
Árvores AVL

Estrutura de Dados II
Profª Jacqueline Midlej

0 curso: árvores e suas variações

- Árvore Binária
- Árvore Binária de Busca
- Árvore AVL
- Árvore B

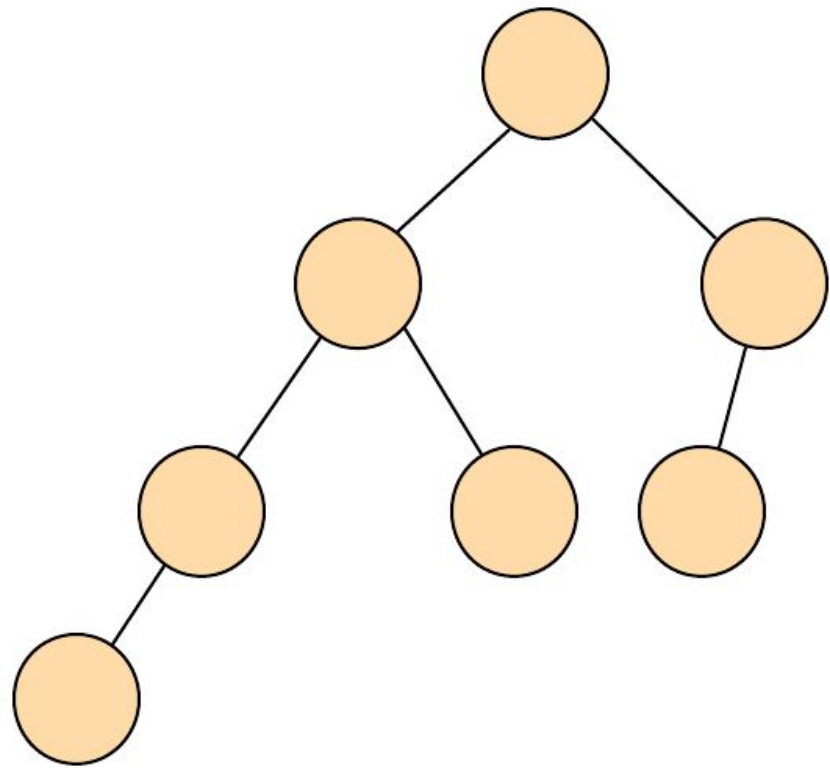
Relembrando...

- **Árvore Binária**
- **Árvore Binária de Busca**
- Árvore AVL
- Árvore B

Árvore Binária

Dados não lineares

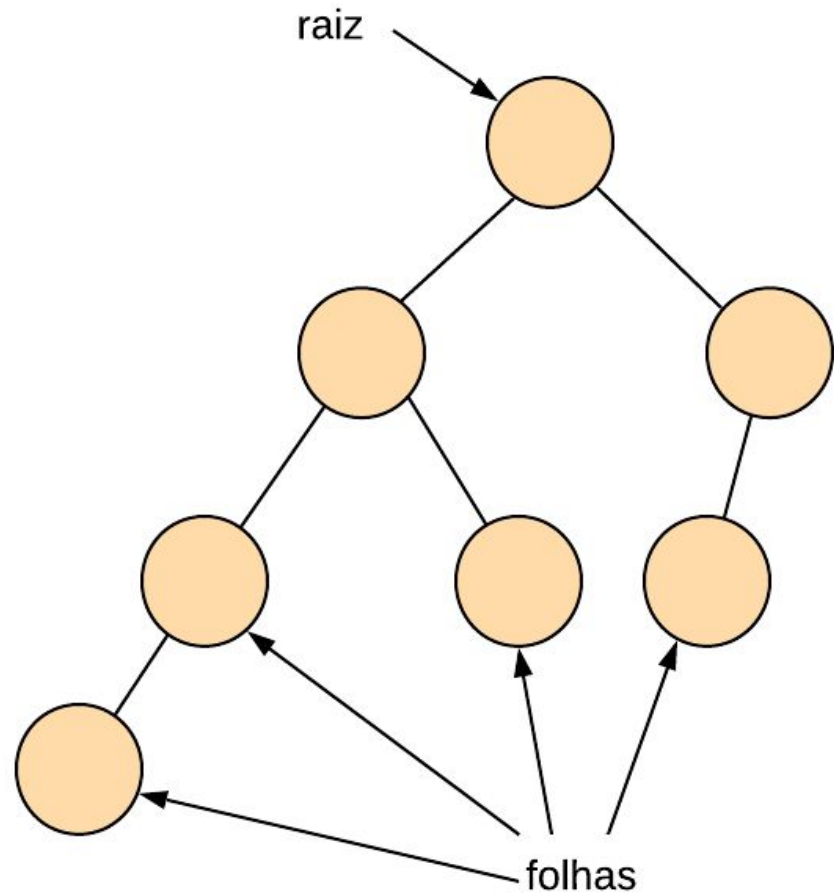
Máximo de 2 filhos



Árvore Binária

Dados não lineares

Máximo de 2 filhos



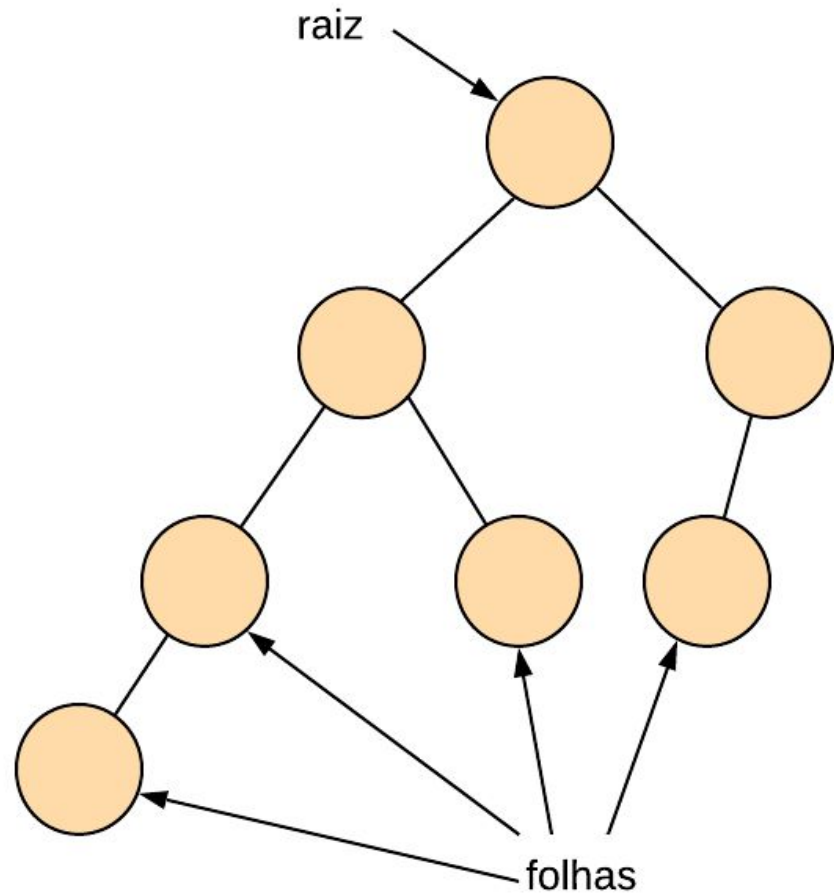
Árvore Binária

Dados não lineares

Máximo de 2 filhos

**Representação
computacional:**

```
struct No {  
    int dado ;  
  
    struct No * esq, * dir;  
}
```



Problema?

- Buscar um dado nesta árvore, no pior caso, tem tempo linear

Problema?

