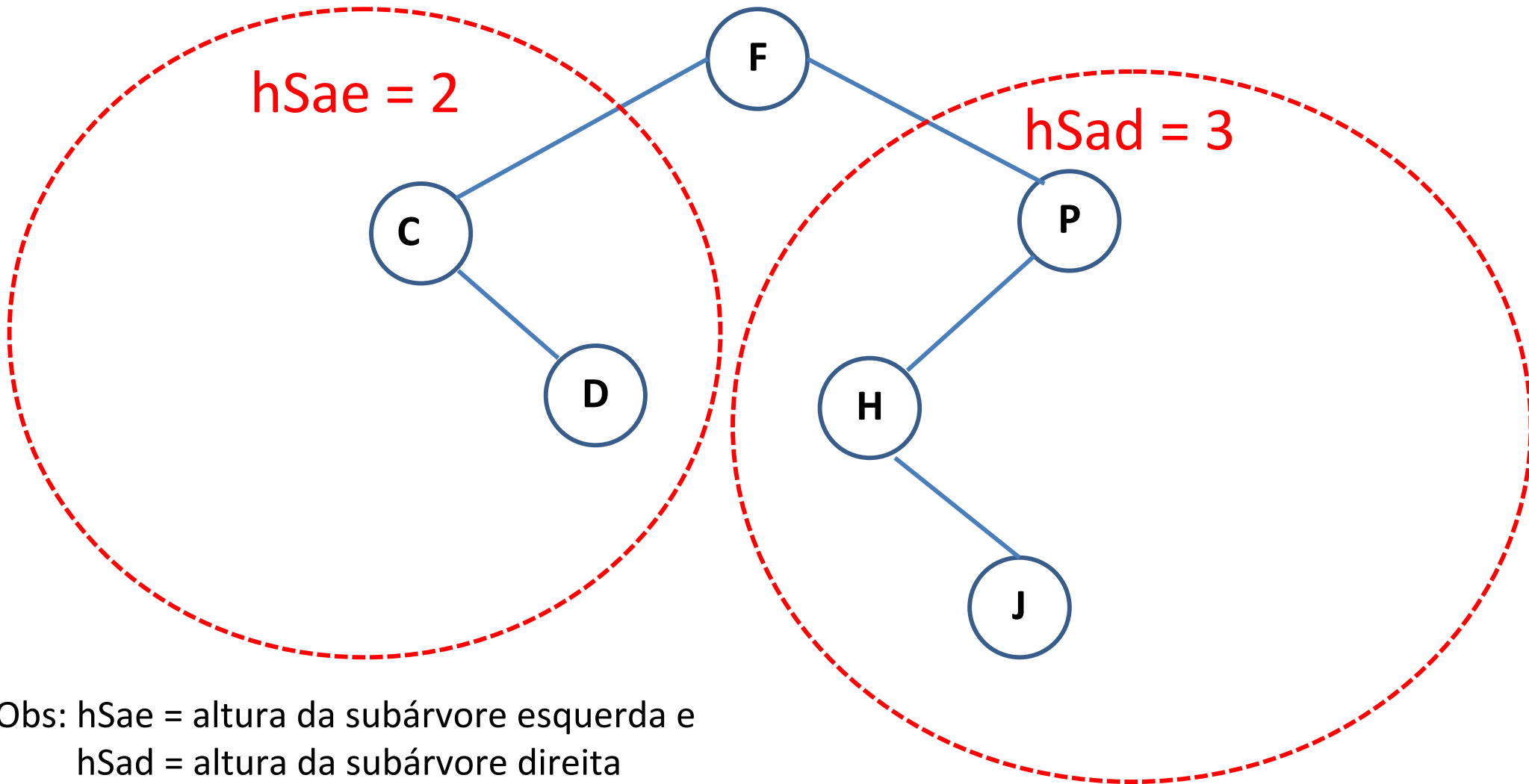
- Árvore Binária de Pesquisa (mas não AVL)



Obs: hSae = altura da subárvore esquerda e  
hSad = altura da subárvore direita

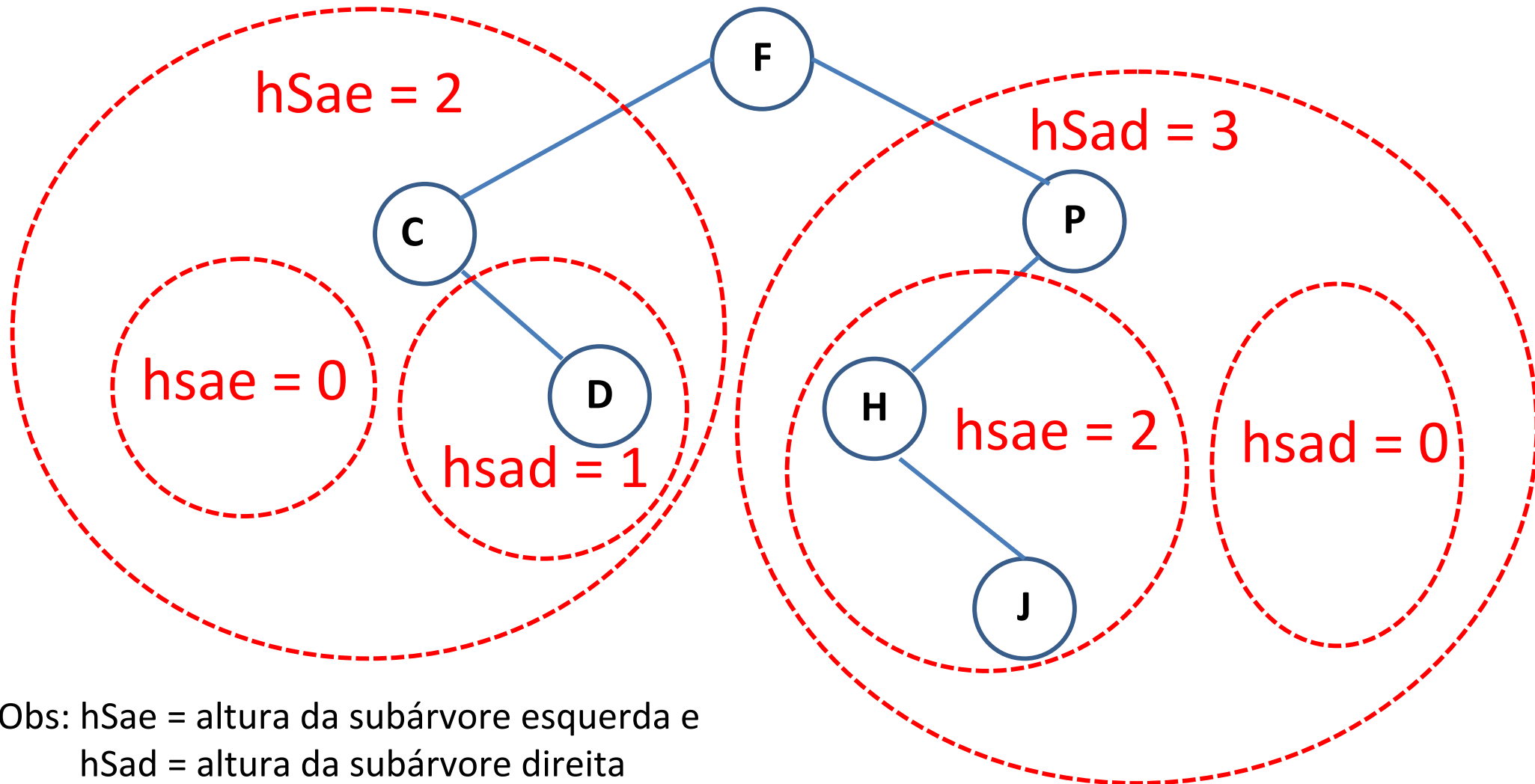
# Exemplo 1

- Árvore Binária de Pesquisa (mas não AVL)



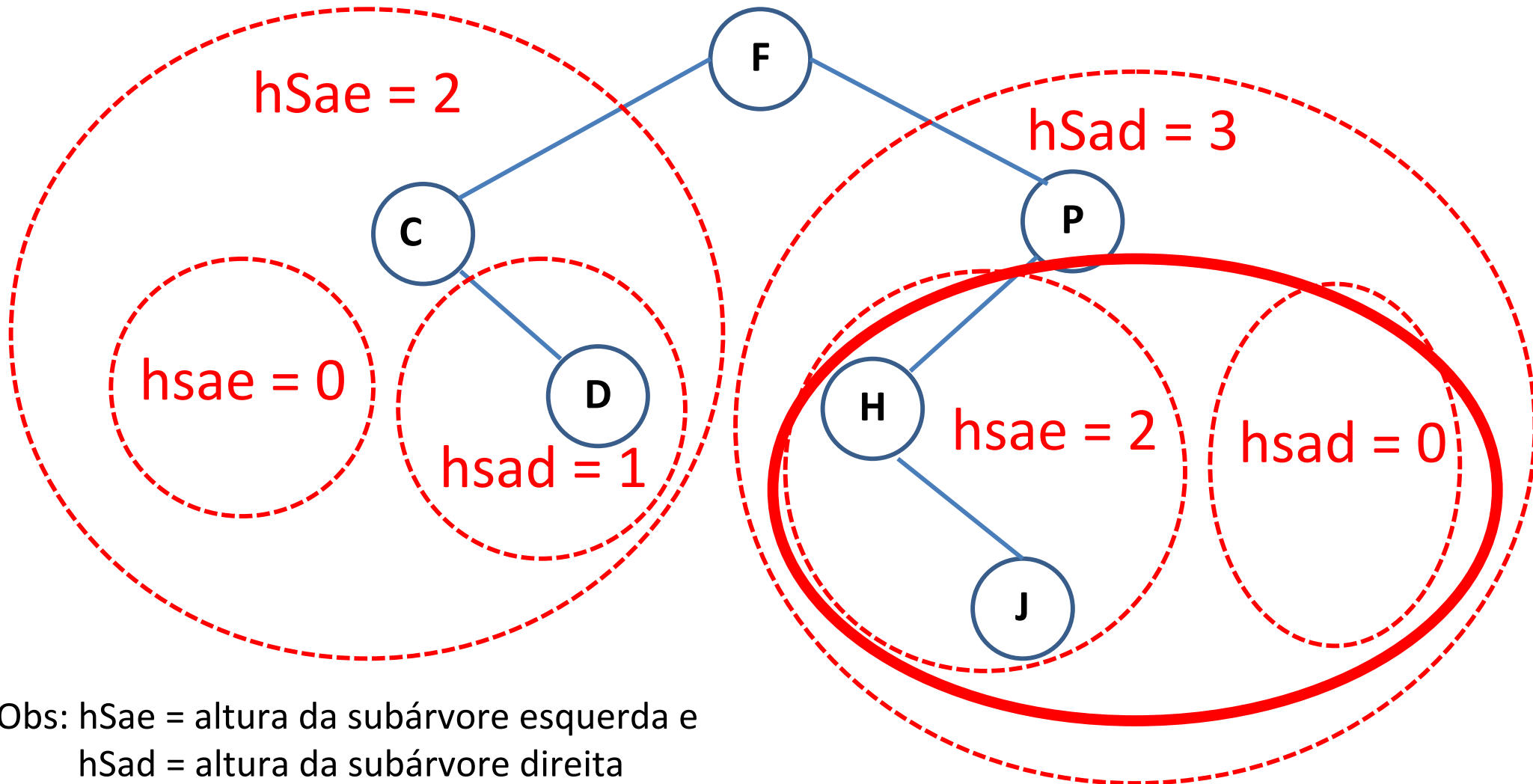
# Exemplo 1

- Árvore Binária de Pesquisa (mas não AVL)



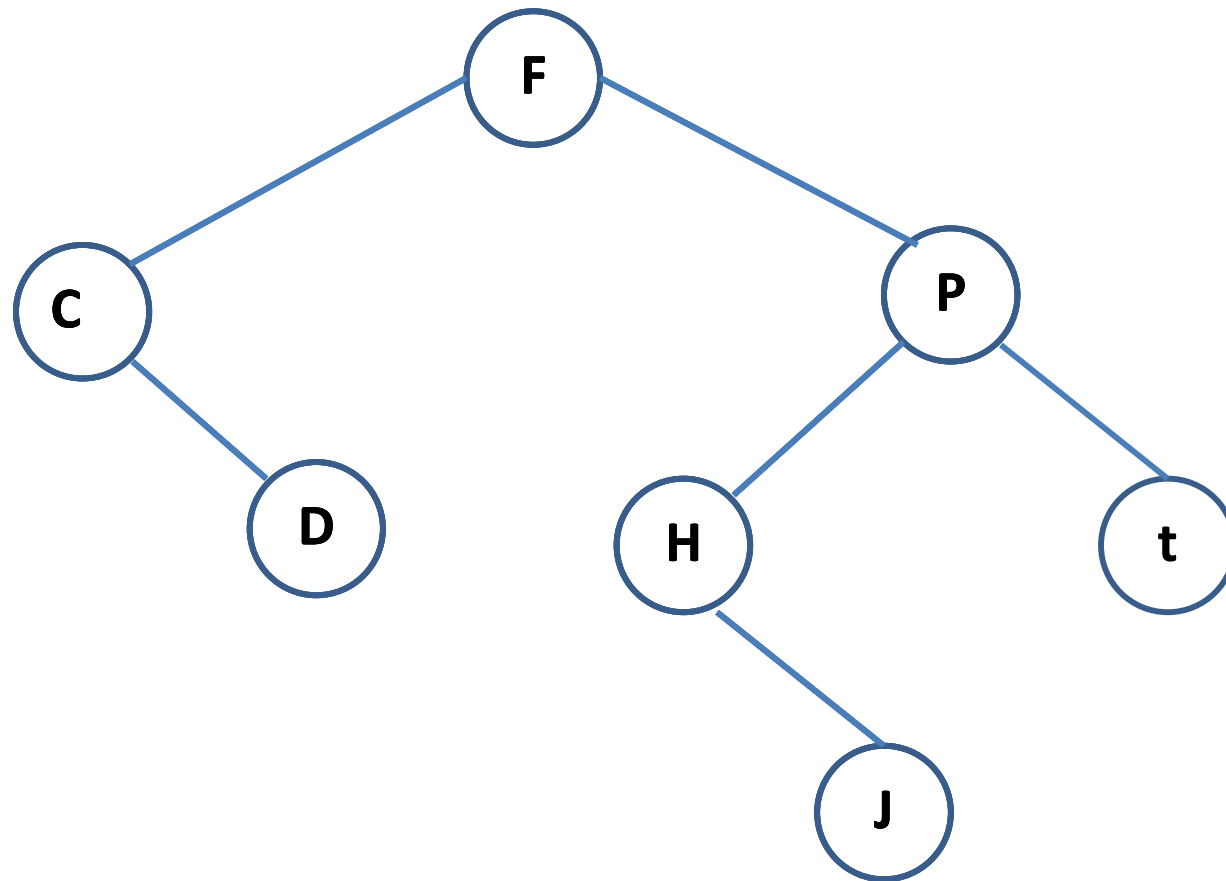
# Exemplo 1

- Árvore Binária de Pesquisa (mas não AVL)



# Exemplo 2

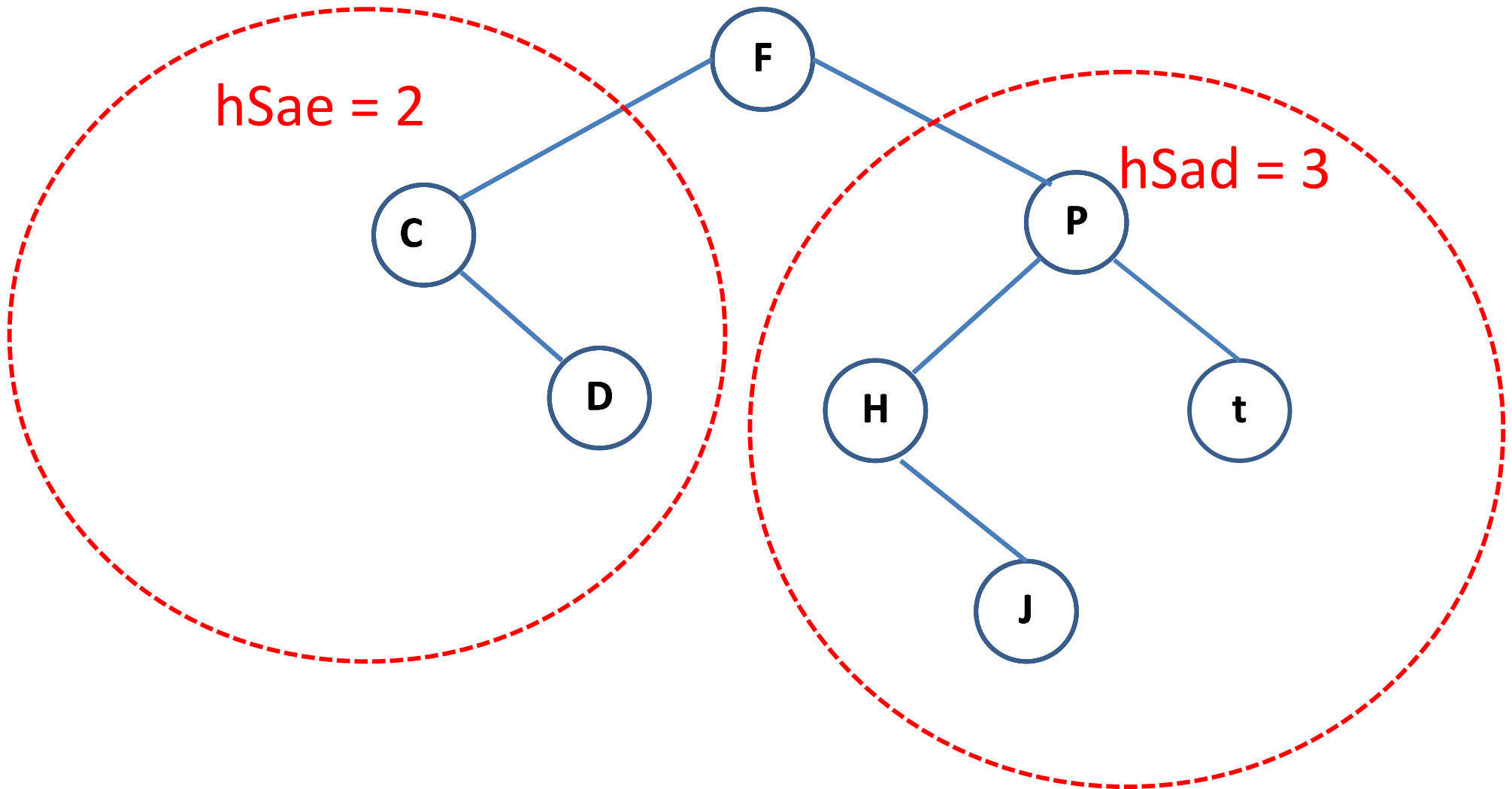
- Árvore Binária de Pesquisa - AVL





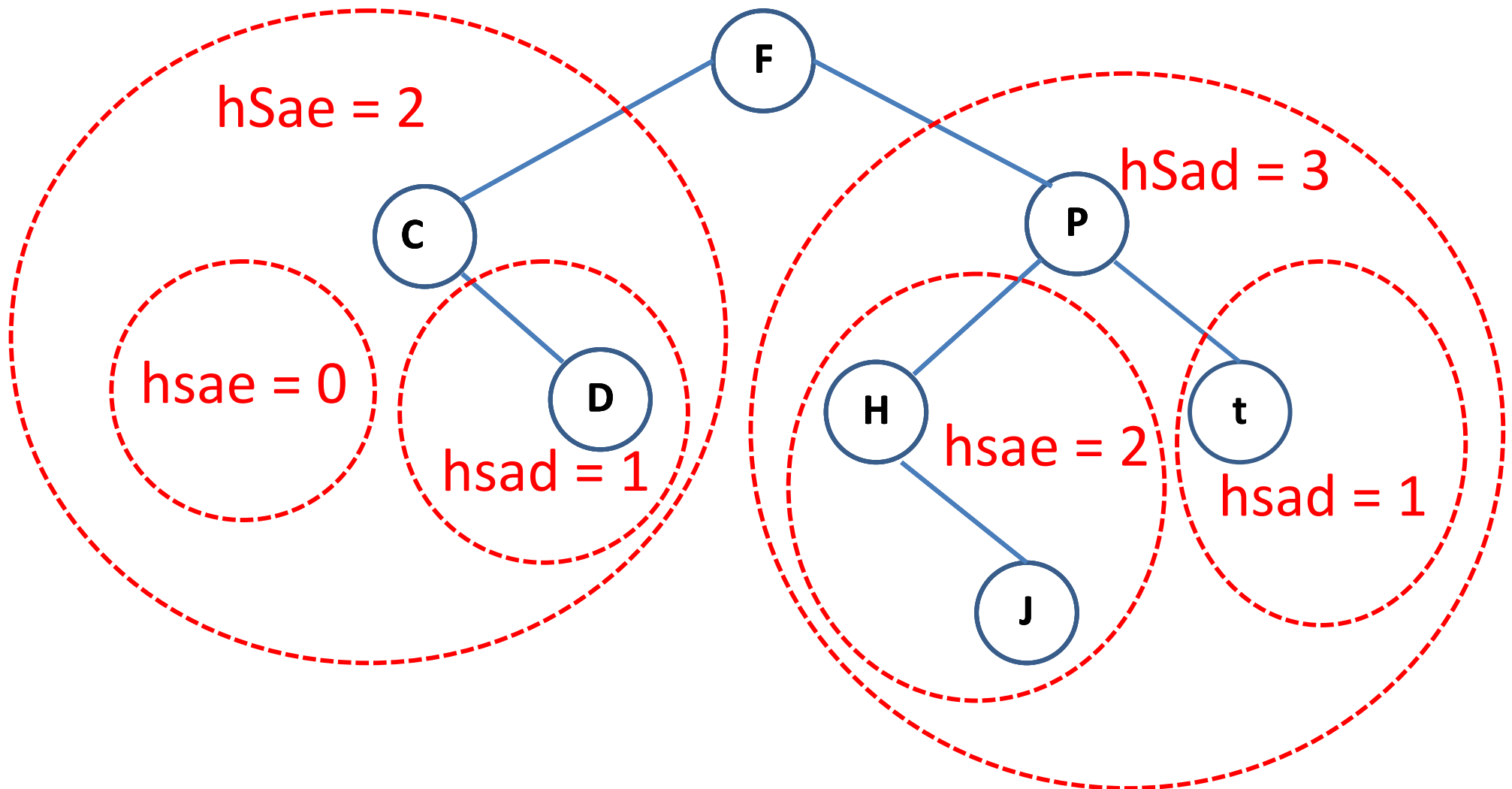
# Exemplo 2

- Árvore Binária de Pesquisa - AVL



# Exemplo 2

- Árvore Binária de Pesquisa - AVL

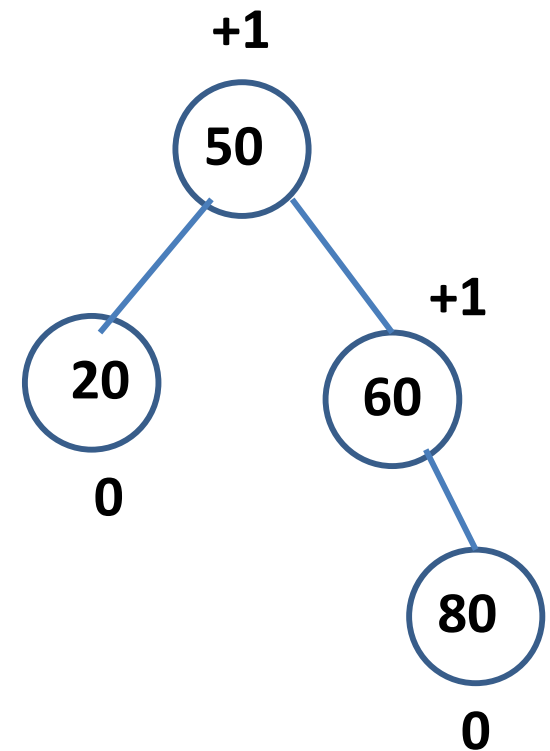
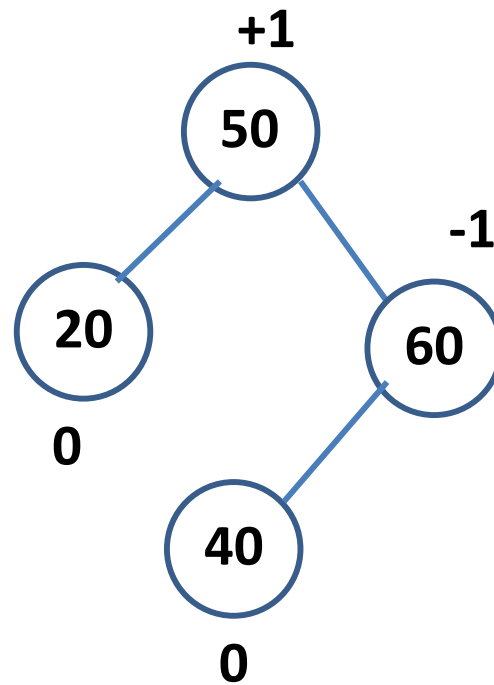
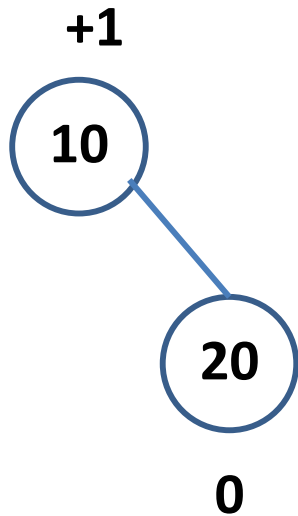


# Fator de Balanceamento

- Cada nó em uma árvore AVL deve ter um Fator de Balanceamento (FB) que dever ser: **-1**, **0** ou **+1**.
- Valores diferentes destes indicam árvore desbalanceada.
  - indica que a altura da subárvore esquerda é maior que a altura da subárvore direita.
  - + indica que a altura da subárvore direita é maior que a altura da subárvore esquerda.
  - 0** indica que as alturas das subárvores esquerda e direita são iguais (árvore equilibrada).
- O fator de balanceamento é calculado como:

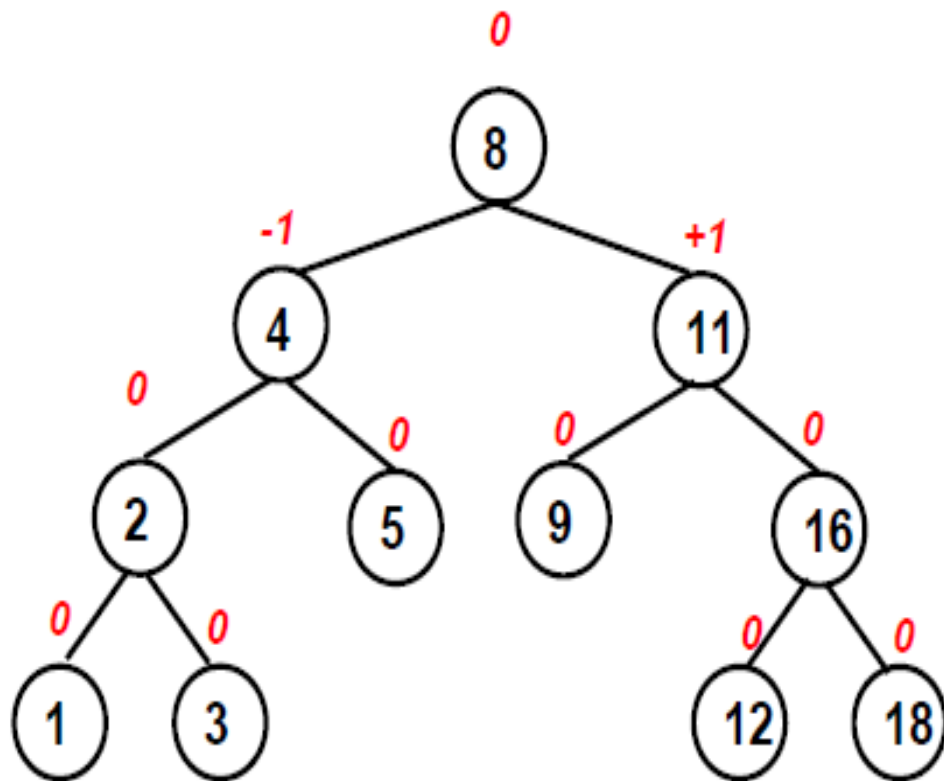
$$\text{Factor de Balanceamento(FB)} = \text{altura(subárvore\_direita(P))} - \text{altura(subárvore\_esquerda(P))}$$

# Exemplos FB

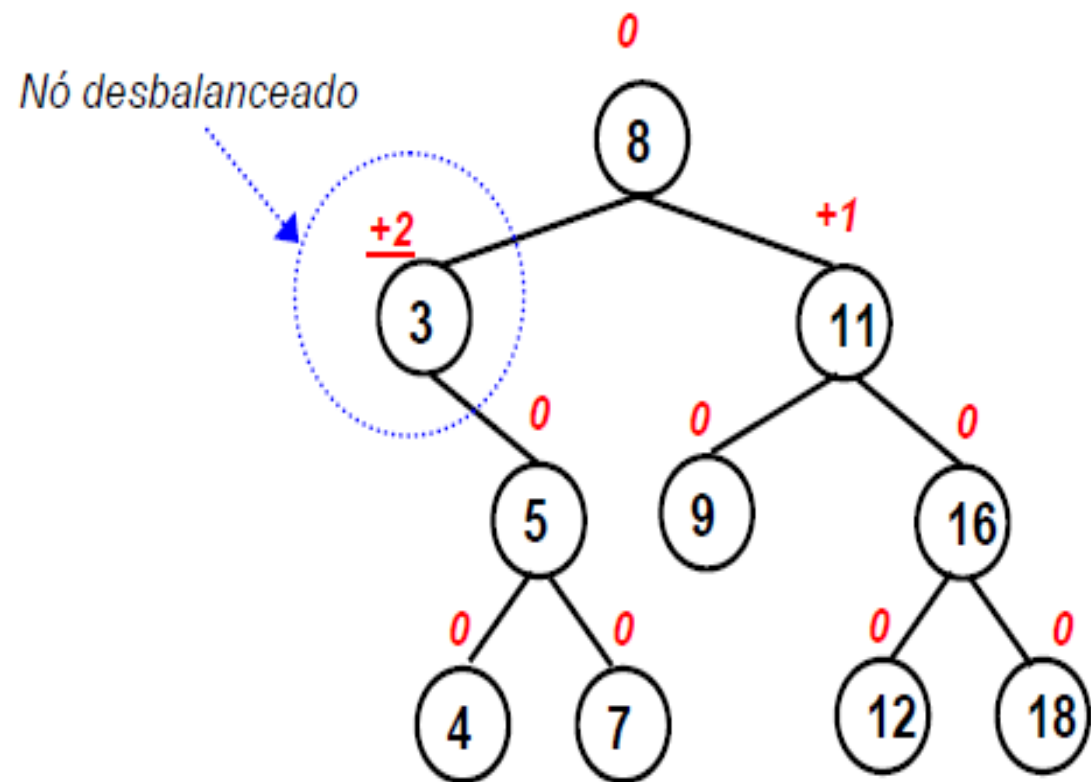


# Exemplos de Árvores Binárias

Árvore AVL



Árvore Não - AVL



# Estrutura de Dados

esq	info	pai	fb	dir
-----	------	-----	----	-----

```
typedef struct no_AVL AVL;  
struct no_AVL {  
    int info;  
    int fb;    // fator de balanceamento  
    AVL *pai;  
    AVL *esq;  
    AVL *dir;  
};
```

# Inserção de Nós

- **Etapas 1:** Inserir um elemento da mesma forma que em uma árvore binária de pesquisa.
- **Etapas 2:** Verificar se após a inserção do nó a árvore resultante continua sendo uma árvore AVL.