- Buscar um dado nesta árvore, no pior caso, tem tempo linear

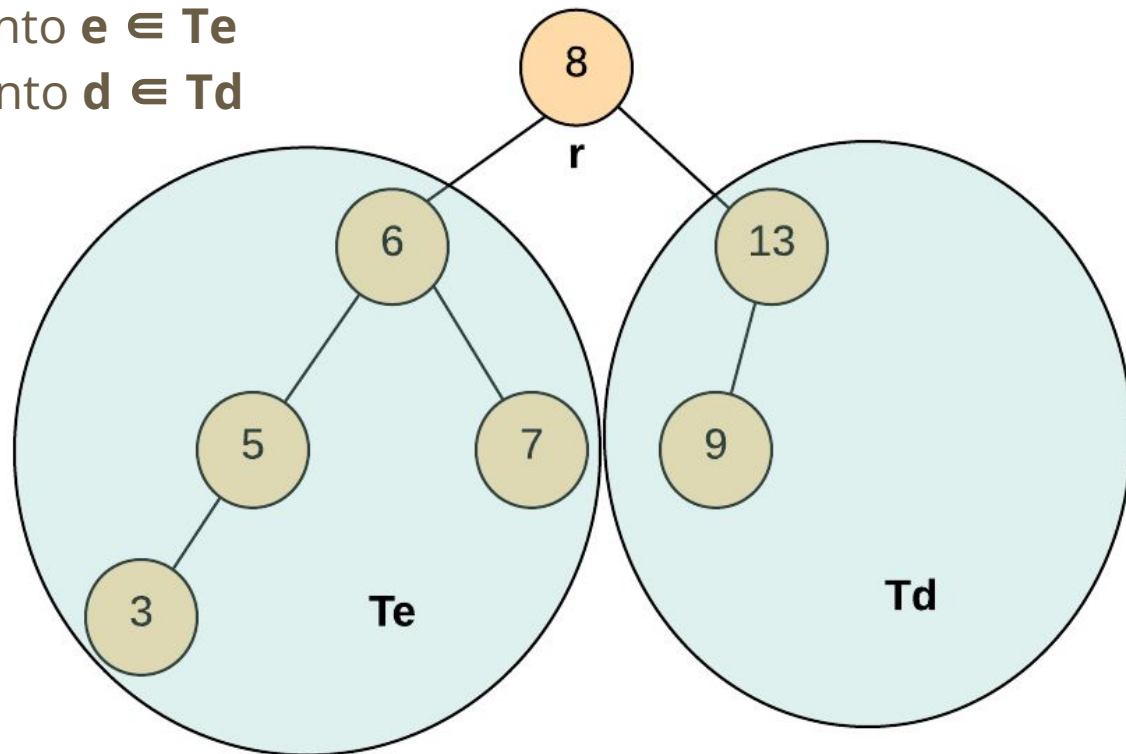
Solução inicial?

Relembrando...

- Árvore Binária
- **Árvore Binária de Busca**
- Árvore AVL
- Árvore B

Árvore Binária de Busca

- Uma **Árvore Binária de Busca (ABB)** é uma árvore binária em que cada nó contém um elemento de um **conjunto ordenável**
- Seja **Te** a subárvore esquerda de um nó **r** e **Td** a subárvore direita:
 - $e \leq r$ para todo elemento $e \in Te$
 - $r \leq d$ para todo elemento $d \in Td$



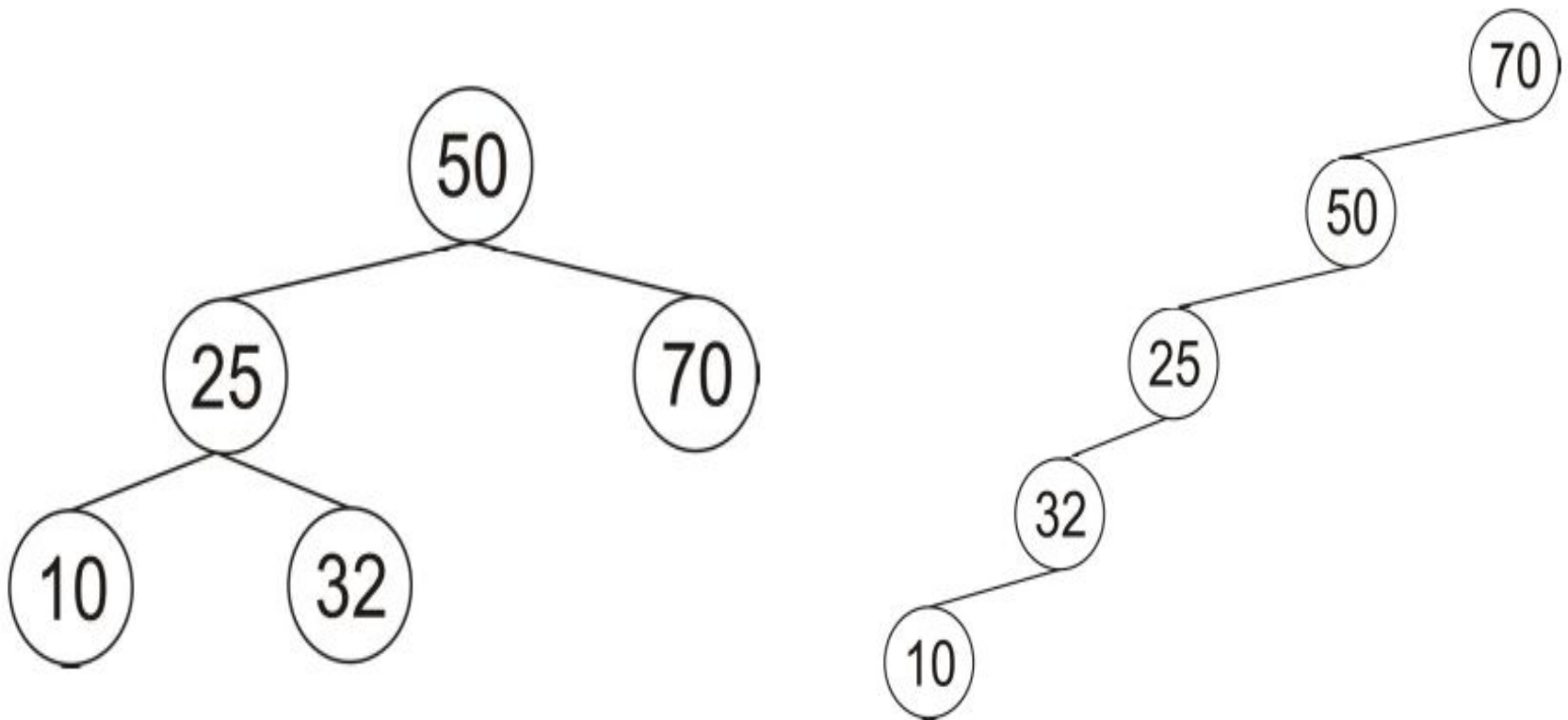
Análise

- Maior complexidade (em termos de implementação) na inserção e remoção de nós
 - precisa verificar local correto
- Menor complexidade (tempo) para busca: no caso médio $\log n$
 - A cada iteração o problema é dividido pela metade



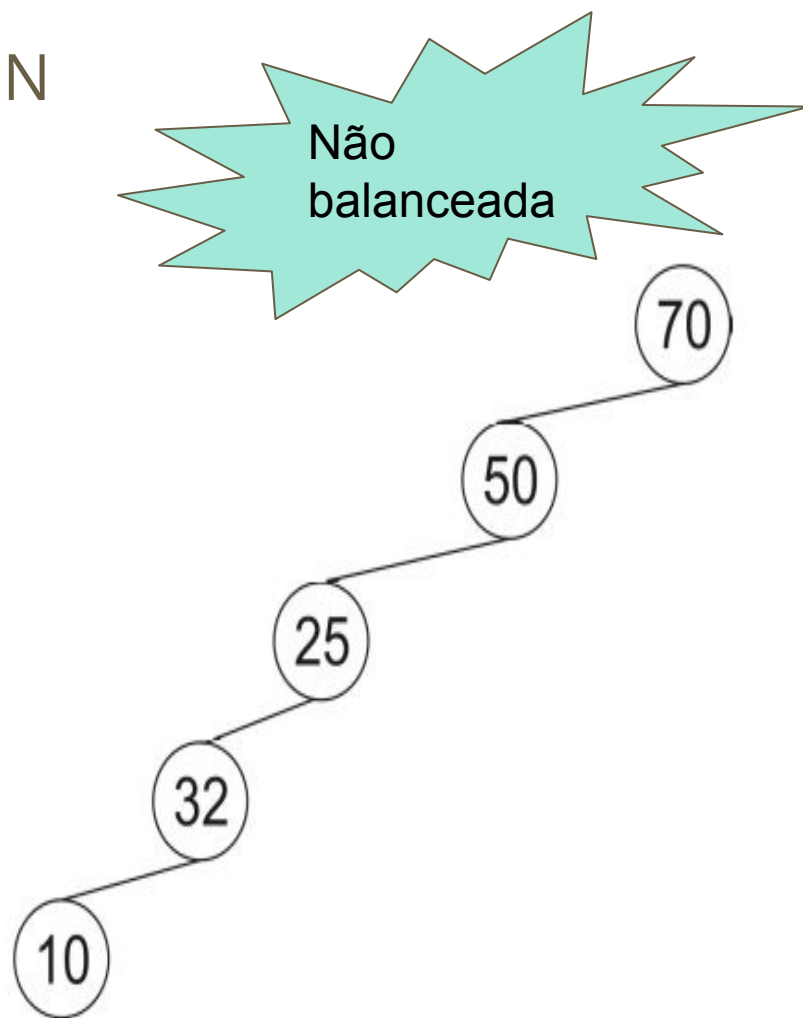
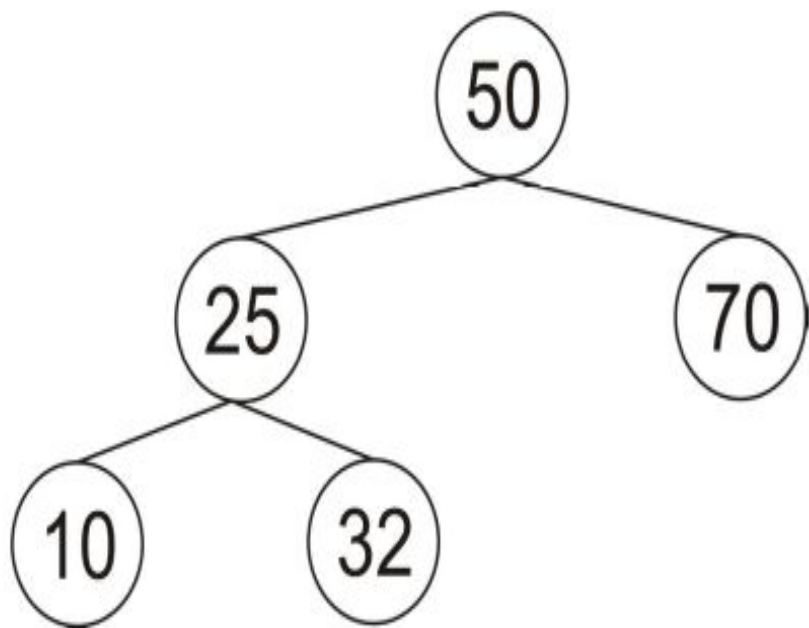
Problema?

- Pior caso tem complexidade N



Problema?

- Pior caso tem complexidade N



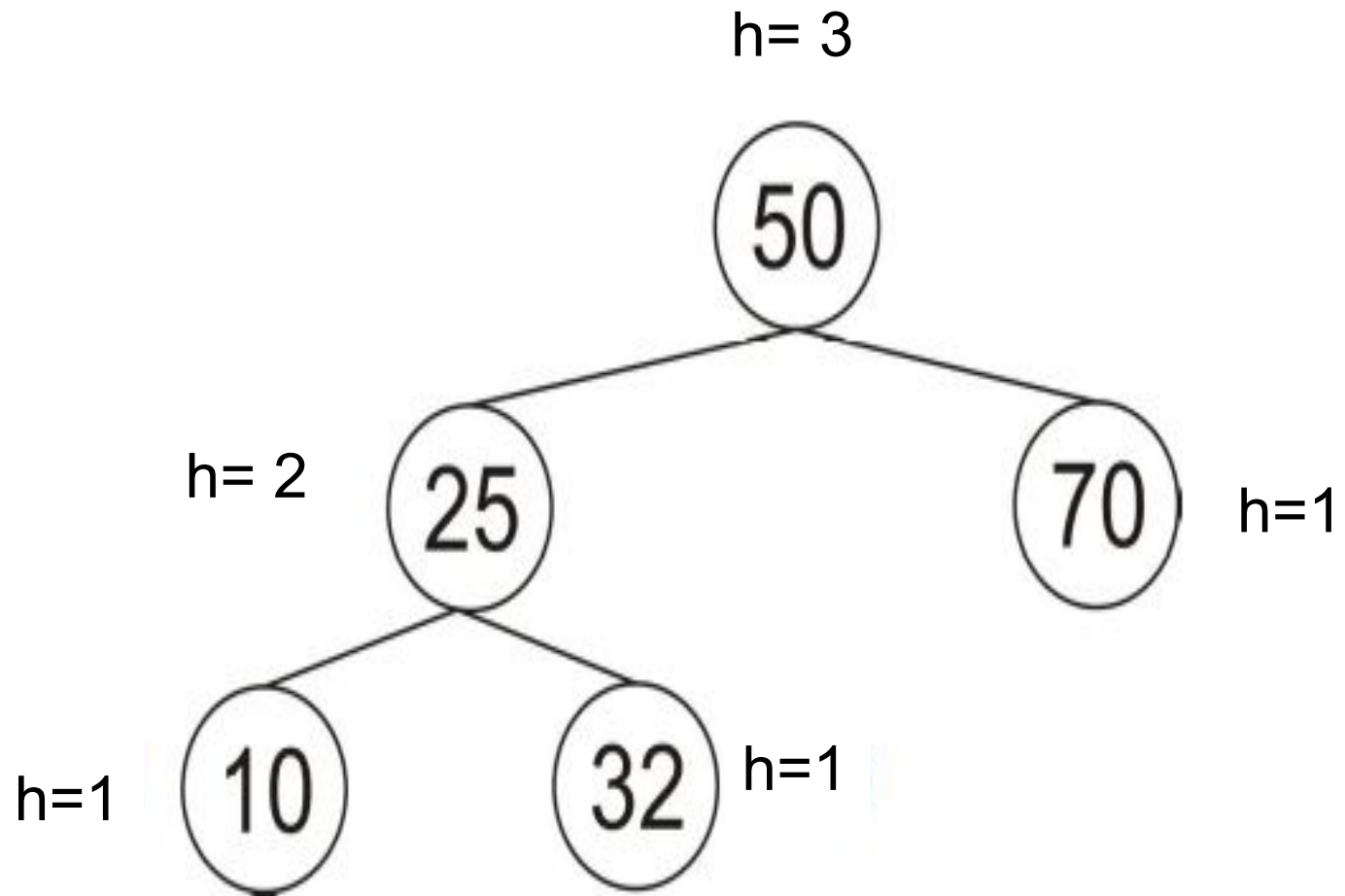
Hoje

- Árvore Binária
- Árvore Binária de Busca
- **Árvore AVL**
- Árvore B

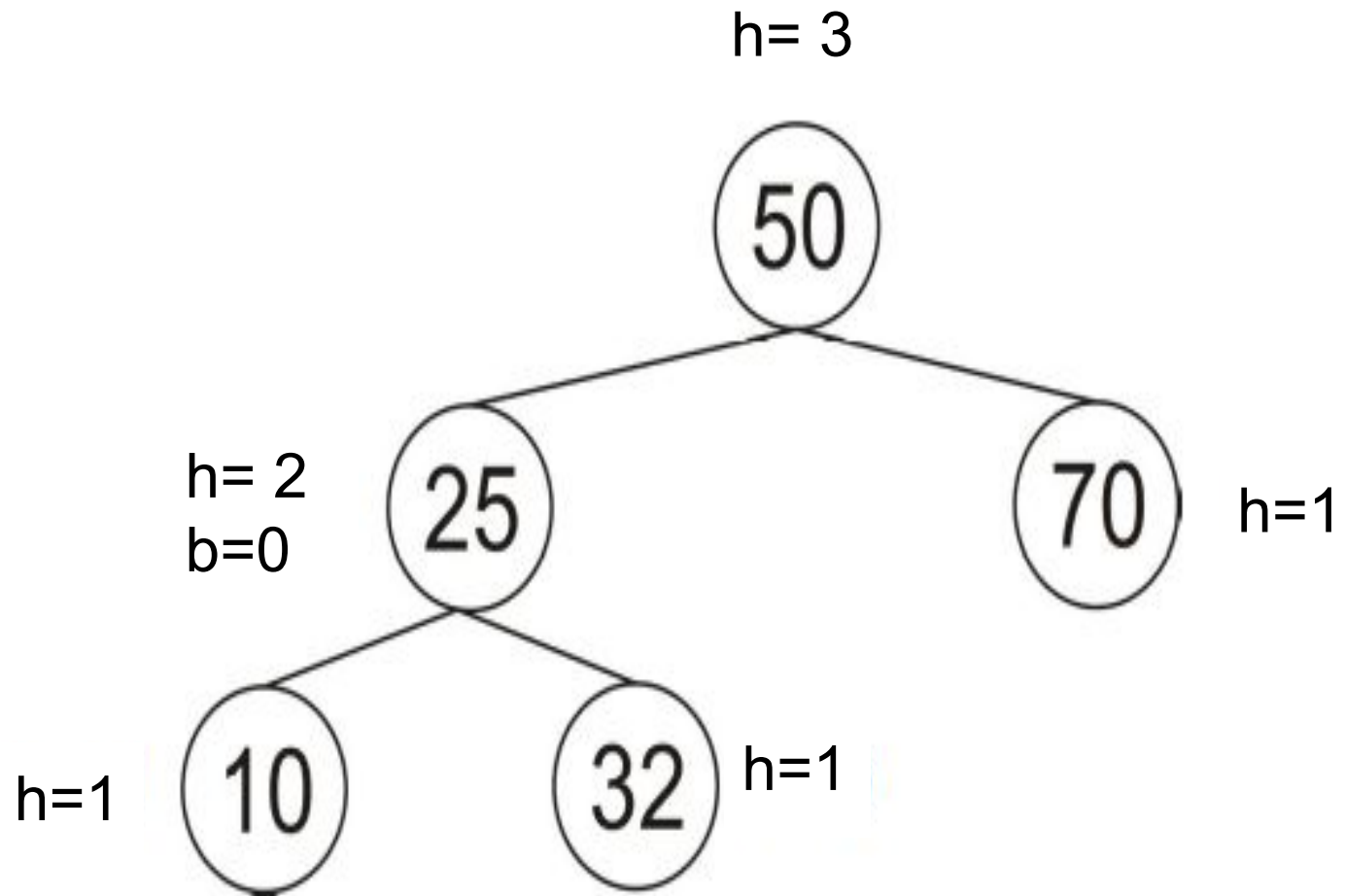
Árvore AVL

- Árvore AVL é uma árvore binária com operações para manter o **balanceamento**, dada a inserção e remoção de elementos
- Grau de balanceamento exigido na AVL:
 - $|H_e - H_d| < 2$
 - Isto é, o grau de balanceamento de cada nó deve ser -1, 0 ou 1.