# Exemplo de Inserção

- Inserir: G, E, J, C, F, K, B, D, A



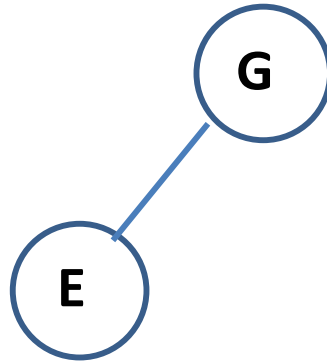
# Exemplo de Inserção

- Inserir: G, E, J, C, F, K, B, D, A



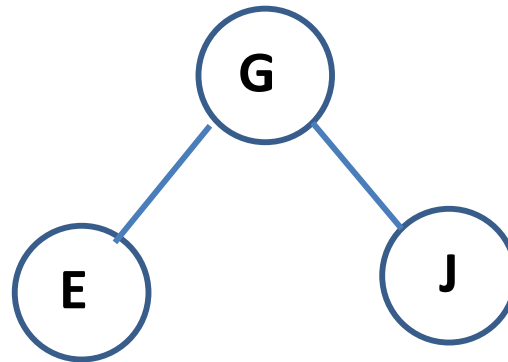
# Exemplo de Inserção

- Inserir: G, E, J, C, F, K, B, D, A



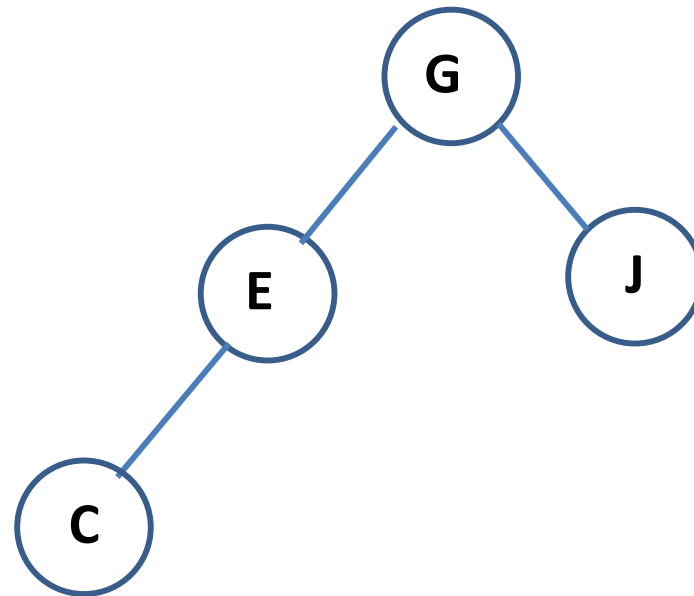
# Exemplo de Inserção

- Inserir: G, E, J, C, F, K, B, D, A



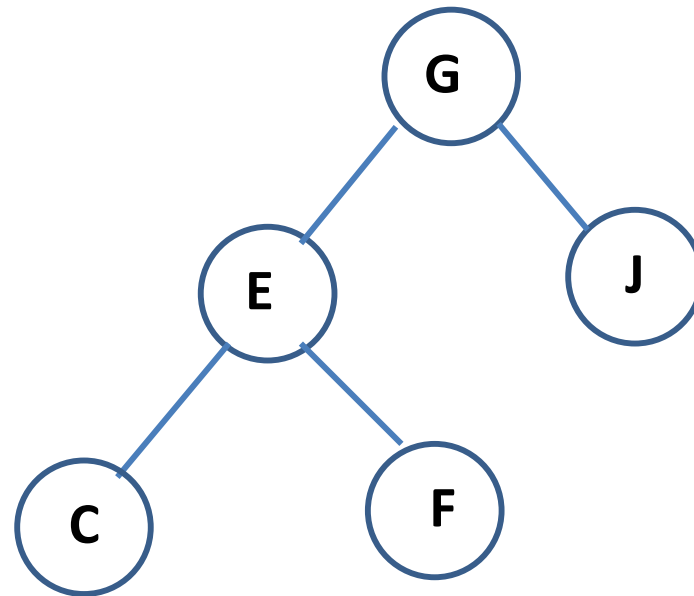
# Exemplo de Inserção

- Inserir: G, E, J, C, F, K, B, D, A



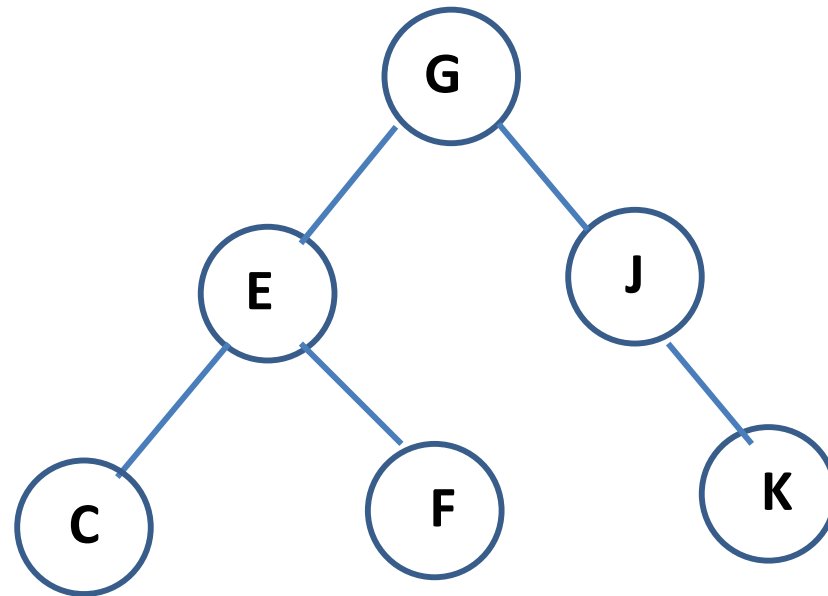
# Exemplo de Inserção

- Inserir: G, E, J, C, F, K, B, D, A



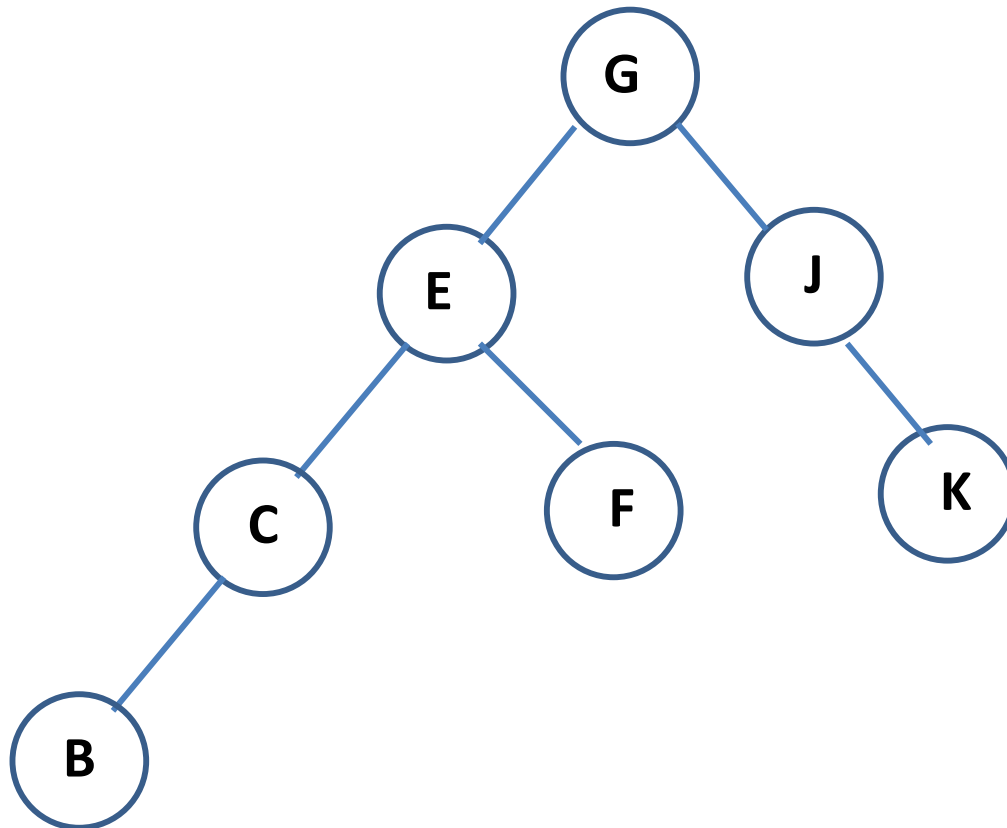
# Exemplo de Inserção

- Inserir: G, E, J, C, F, K, B, D, A



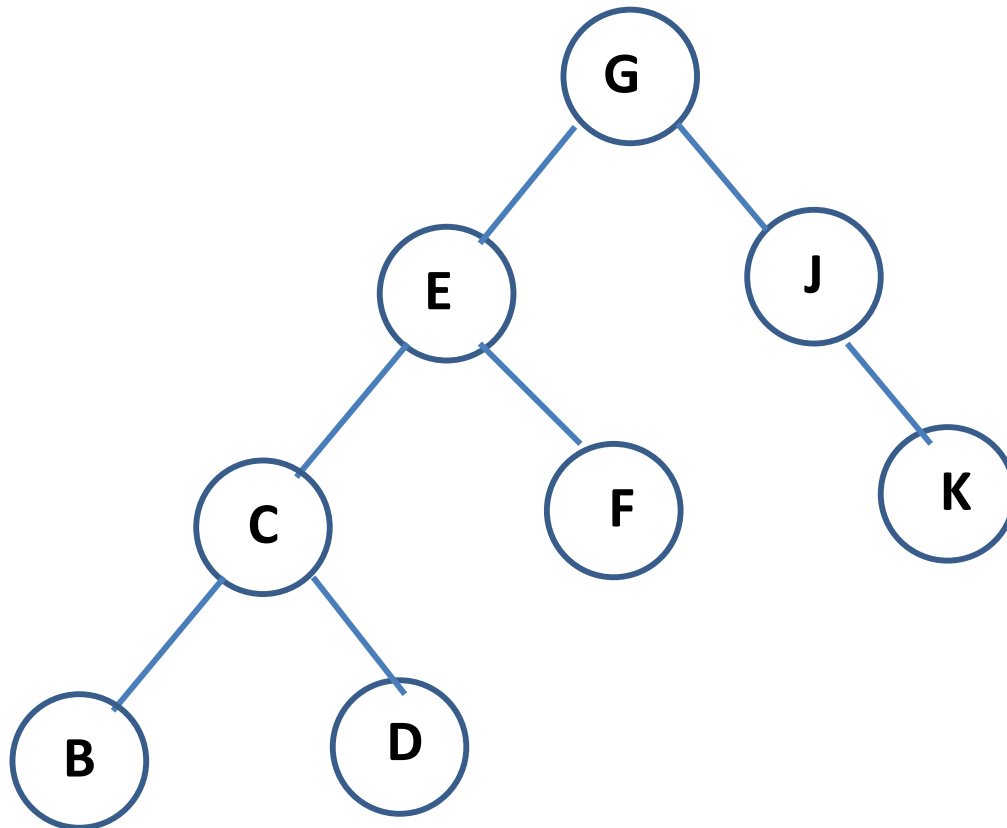
# Exemplo de Inserção

- Inserir: G, E, J, C, F, K, B, D, A



# Exemplo de Inserção

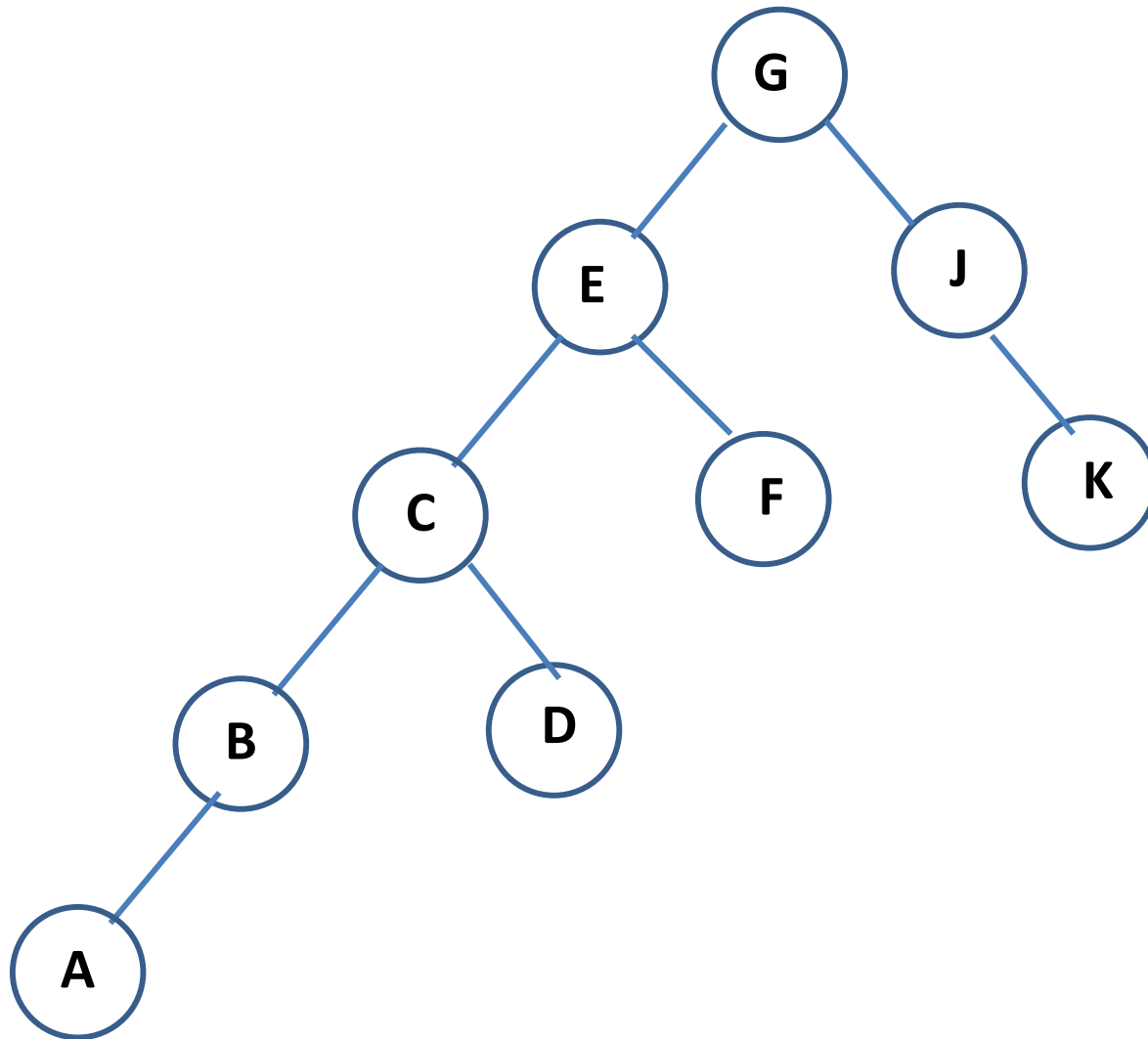
- Inserir: G, E, J, C, F, K, B, D, A





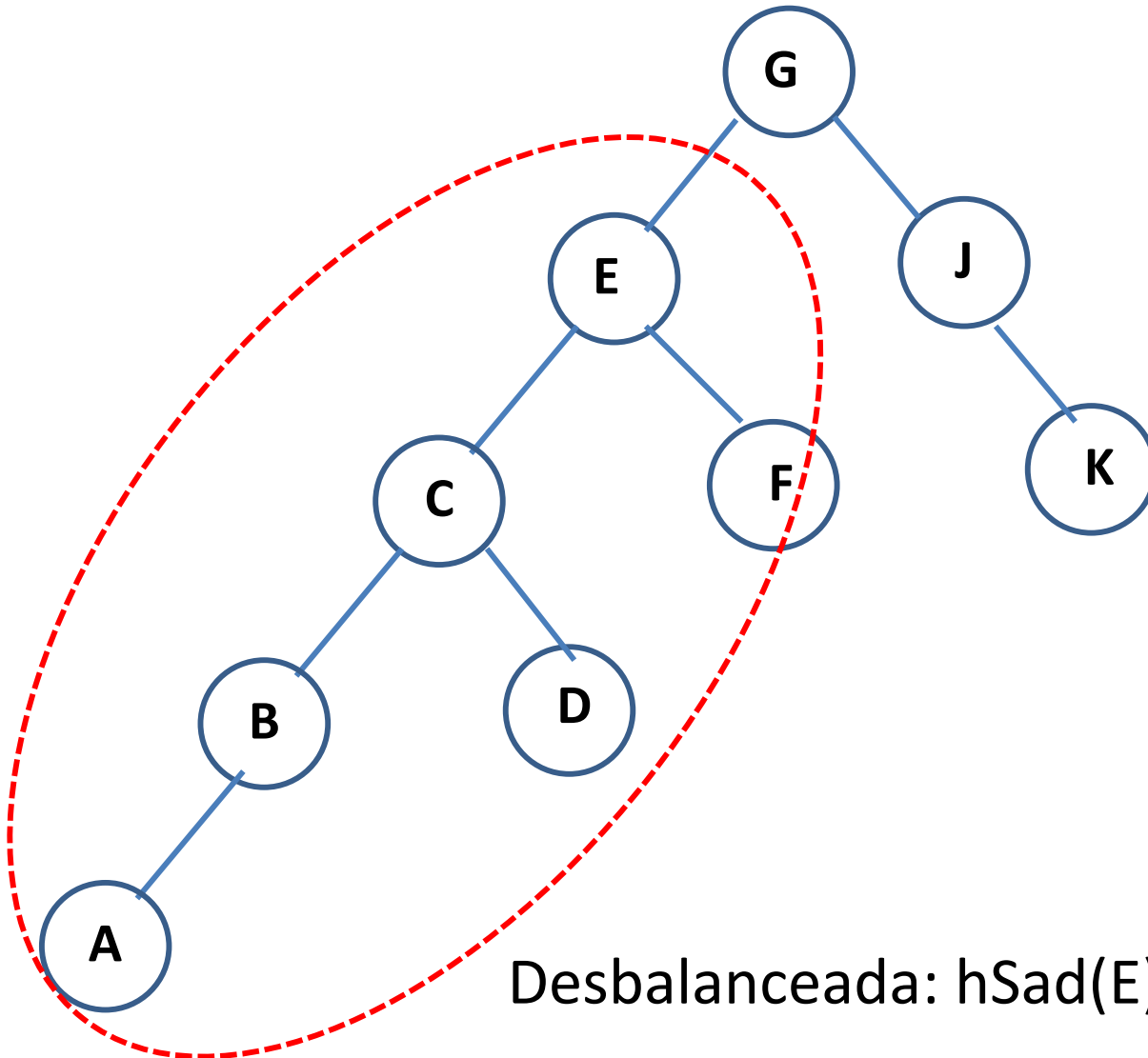
# Exemplo de Inserção

- Inserir: G, E, J, C, F, K, B, D, A



# Exemplo de Inserção

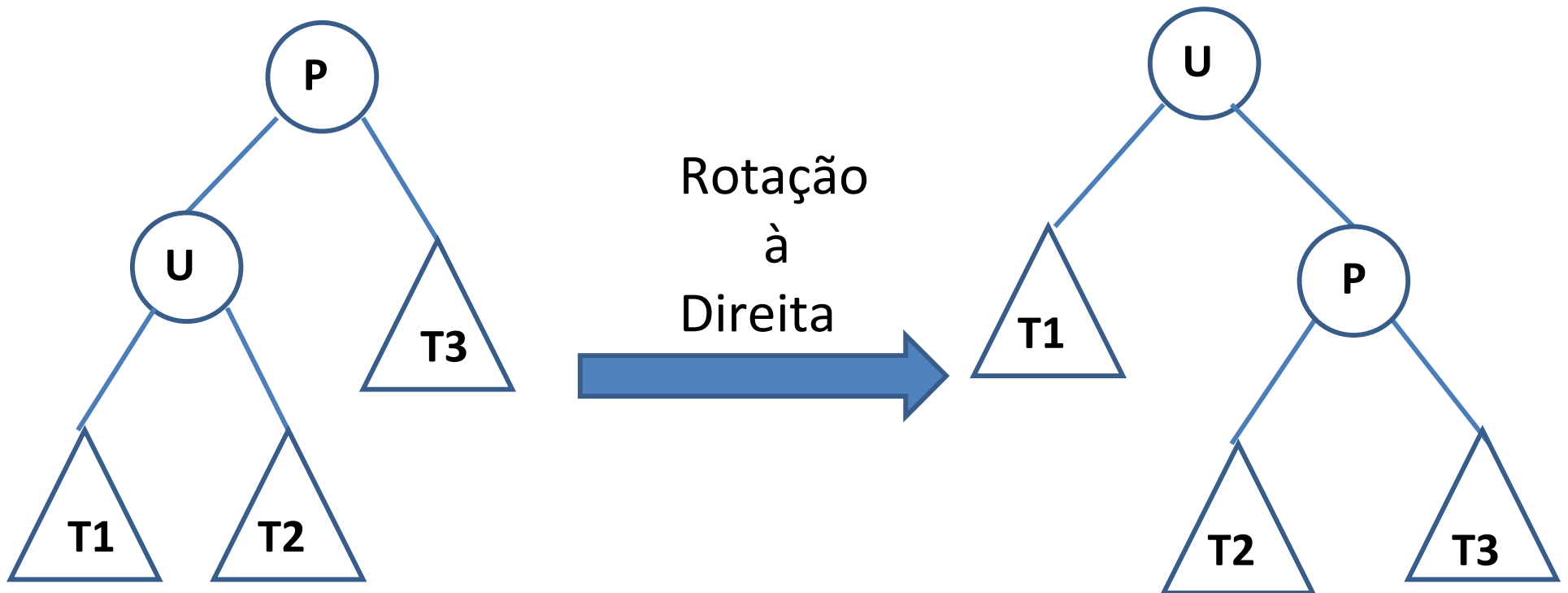
- Inserir: G, E, J, C, F, K, B, D, A



Desbalanceada:  $h_{Sad}(E) - h_{Sae}(E) = 1 - 3 = -2$

# Rotações

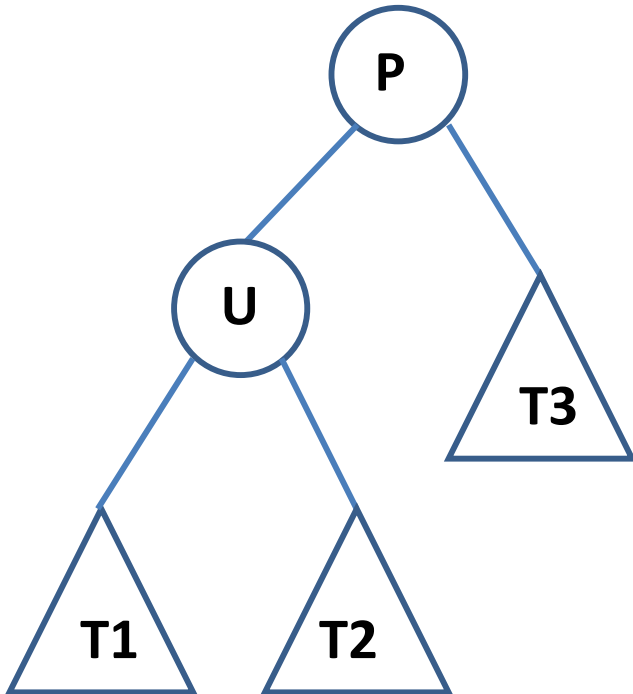
- 1. Rotação (Simples) à Direita



- Seja U o filho à esquerda de P
- Torne o filho à direita de U o filho à esquerda de P.
- Torne P o filho à direita de U