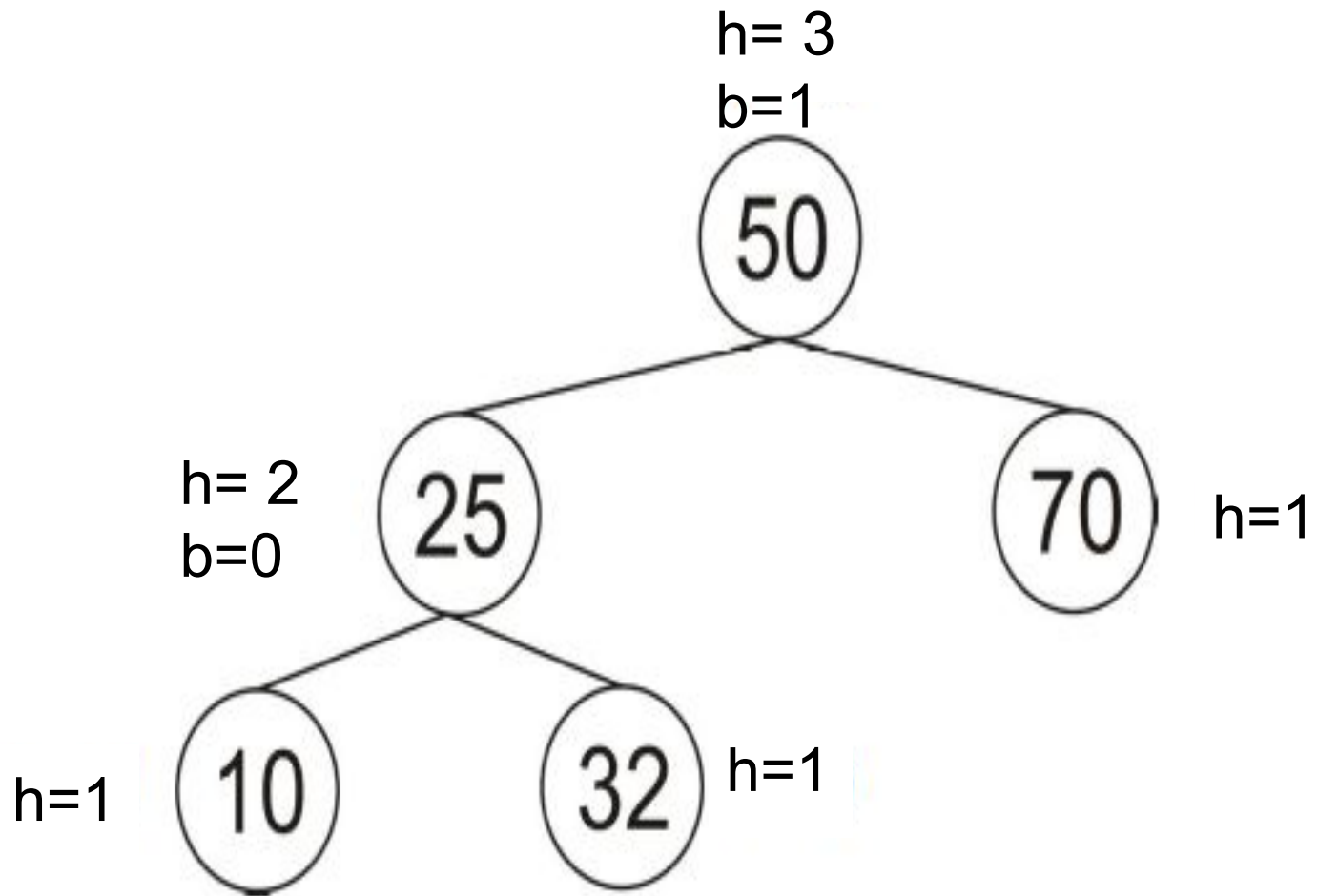
Exemplos de AVL



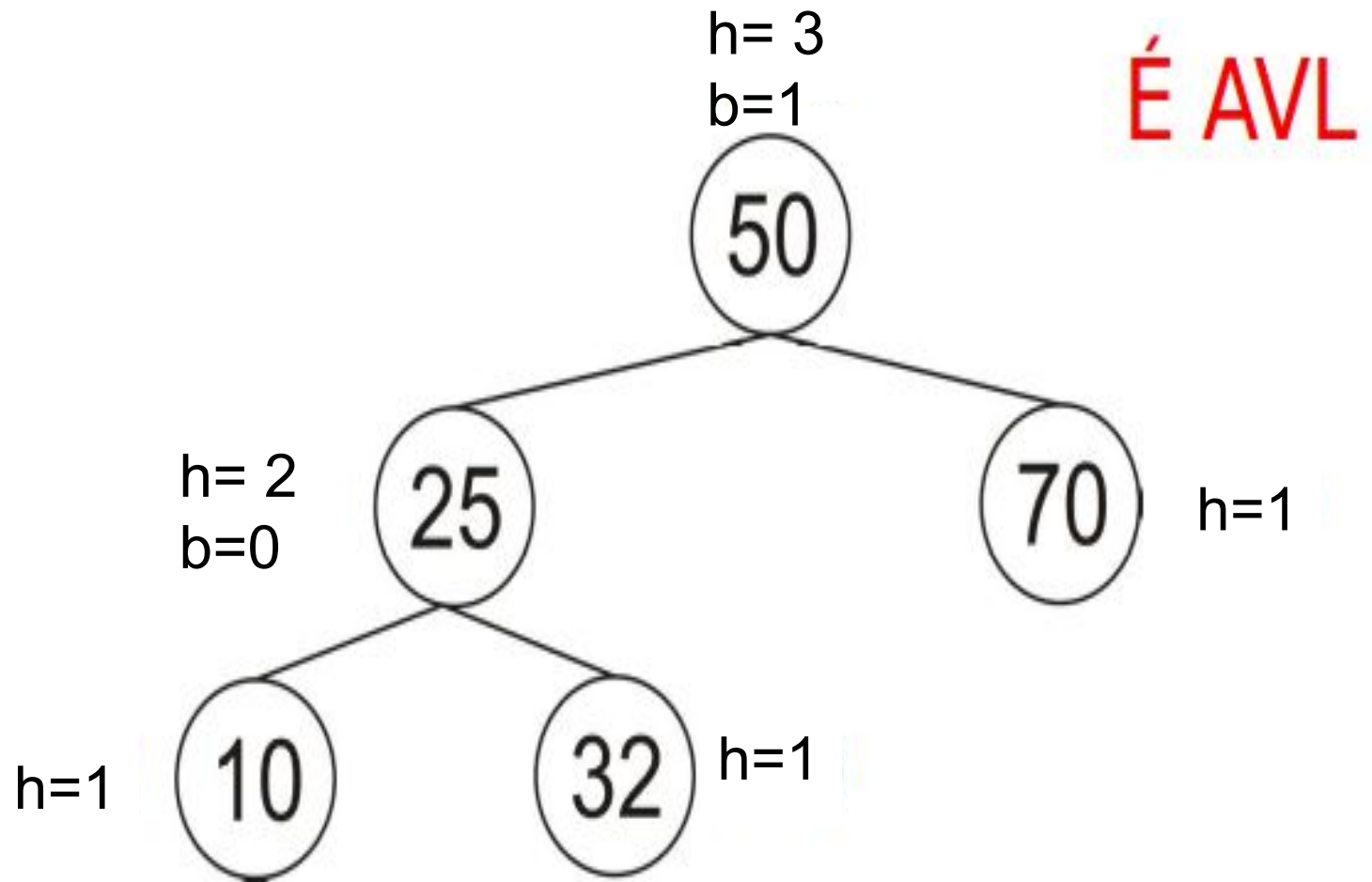
Exemplos de AVL



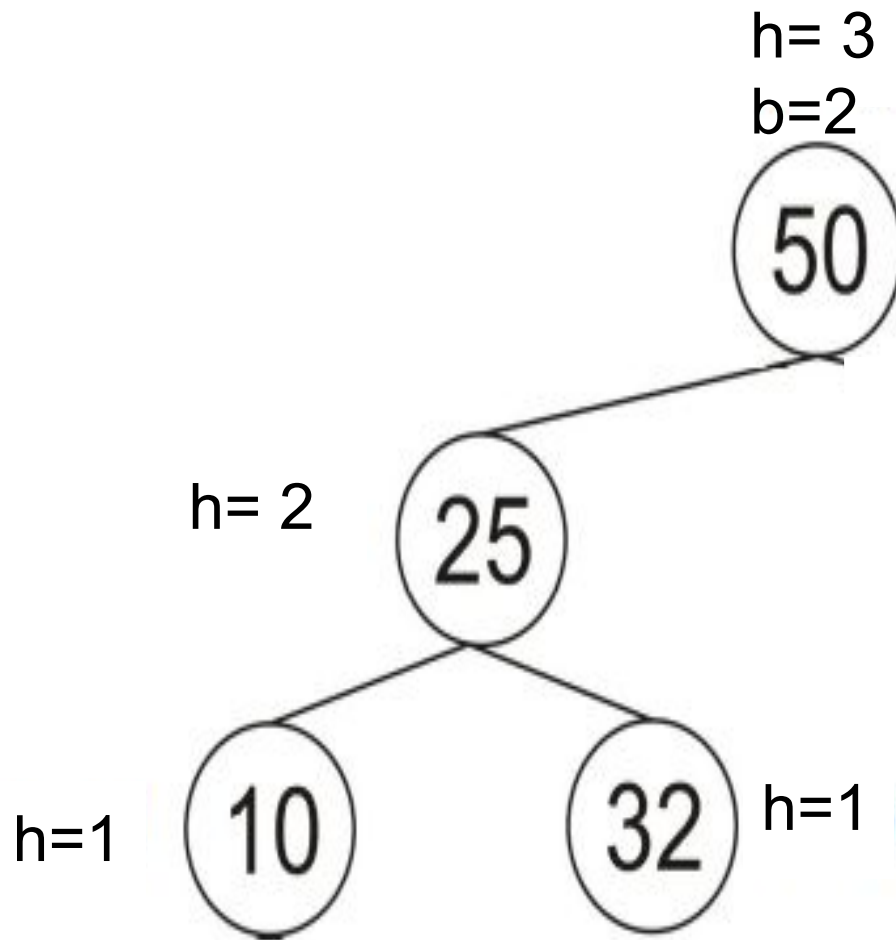
Exemplos de AVL



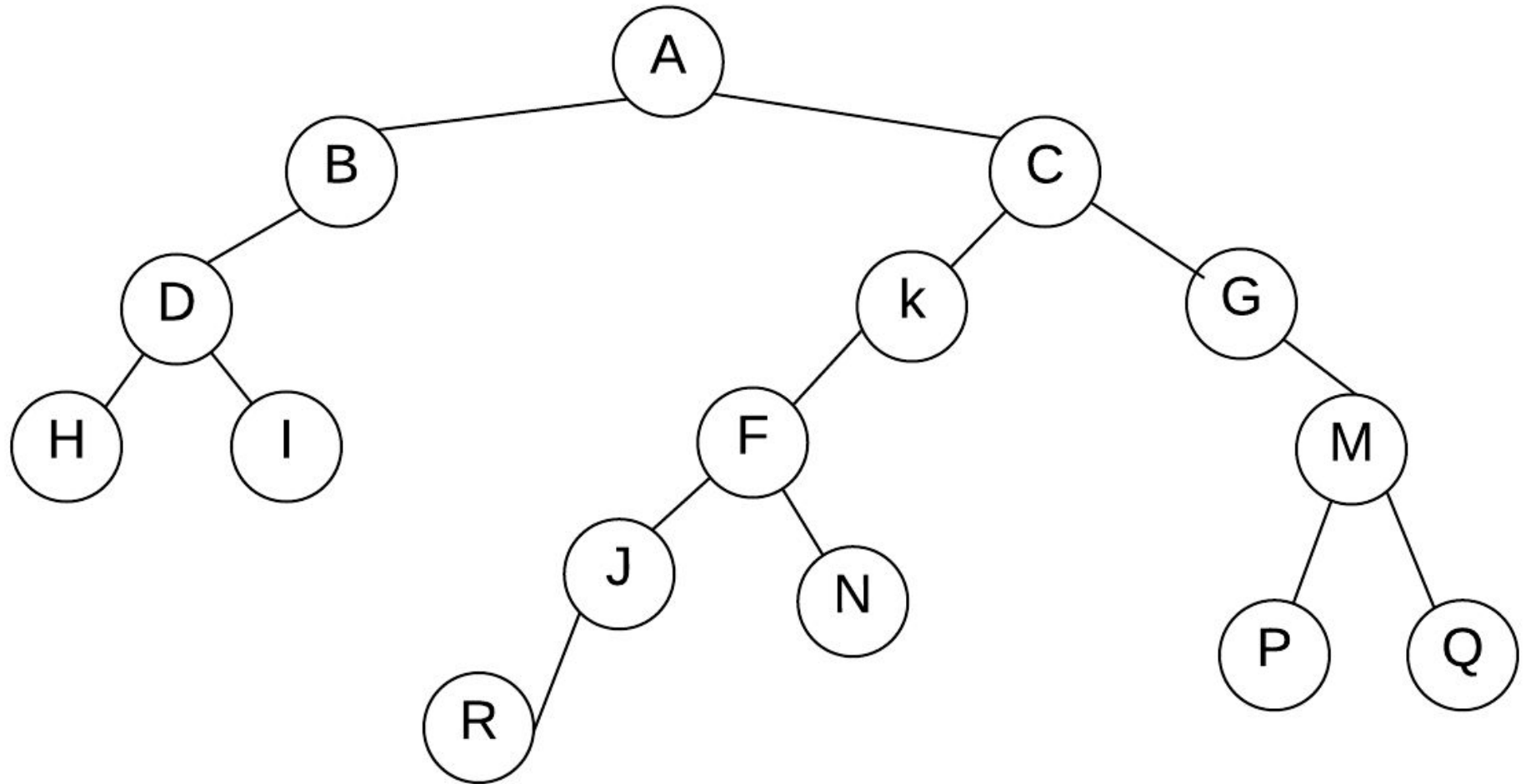
Exemplos de AVL



Exemplos de não-AVL



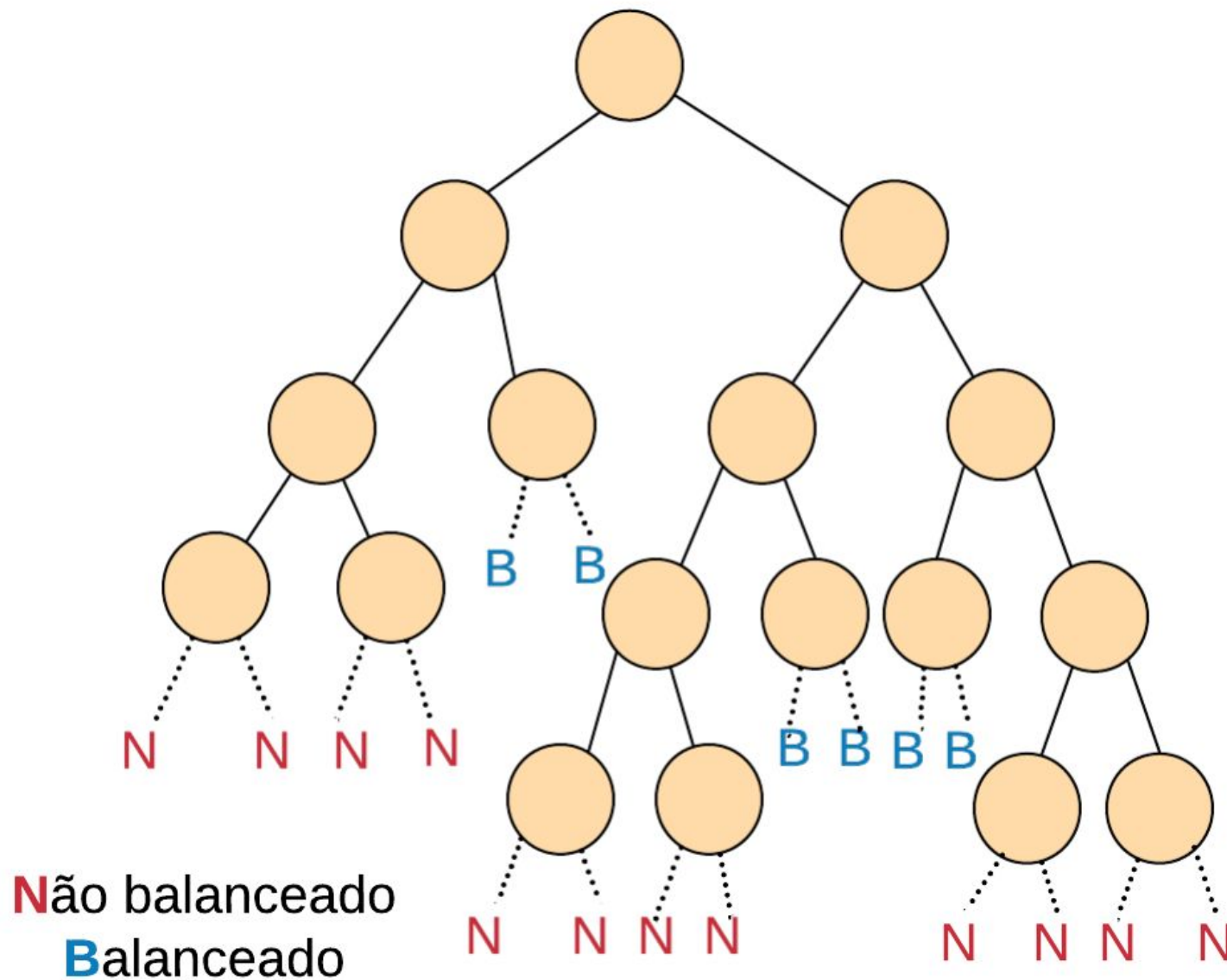
Exercício Rápido - Calcule balanceamento