# Rotações

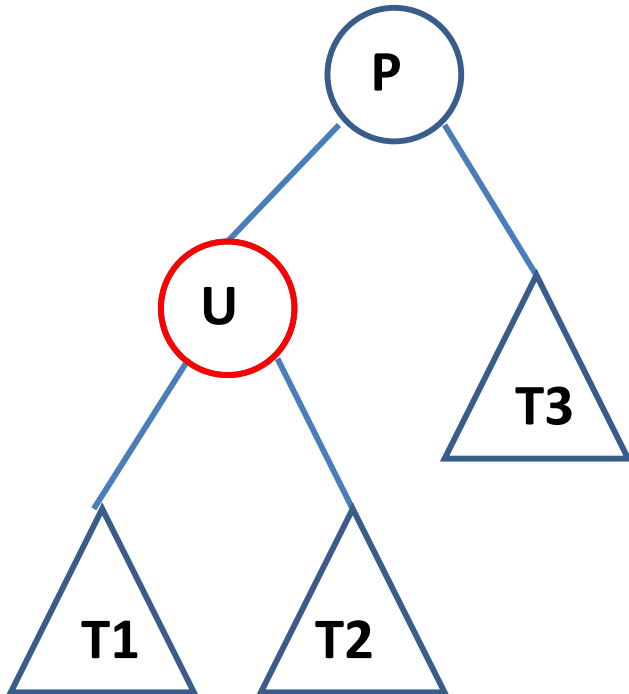
- 1. Rotação (Simple) à Direita



# Rotações

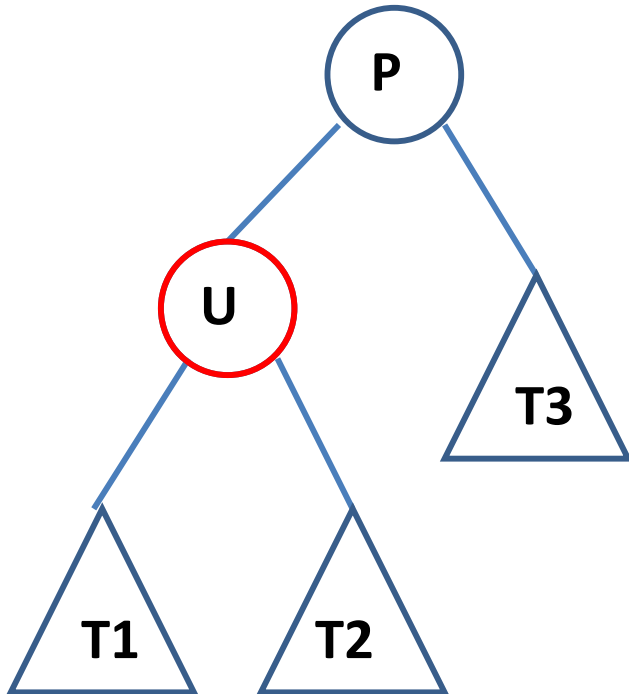
- 1. Rotação (Simples) à Direita

Seja U o filho à  
esquerda de P



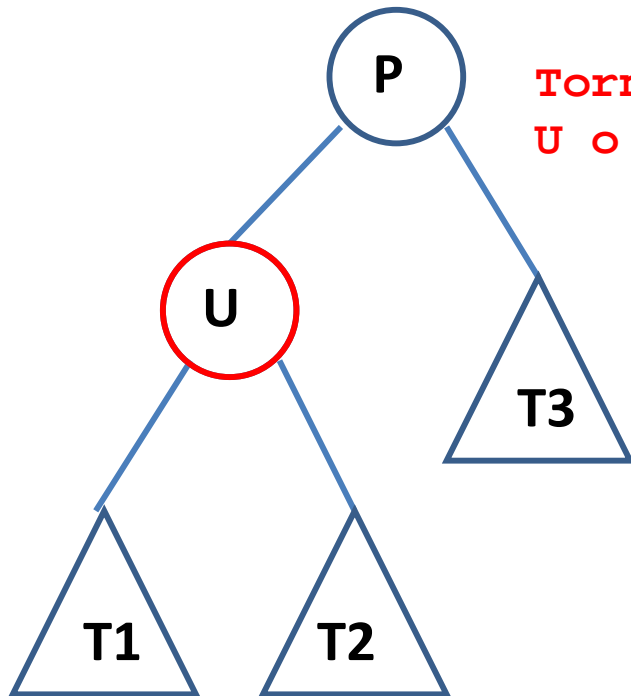
# Rotações

- 1. Rotação (Simples) à Direita



# Rotações

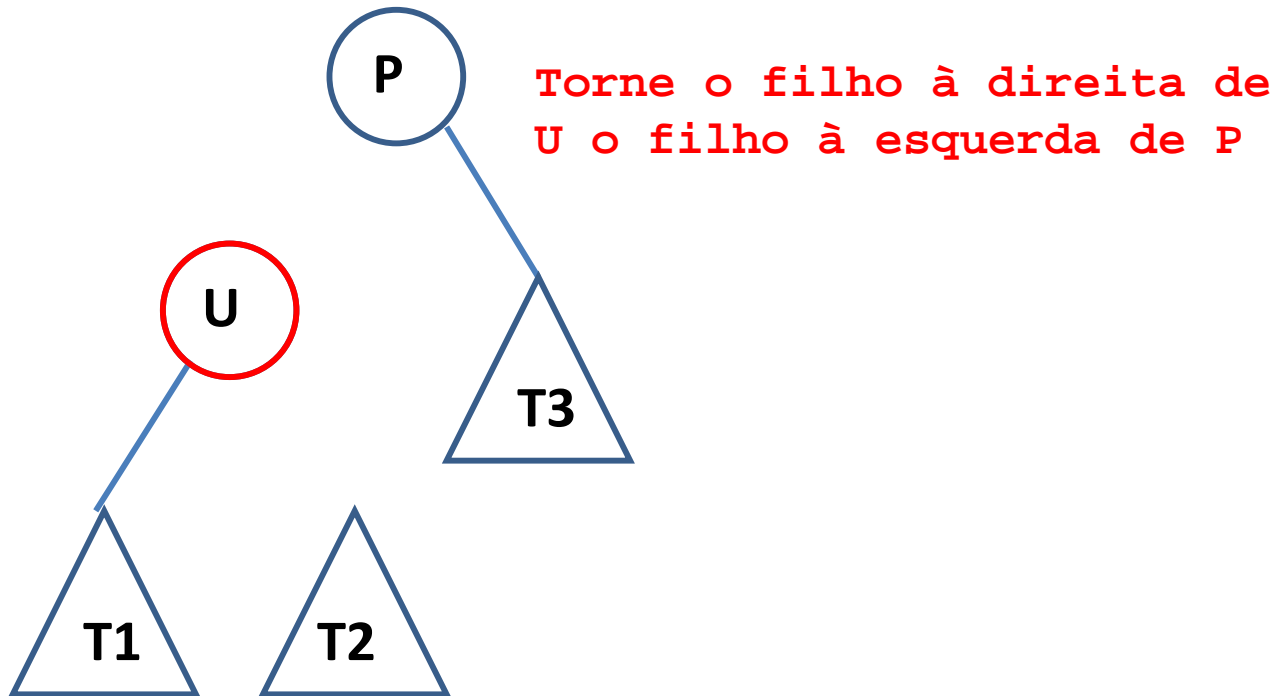
- 1. Rotação (Simples) à Direita



Torne o filho à direita de  
U o filho à esquerda de P

# Rotações

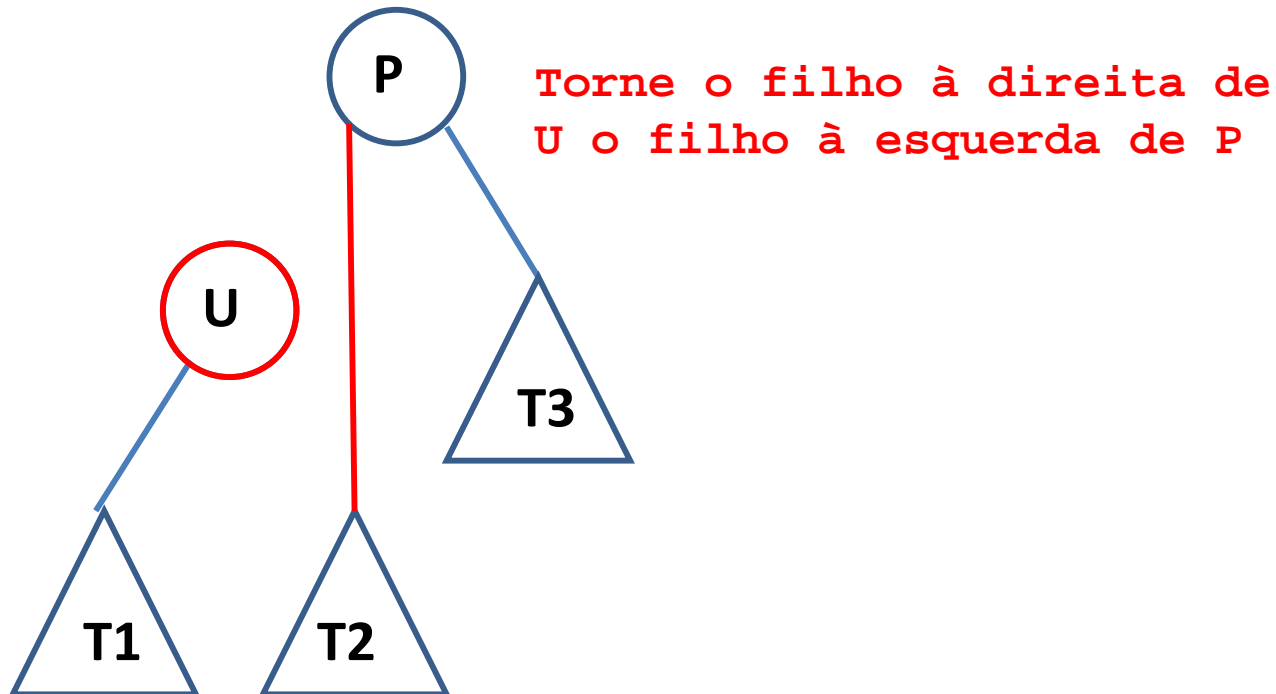
- 1. Rotação (Simples) à Direita





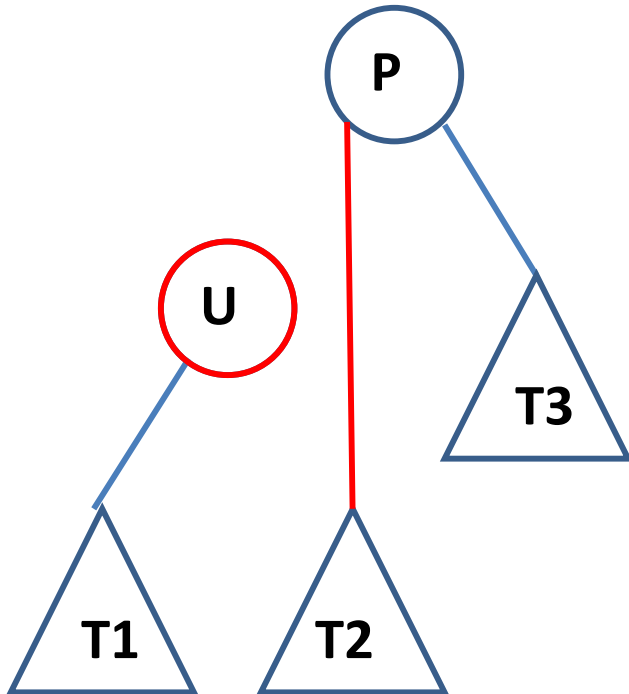
# Rotações

- 1. Rotação (Simples) à Direita



# Rotações

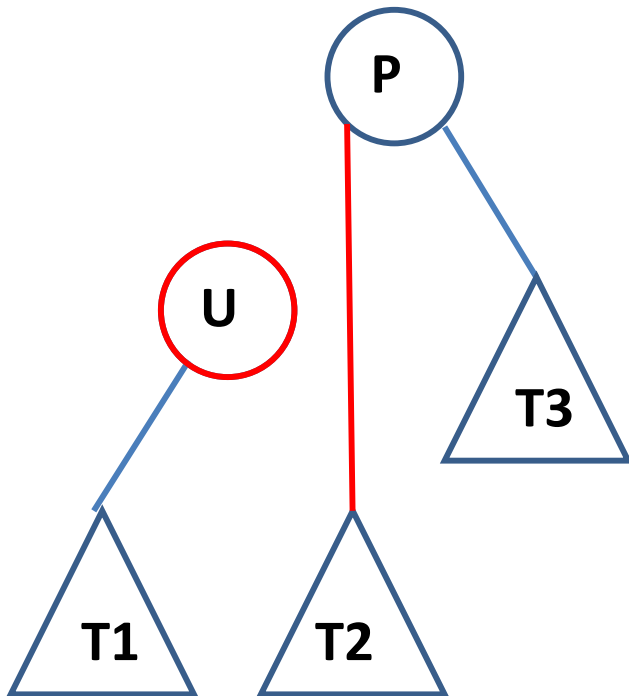
- 1. Rotação (Simples) à Direita



# Rotações

- 1. Rotação (Simples) à Direita

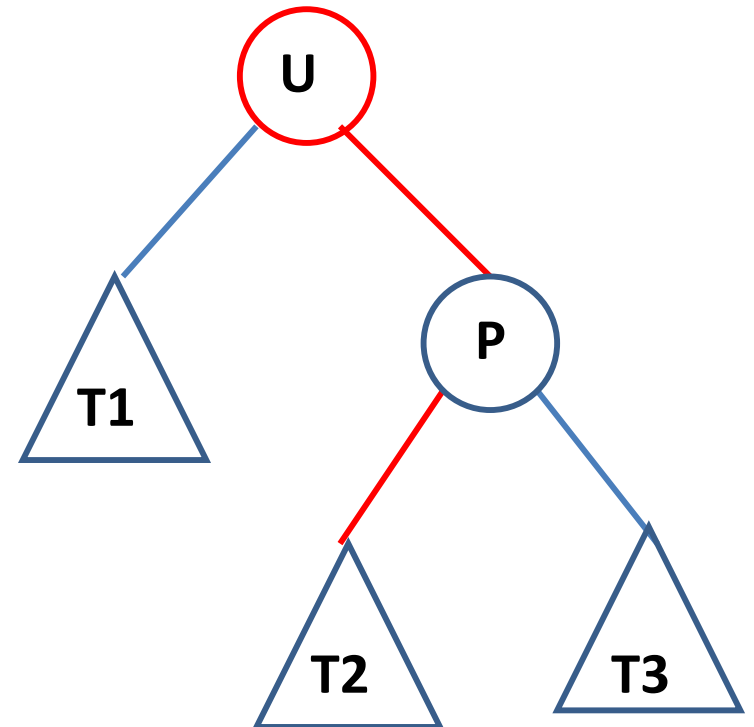
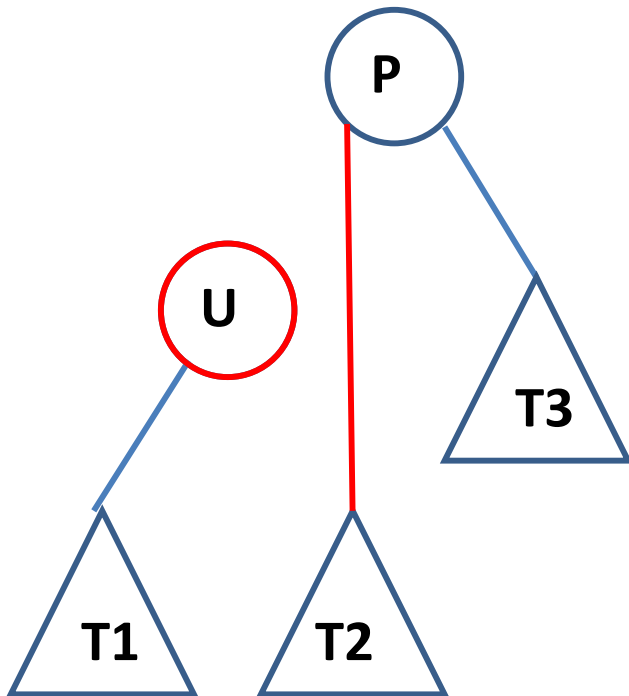
Torne P o filho  
à direita de U



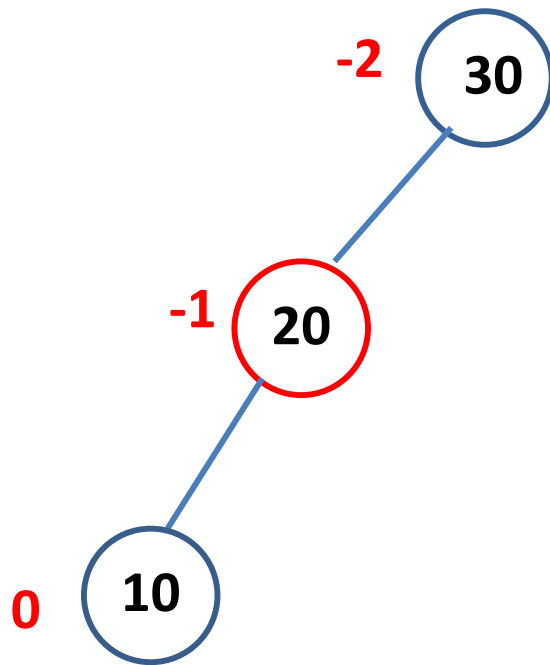
# Rotações

- 1. Rotação (Simples) à Direita

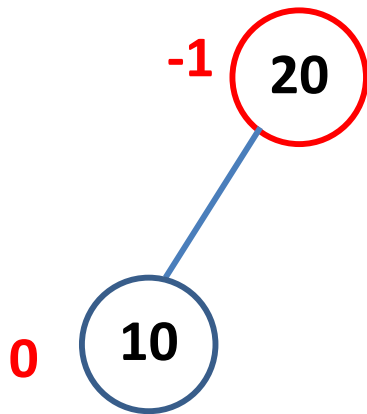
Torne P o filho  
à direita de U



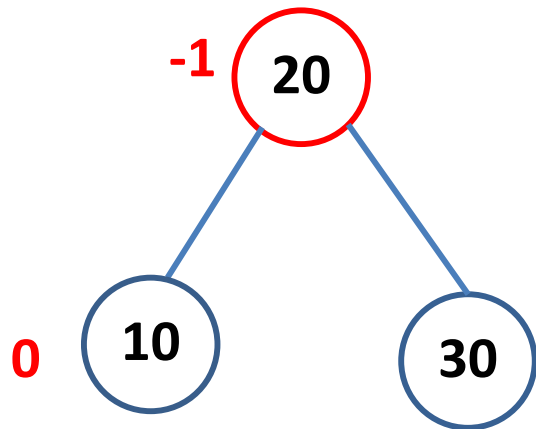
# Exemplo de Rotação à Direita



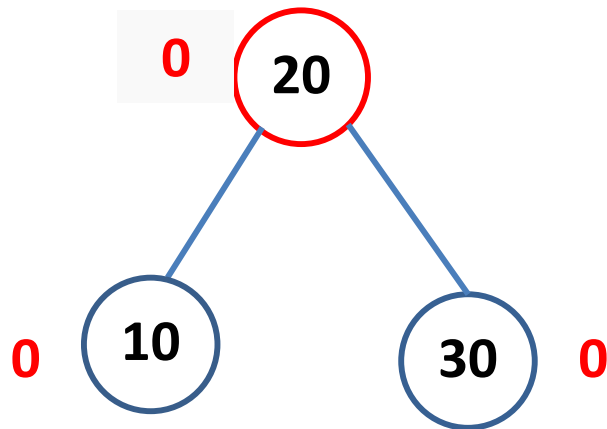
# Exemplo de Rotação à Direita



# Exemplo de Rotação à Direita



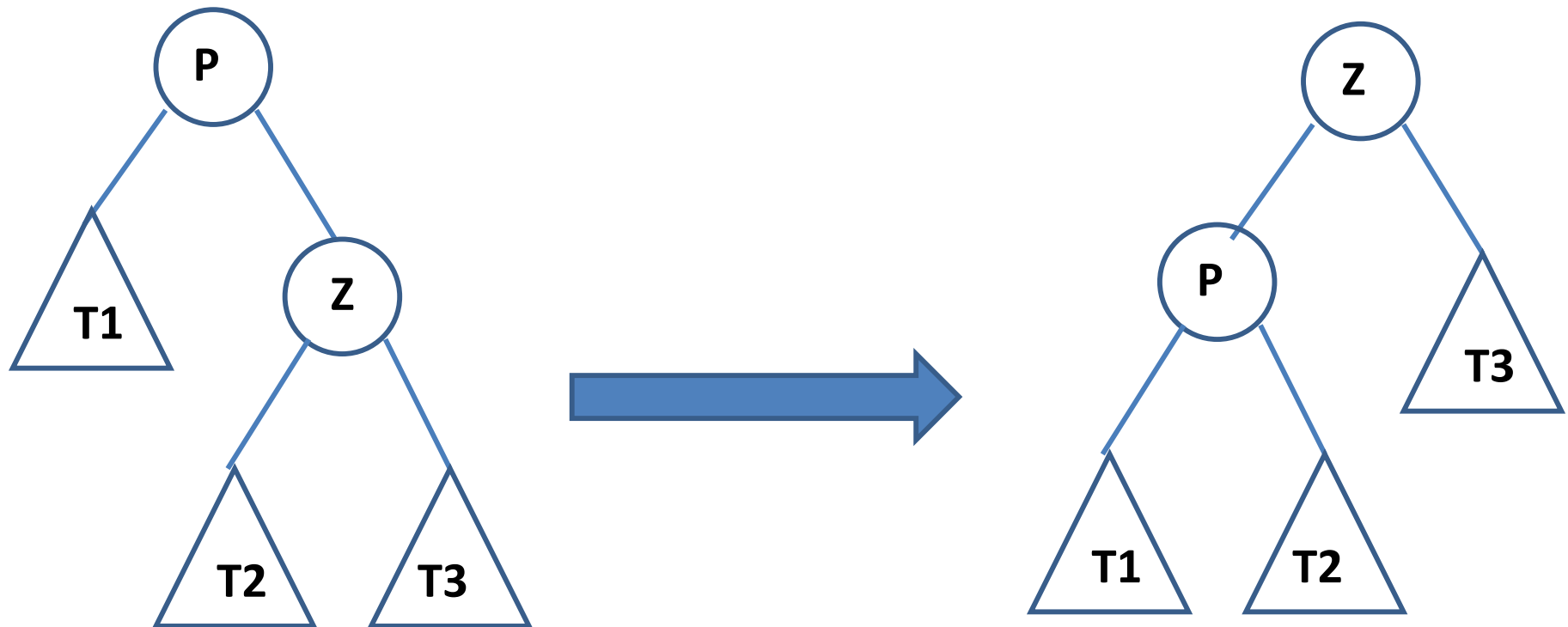
# Exemplo de Rotação à Direita





# Rotações

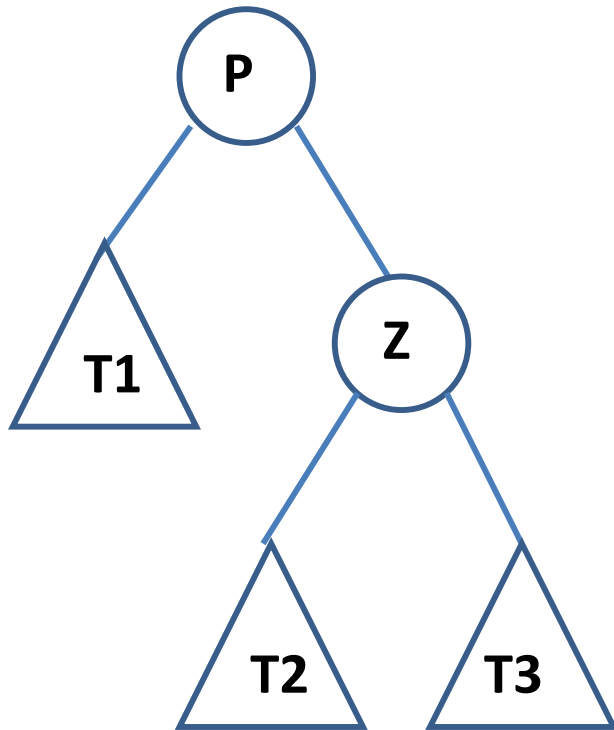
- 2. Rotações (Simples) à Esquerda



- Seja Z o filho à direita de P
- Torne o filho à esquerda de Z o filho à direita de P.
- Torne P filho à esquerda de Z