Ajustando o balanceamento em ABB

- Vamos utilizar os mesmos algoritmos de inserção e remoção das árvores ABB
- No entanto, estas operações podem gerar árvores desbalanceadas
- Vamos utilizar rotações para ajustar o balanceamento nestas operações
 - As rotações devem manter as propriedades da ABB

Inserções balanceadas e não balanceadas

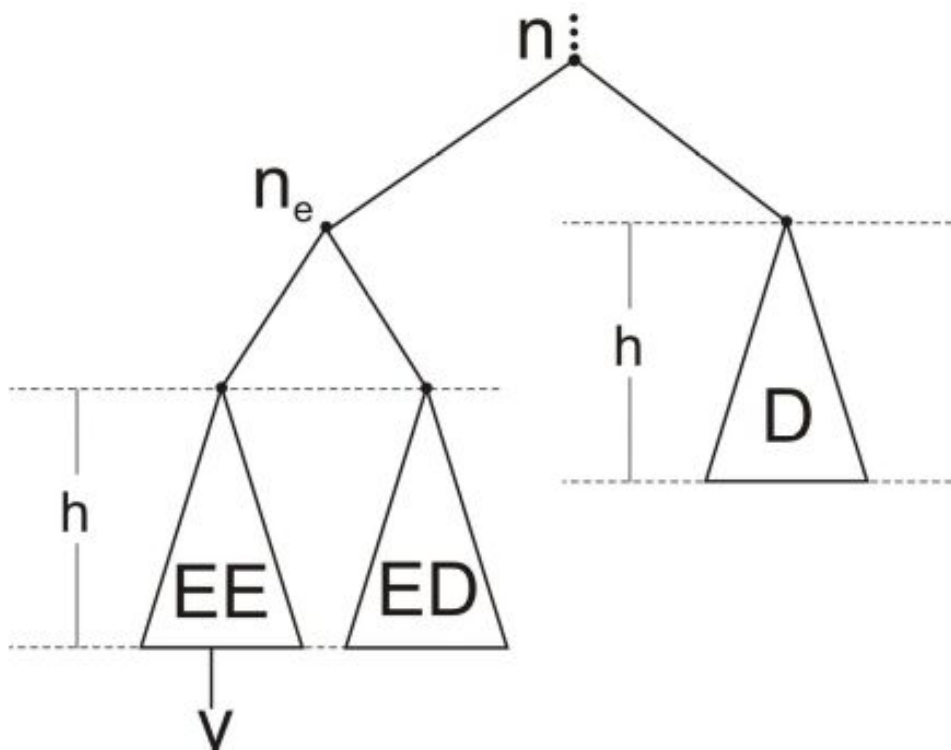


Vamos nos guiar pelos casos 4 em que ocorre desbalanceamento e como consertá-los

**1º passo, buscar o nó mais baixo desbalanceado.
Vamos chamá-lo de n**

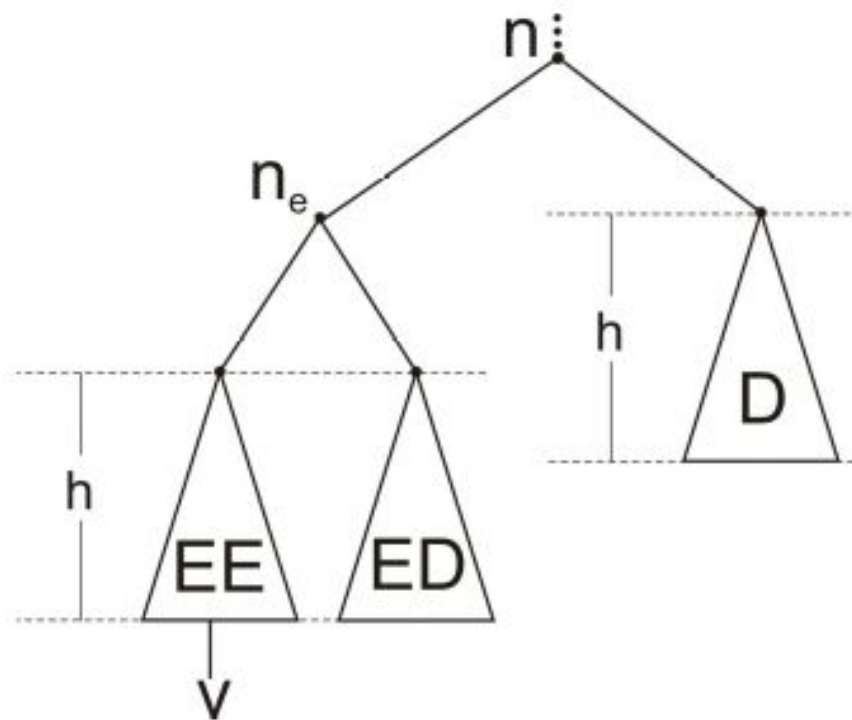
Caso A – EE

- A inserção ocorreu na **subárvore esquerda** do nó à esquerda n (n_e)
- **IMPORTANTE:** n é sempre o nó mais baixo que está desbalanceado

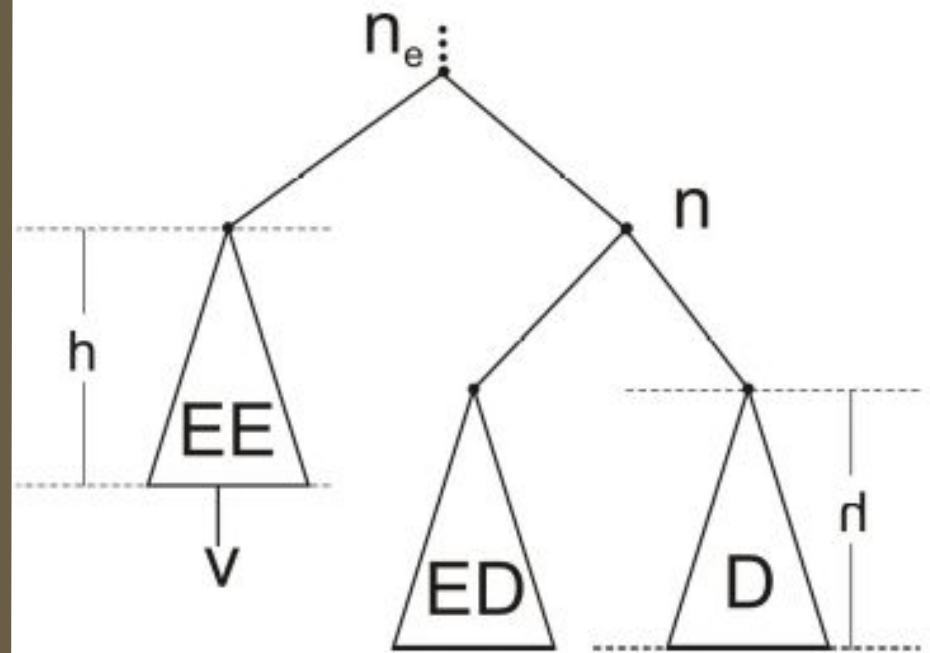


Rotação para direita (simples)

antes



depois



Passos

- n torna-se filho da direita de n_e (**desce n**)
- n_e é colocado na raiz (**sobe n_e**)

Passos

- n torna-se filho da direita de n_e (**desce n**)
- n_e é colocado na raiz (**sobe n_e**)
- EE permanece a sub-árvore esquerda de n_e
- D permanece a sub-árvore direita de n



(mantém subárvores das extremidades esquerdas e direitas)

Passos

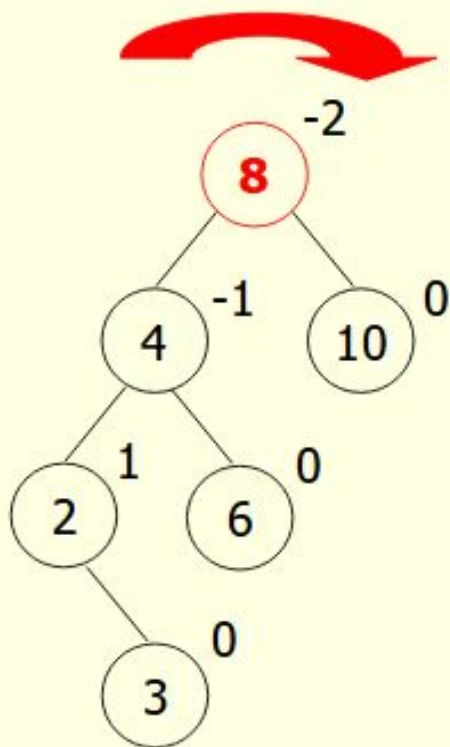
- n torna-se filho da direita de n_e (**desce n**)
- n_e é colocado na raiz (**sobe n_e**)
- EE permanece a sub-árvore esquerda de n_e
- D permanece a sub-árvore direita de n



(**mantém subárvores das extremidades esquerdas e direitas**)

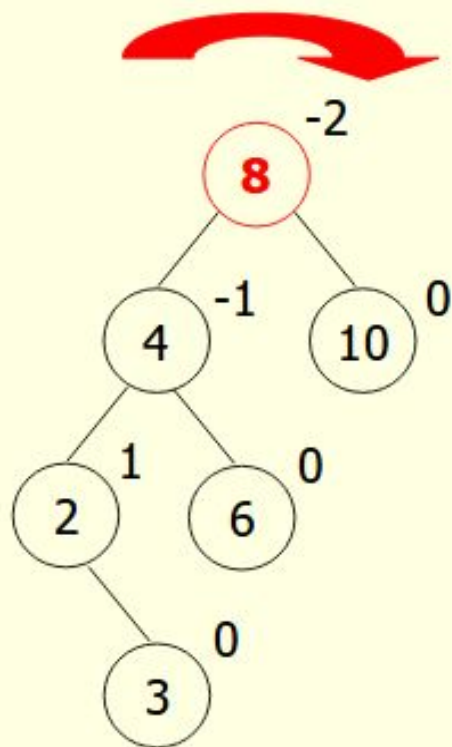
- ED torna-se filho da esquerda de n (**inverte lado da árvore do meio**)

Exemplo

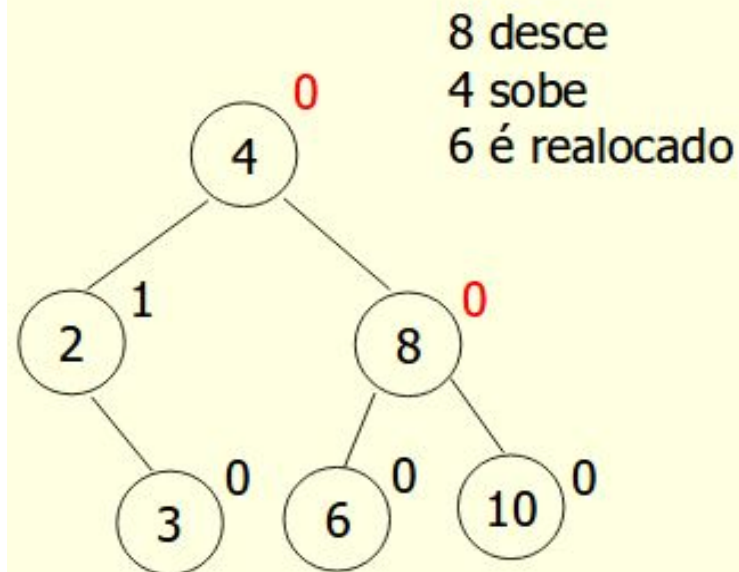


Pendendo para
a esquerda

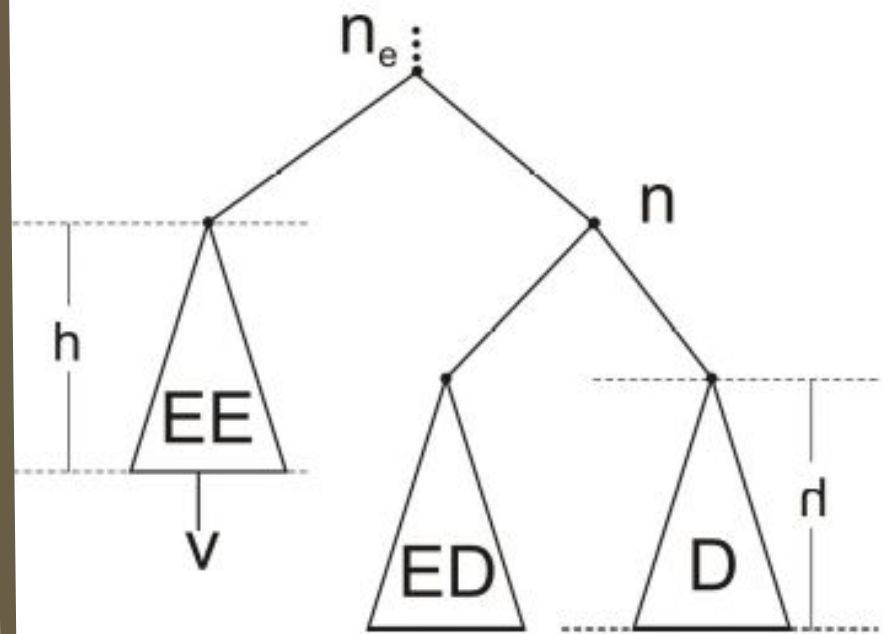
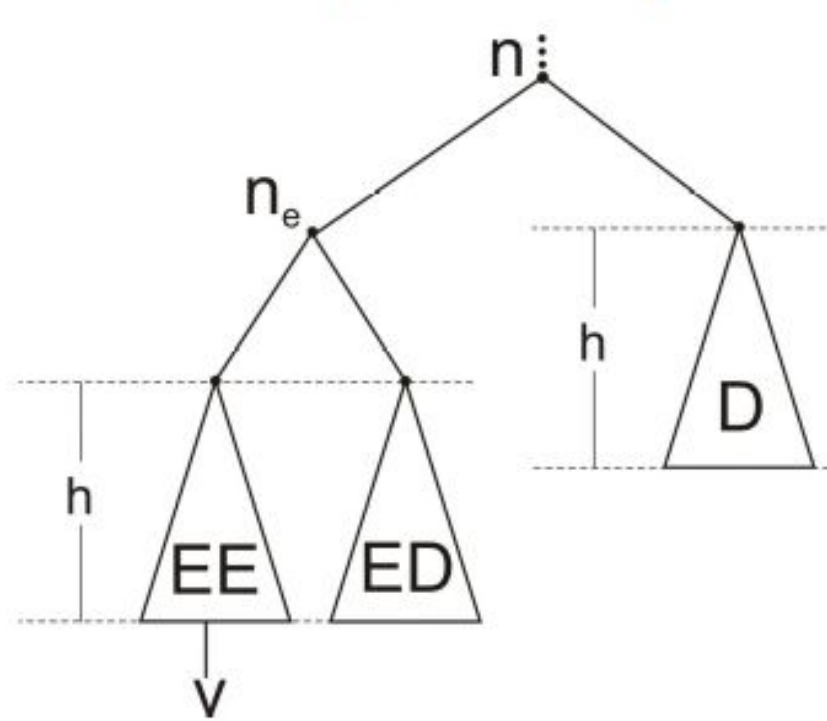
Exemplo



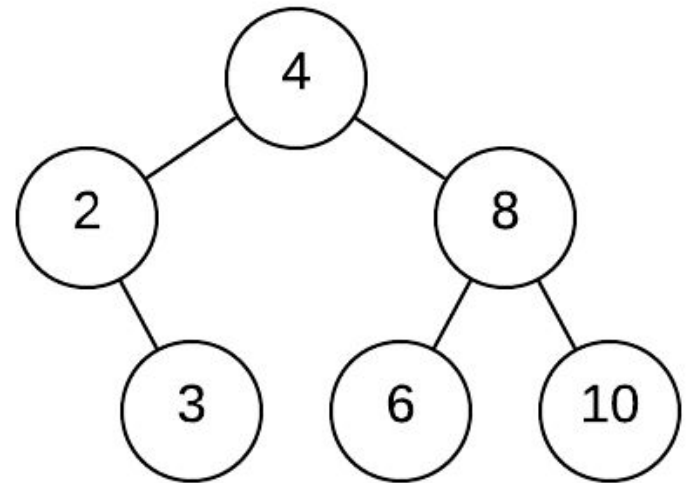
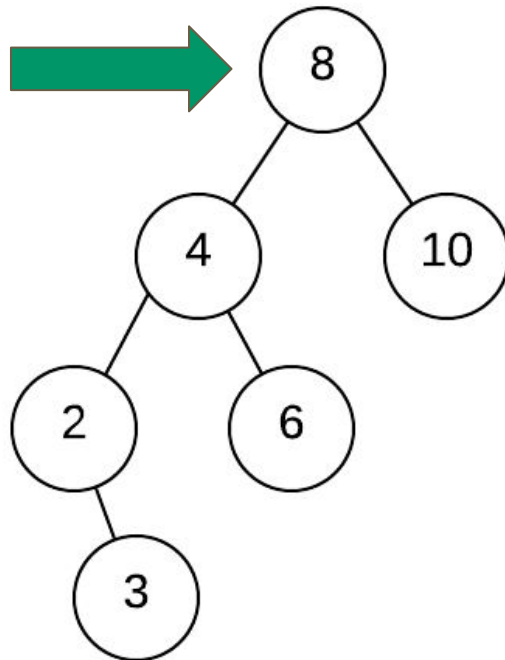
Pendendo para
a esquerda