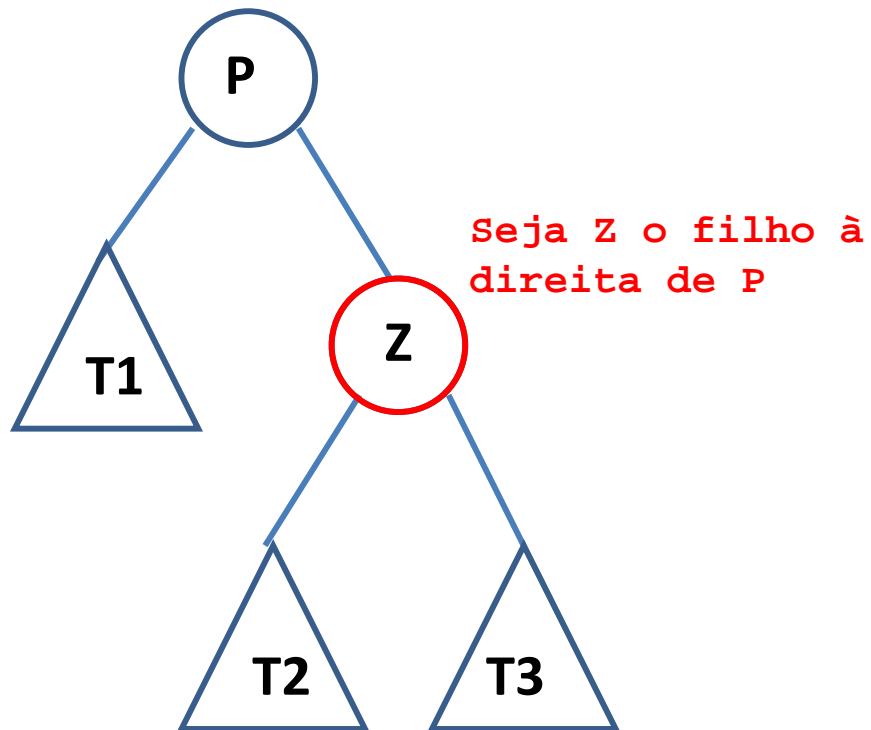
# Rotações

- 2. Rotações (Simples) à Esquerda



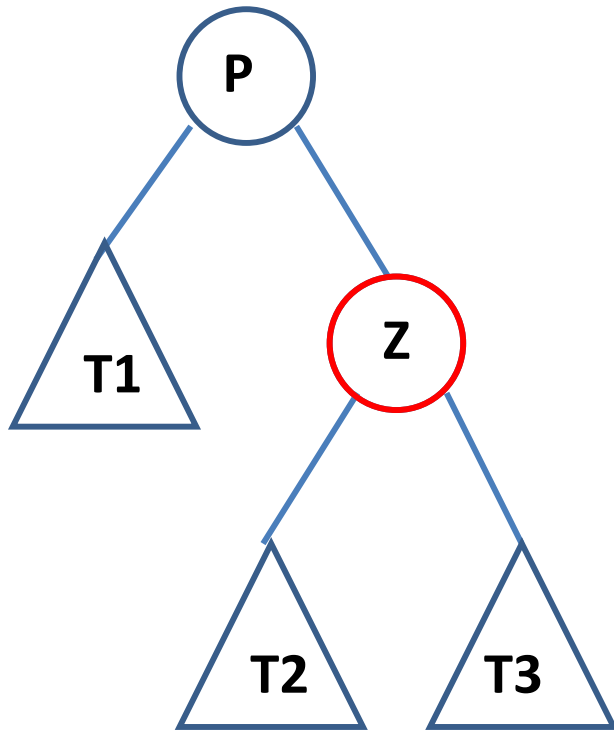
# Rotações

- 2. Rotações (Simples) à Esquerda



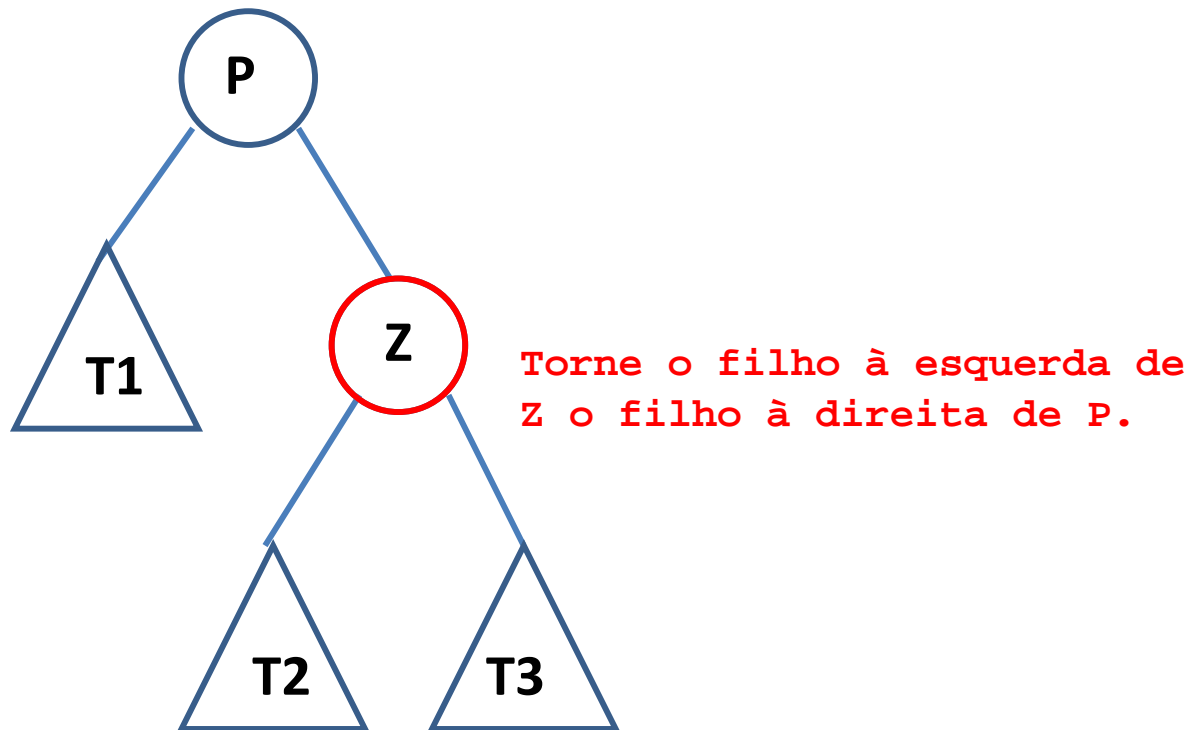
# Rotações

- 2. Rotações (Simples) à Esquerda



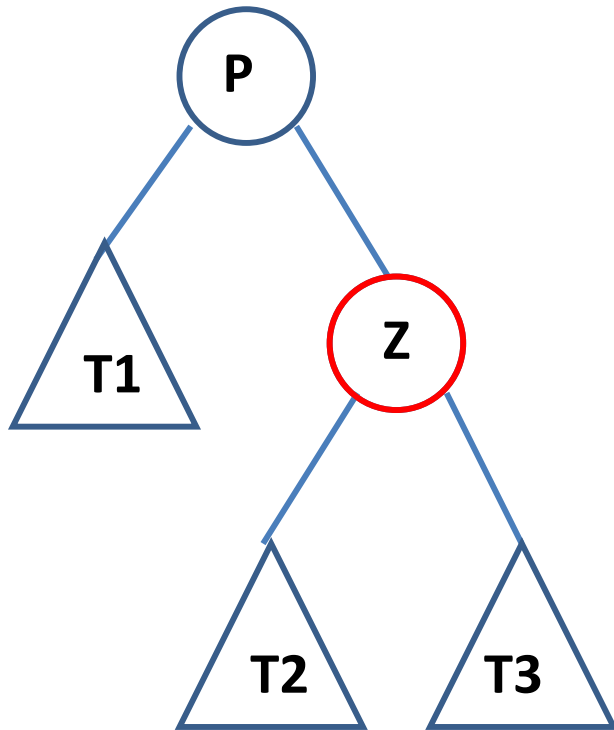
# Rotações

- 2. Rotações (Simples) à Esquerda



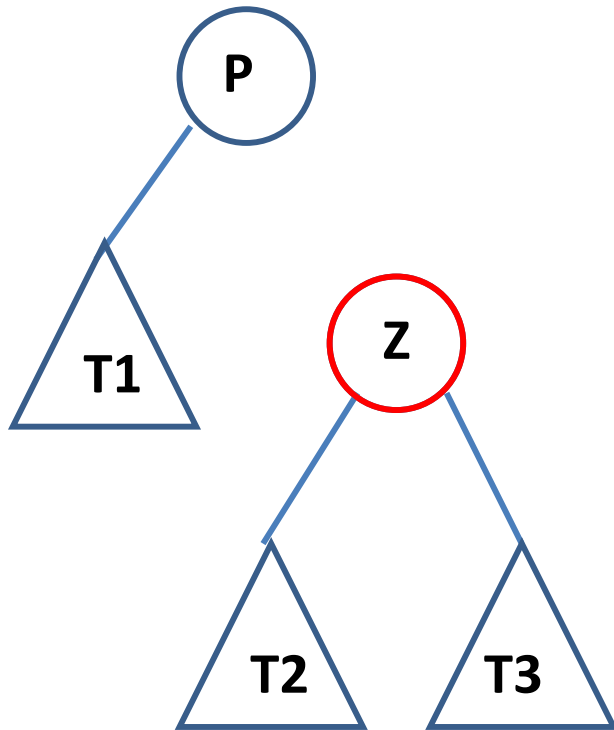
# Rotações

- 2. Rotações (Simples) à Esquerda



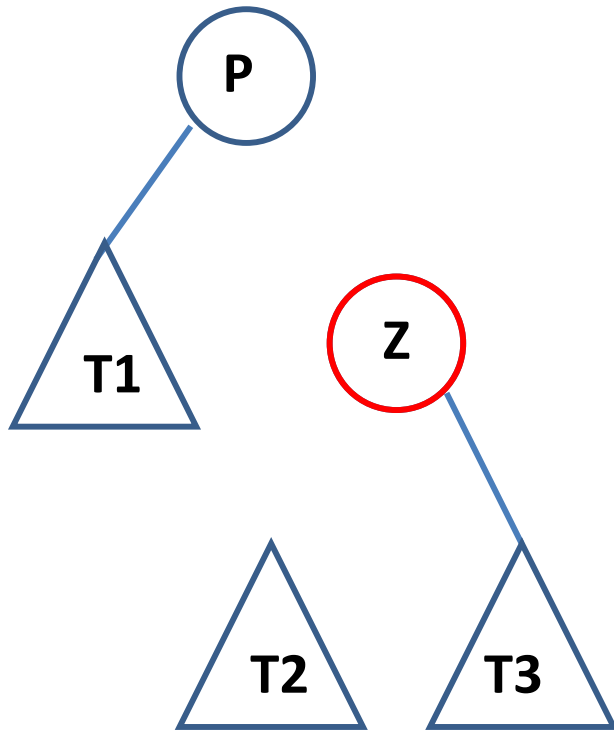
# Rotações

- 2. Rotações (Simples) à Esquerda



# Rotações

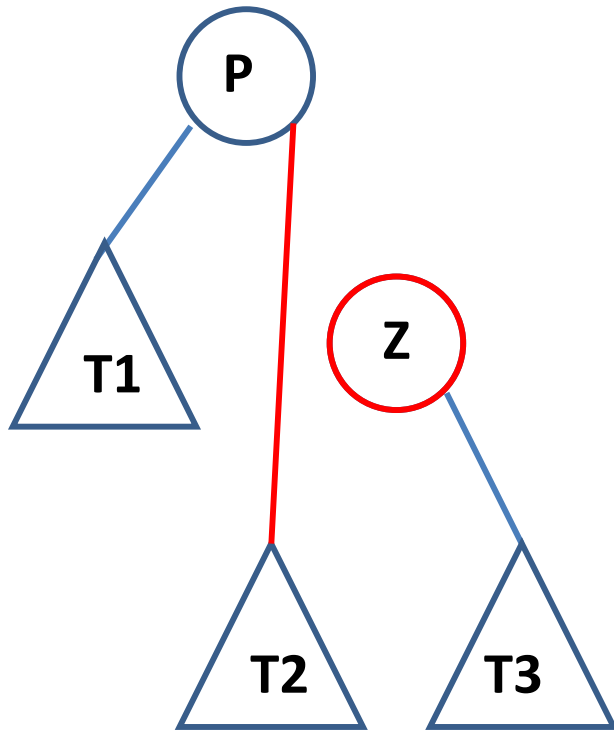
- 2. Rotações (Simples) à Esquerda





# Rotações

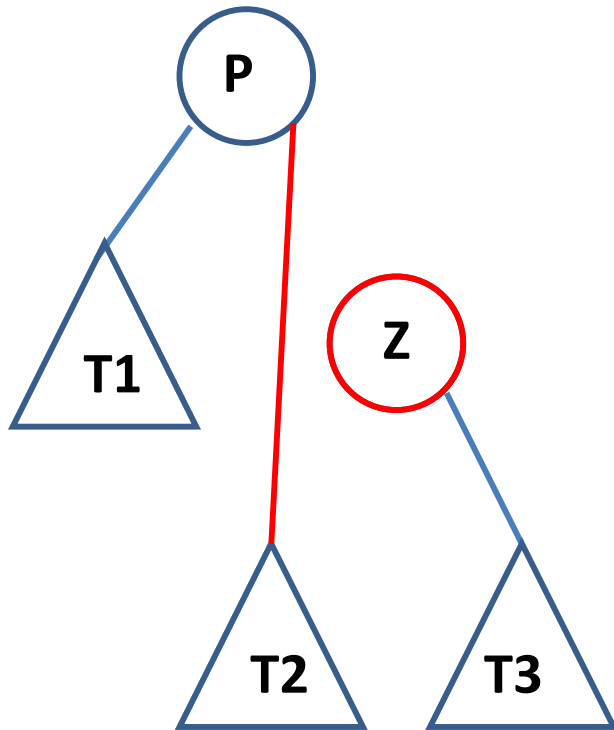
- 2. Rotações (Simples) à Esquerda



# Rotações

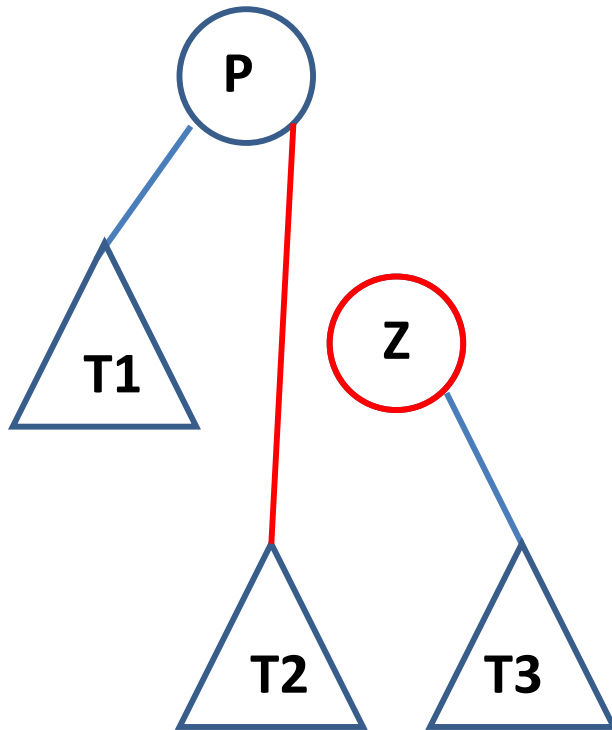
- 2. Rotações (Simples) à Esquerda

Torne P filho à esquerda Z



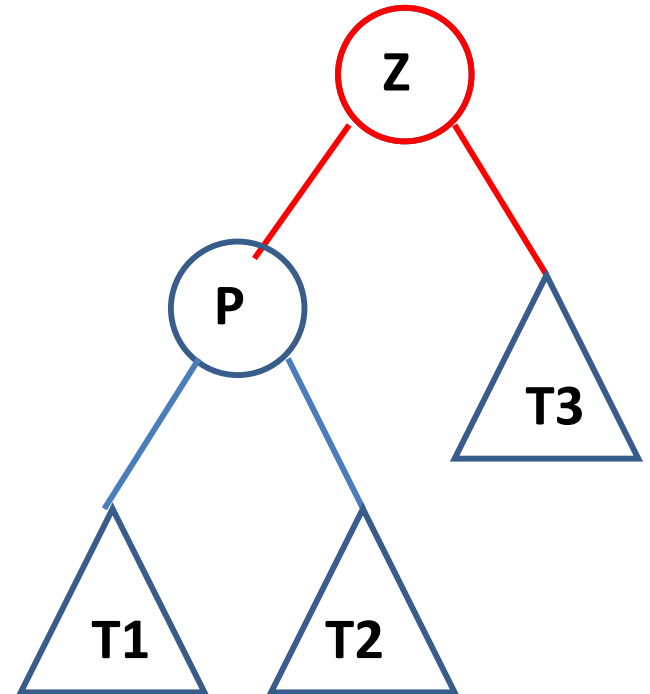
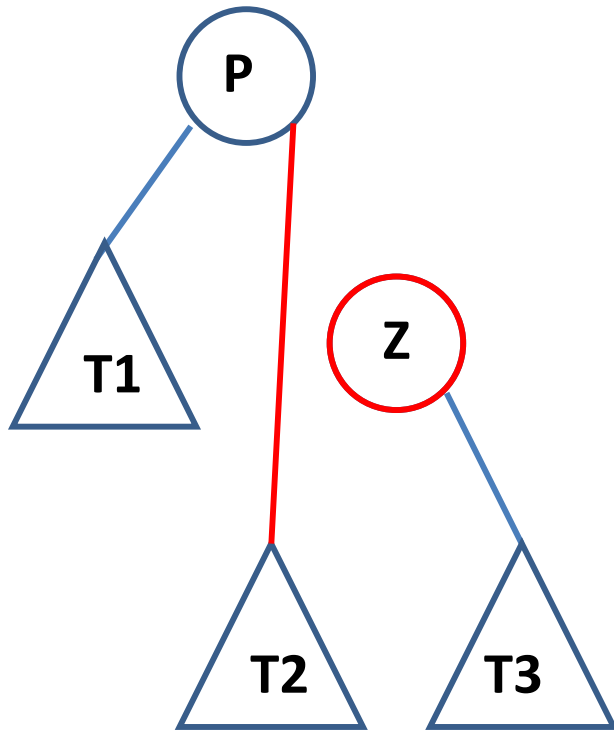
# Rotações

- 2. Rotações (Simples) à Esquerda

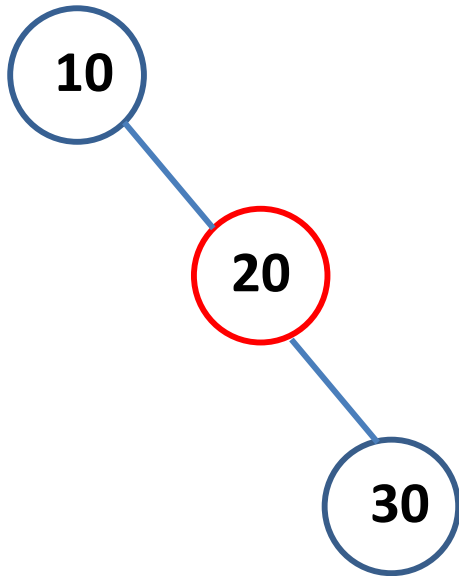


# Rotações

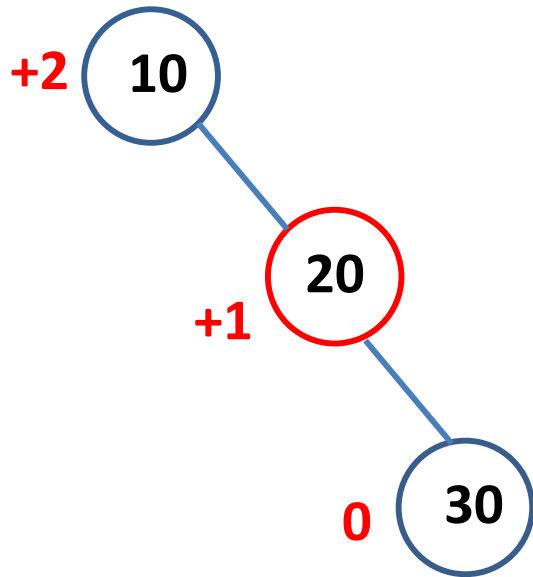
- 2. Rotações (Simples) à Esquerda



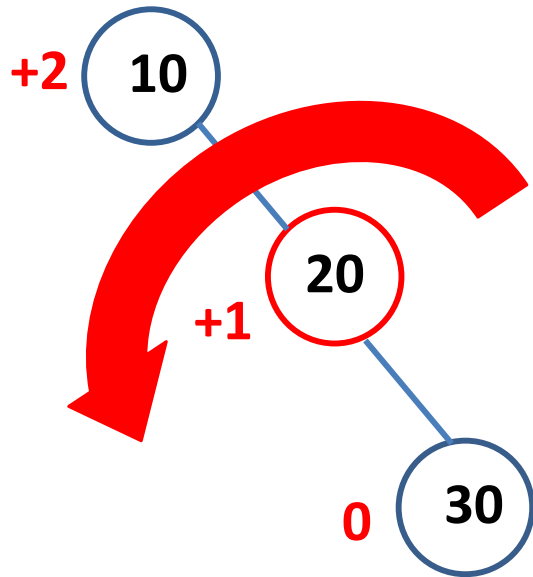
# Exemplo Rotação (Simples) à Esquerda



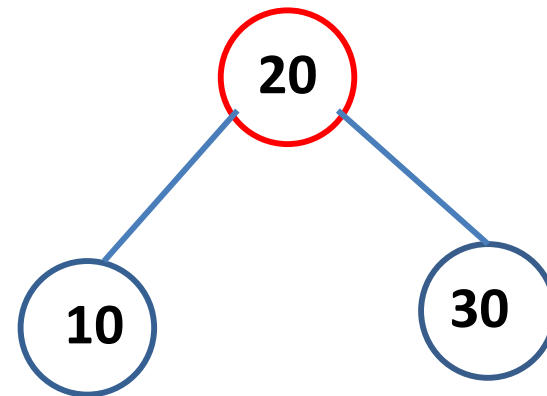
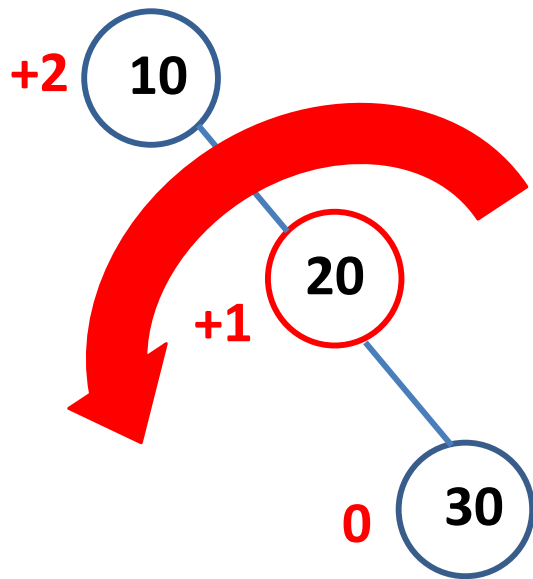
# Exemplo Rotação (Simples) à Esquerda



# Exemplo Rotação (Simples) à Esquerda



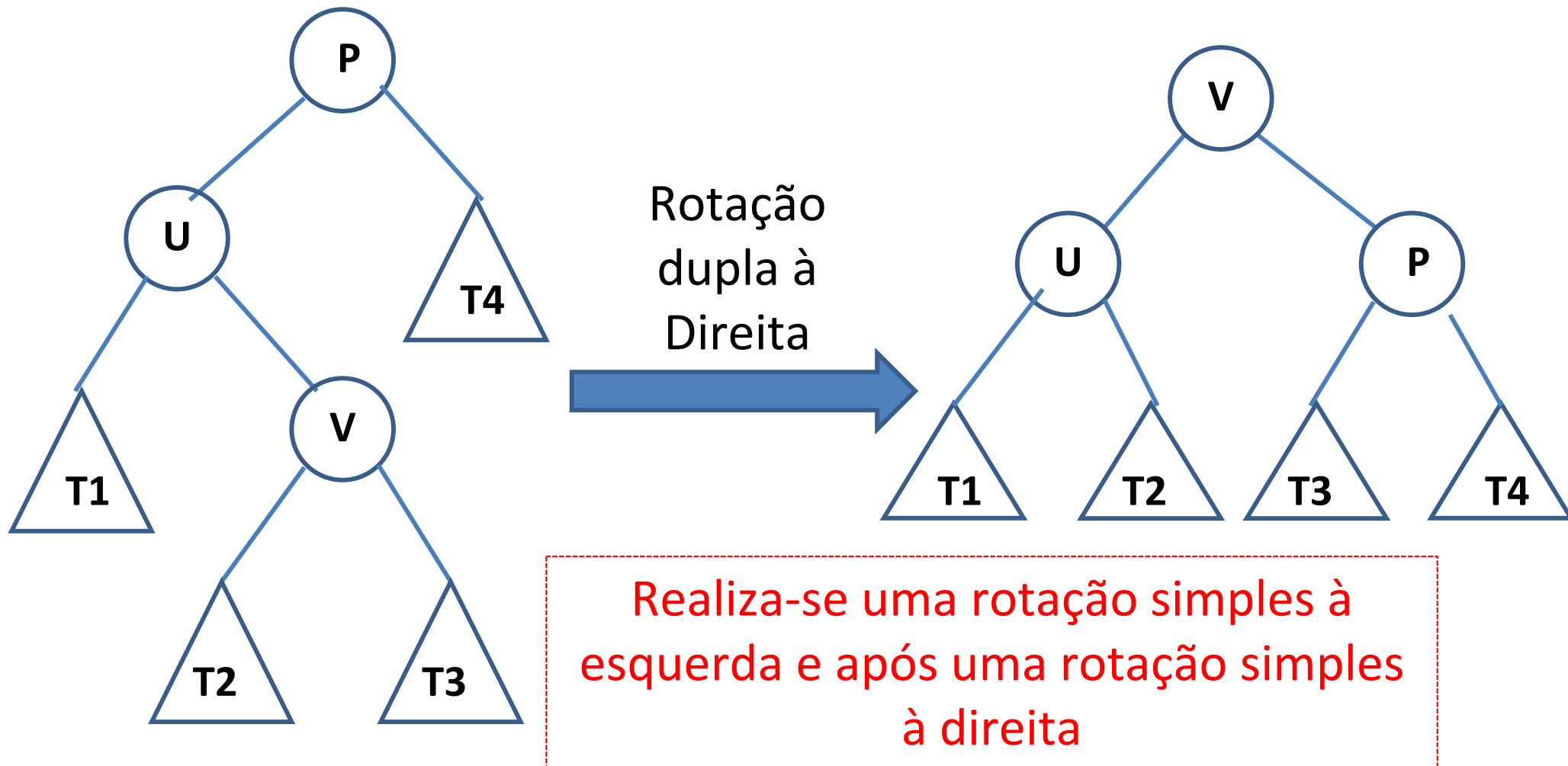
# Exemplo Rotação (Simples) à Esquerda



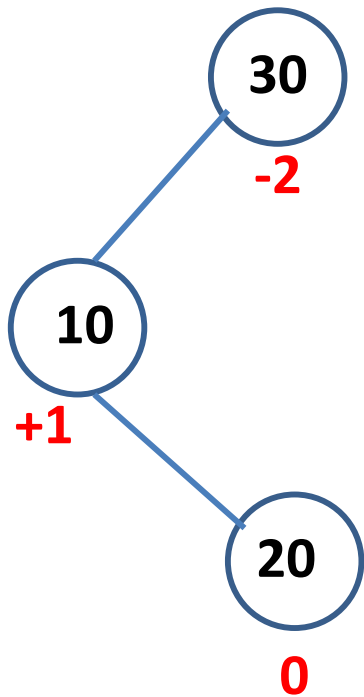


# Rotações

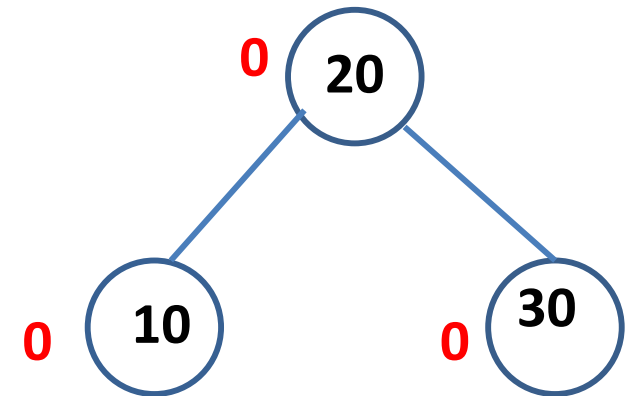
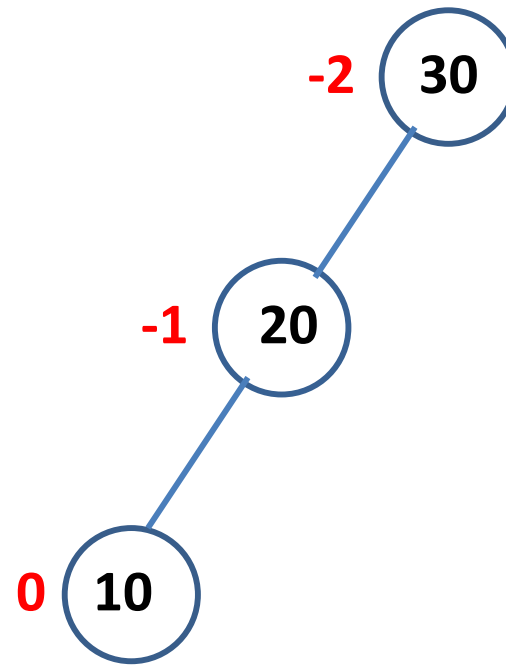
- 3. Rotação Dupla à Direita



# Exemplo Rotação dupla Direita



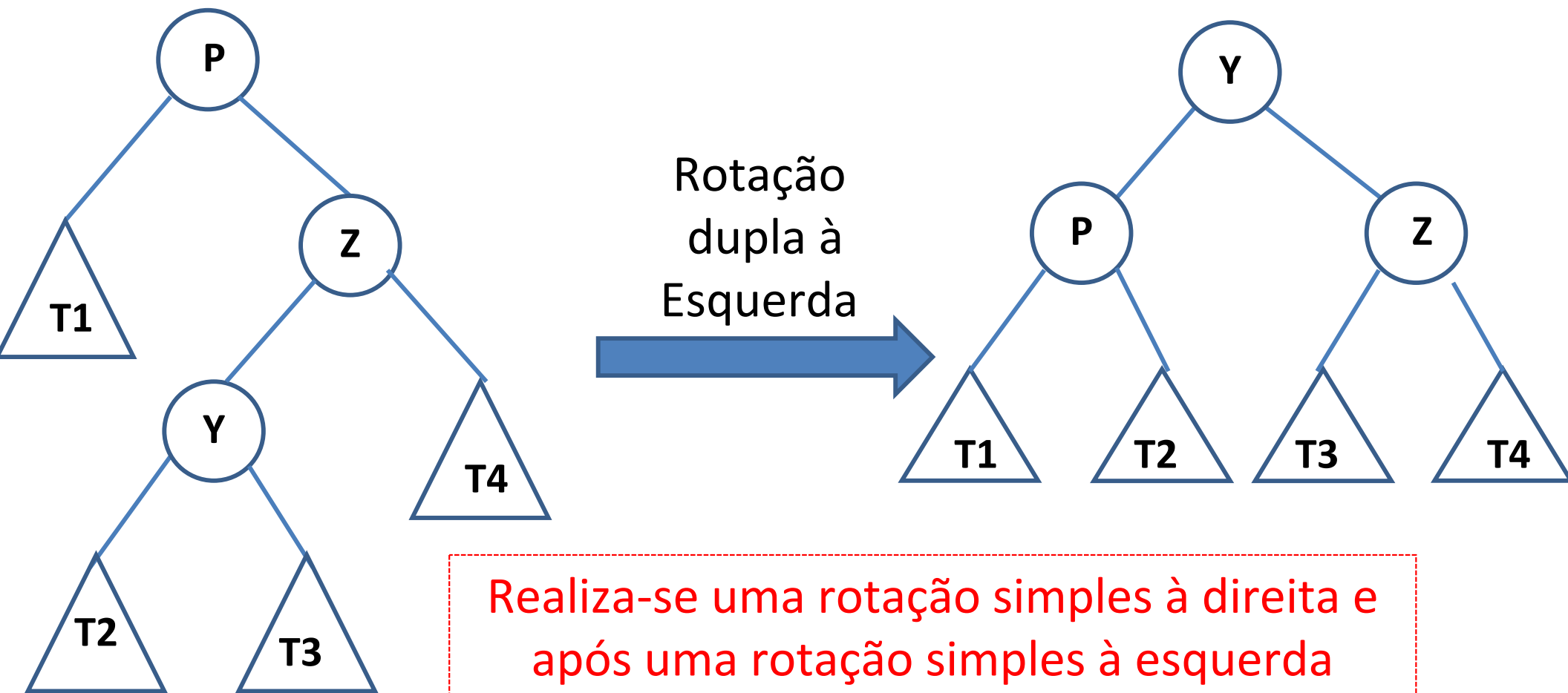
Rotação simples à esquerda



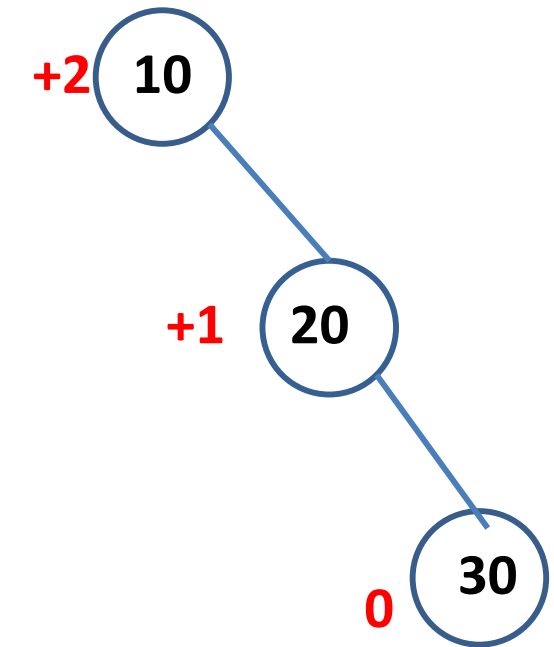
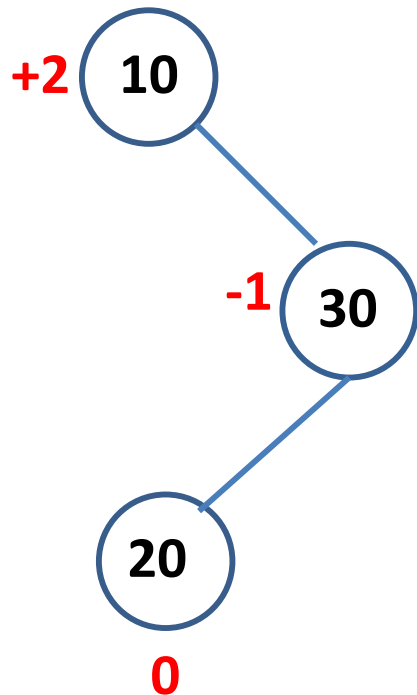
Rotação simples à direita

# Rotações

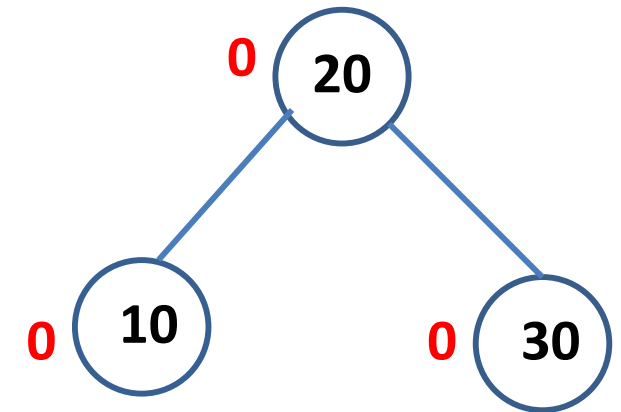
- 4. Rotação Dupla à Esquerda



# Exemplo Rotação dupla Esquerda



Rotação simples à direita



Rotação simples à esquerda

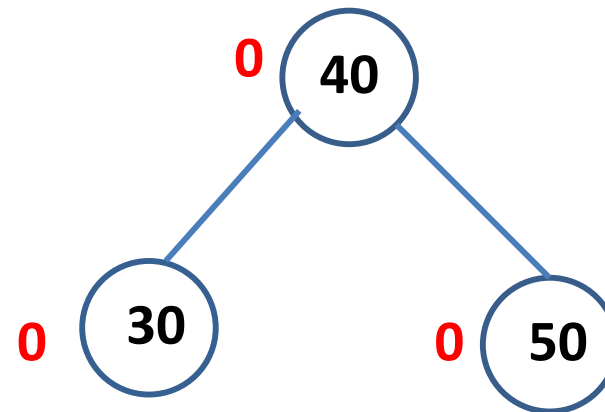
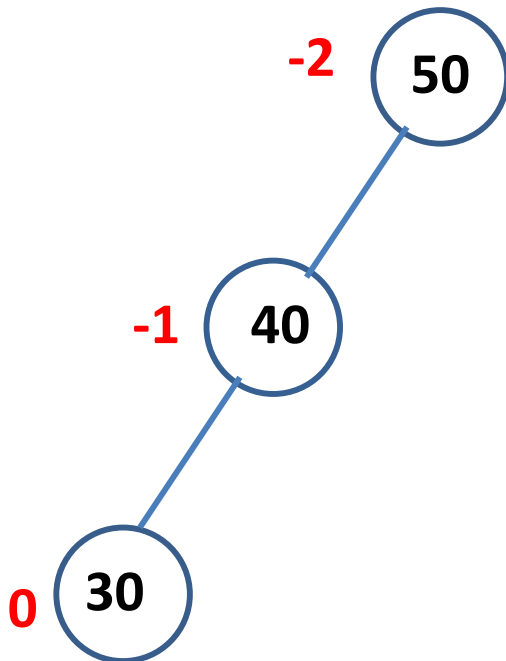
# Rotação

- Para identificar quando uma rotação é simples ou dupla devemos observar os sinais do FB:
  - a) Se o sinal for igual, a rotação é simples;
  - b) Se o sinal for diferente, a rotação é dupla.
  - c) Se o FB for positivo (+), a rotação é à esquerda;
  - d) Se o FB for negativo (-), a rotação é à direita.

# Exemplo 1

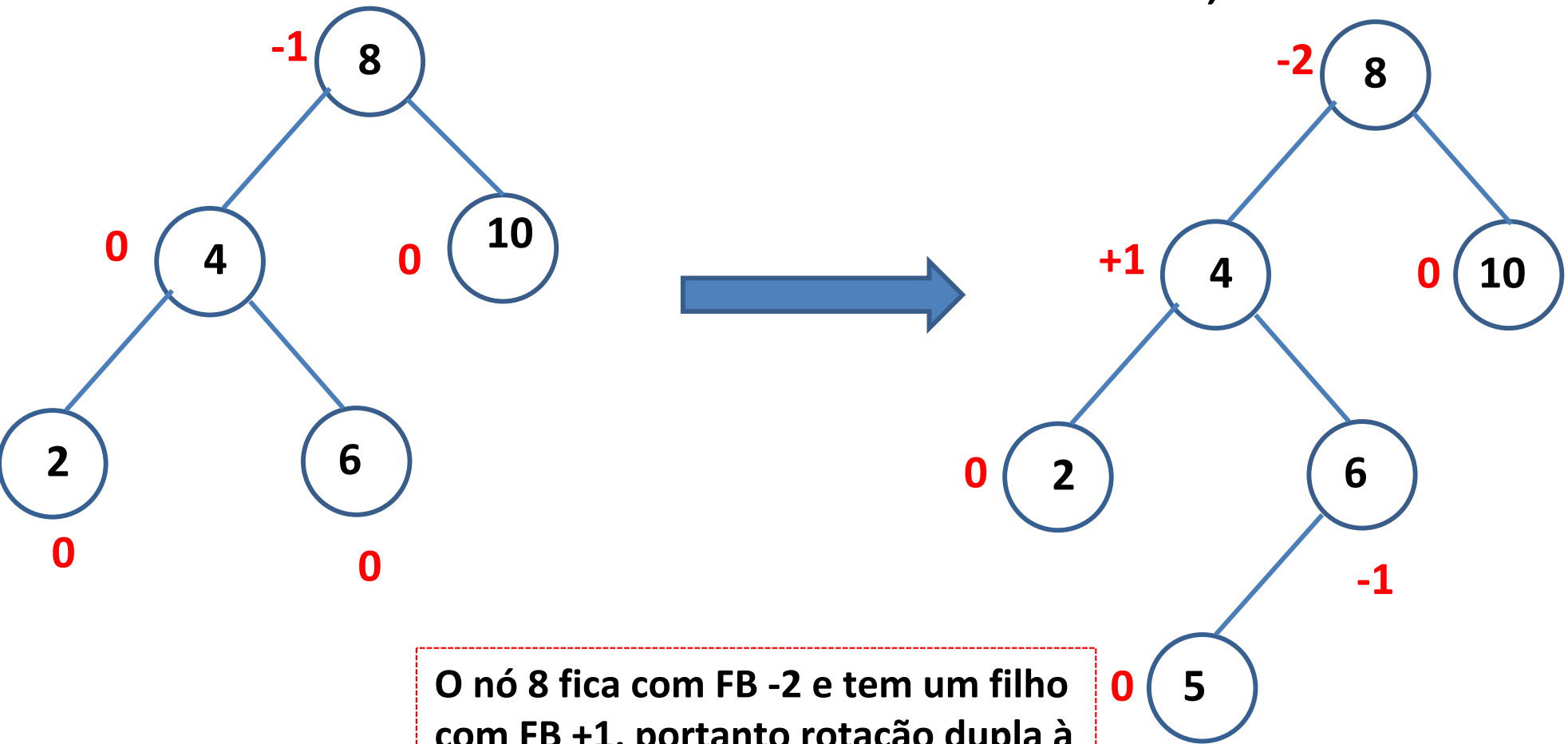
- Suponha a inserção dos nós: 50, 40, 30

A inserção produziu um desbalanceamento.  
Como os sinais dos FB são negativos o balanceamento  
deve ser simples e à direita no nó com FB -2.  
Se fosse FB +2 faríamos uma rotação simples à esquerda



# Exemplo 2

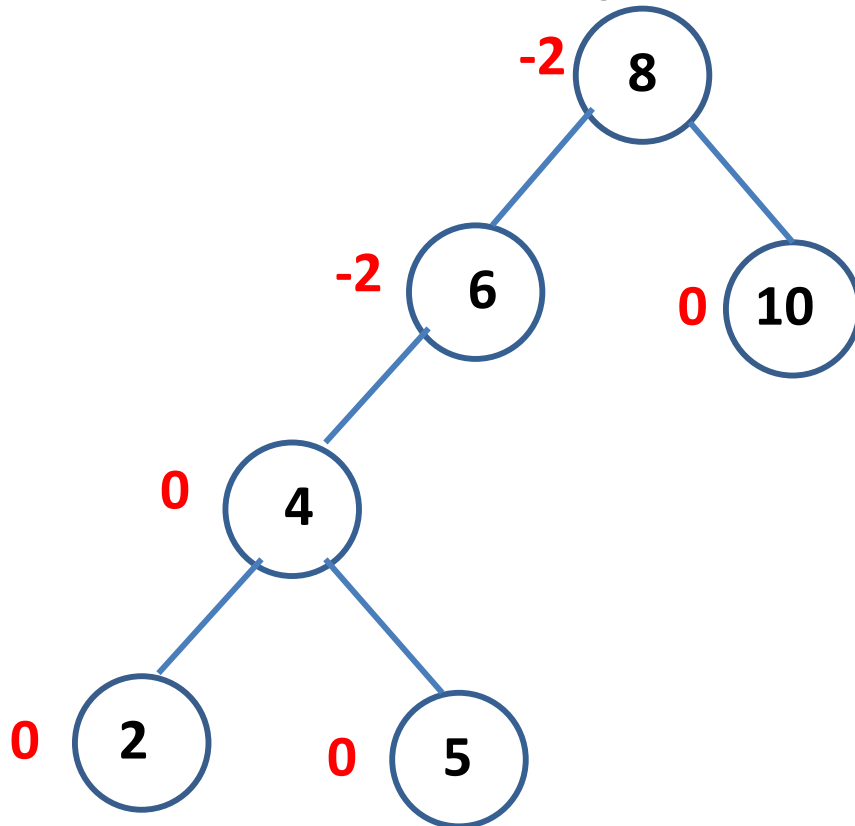
- Ao inserir o nó 5 na árvore abaixo, temos:



O nó 8 fica com FB -2 e tem um filho com FB +1, portanto rotação dupla à direita

# Exemplo 2

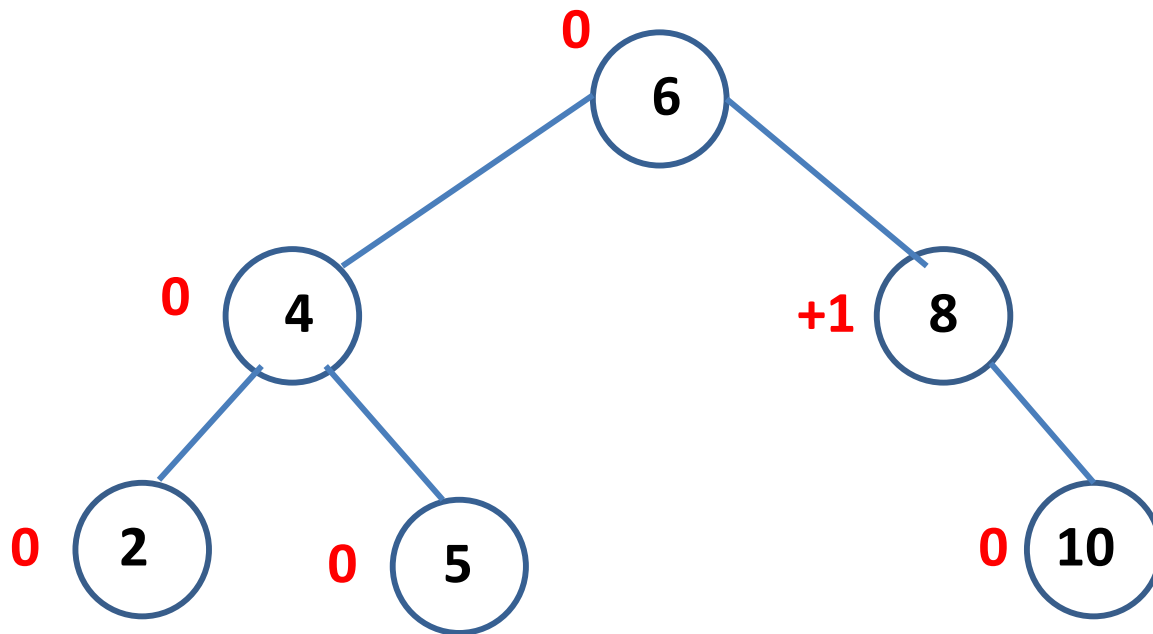
- Portanto: rotação simples à esquerda nos dá:





## Exemplo 2

- E finalmente rotacionando-se o nó 6 (FB -2) obtemos:



# Fator de Balanceamento

```
int altura(Arvore t)
{
    int altE, altD;
    if (t == NULL)
        return -1;
    else
    {
        altE = altura(t->esq);
        altD = altura(t->dir);
        if (altE < altD)
            return altD + 1;
        else
            return altE + 1;
    }
}
```

# Rotação Direita

```
void rotacao_direita(AVL *p)
{
    AVL *q, *temp;
    q = p->esq;
    temp = q->dir;
    q->dir = p;
    p->esq = temp;
}
```

# Rotação Esquerda

```
void rotacao_esquerda(AVL *p)
{
    AVL *q, *temp;
    q = p->dir;
    temp = q->esq;
    q->esq = p;
    p->dir = temp;
}
```

- Szwarcfiter, Jayme Luiz e Markenzon, Lilian. *Estruturas de Dados e seus Algoritmos*. Ed. LTC. 1994.