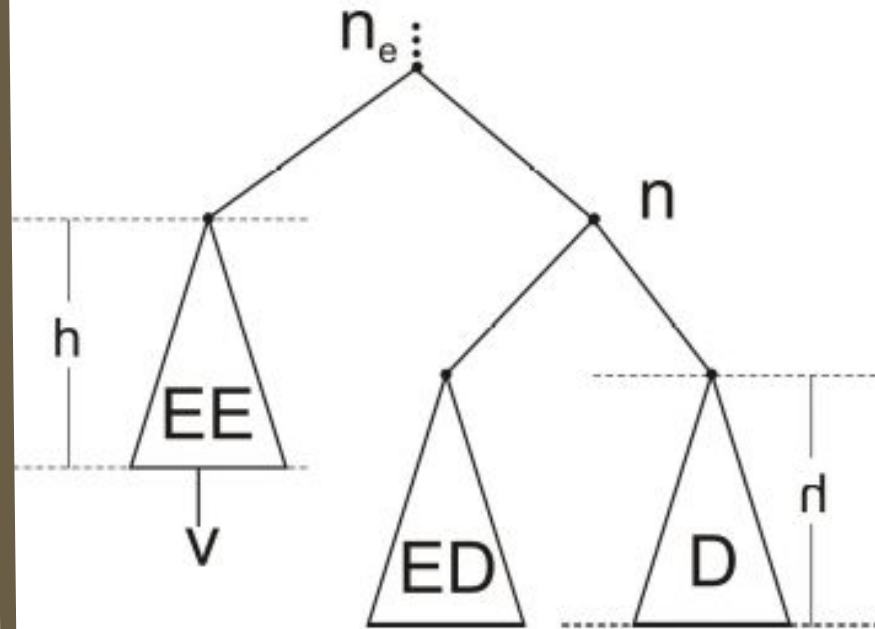
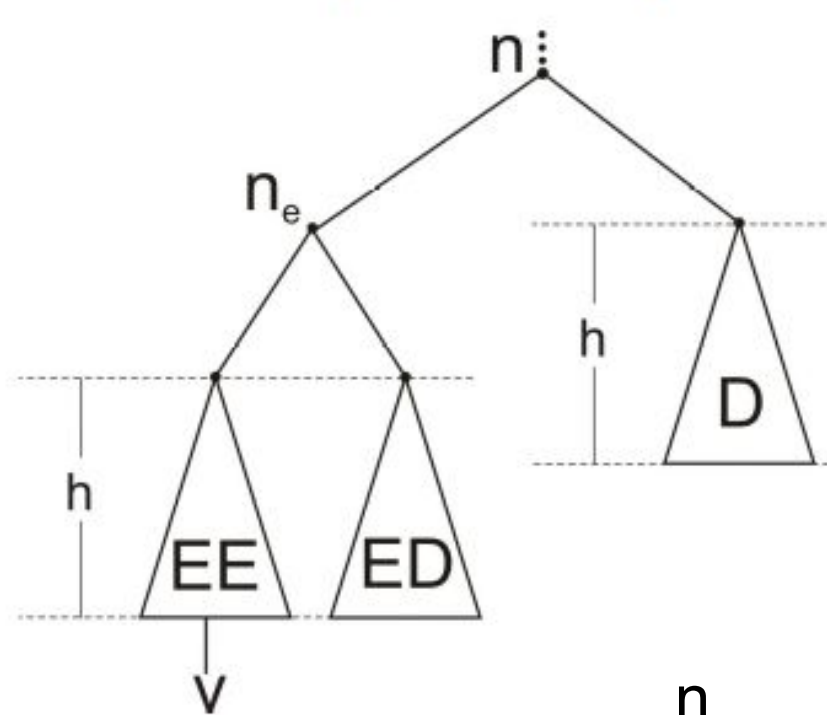


Árvore balanceada!!!

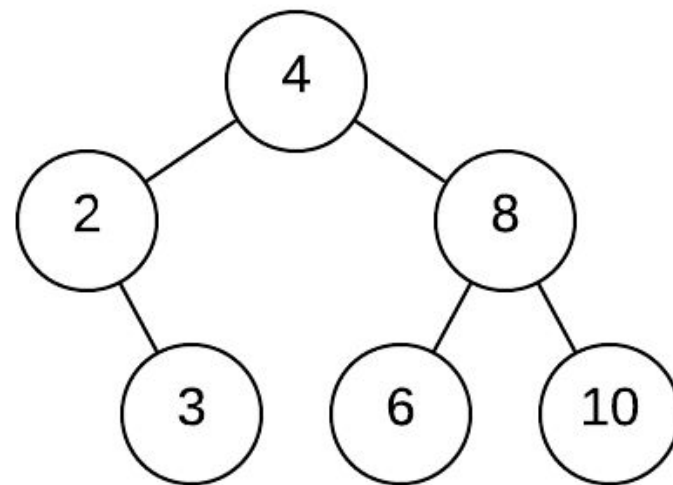
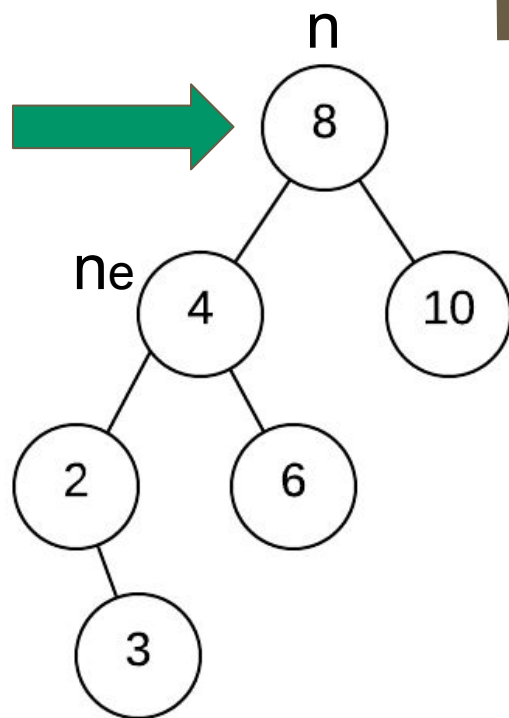


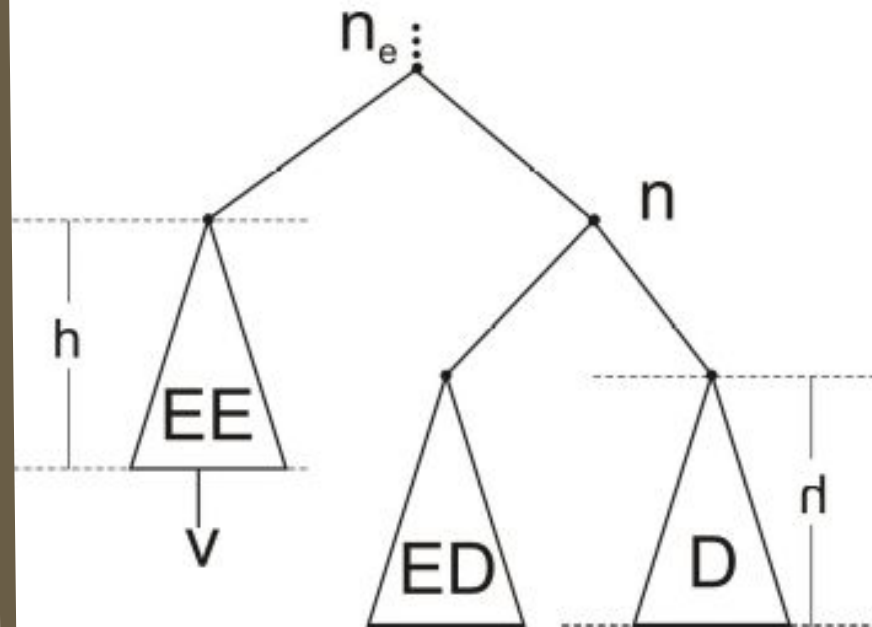
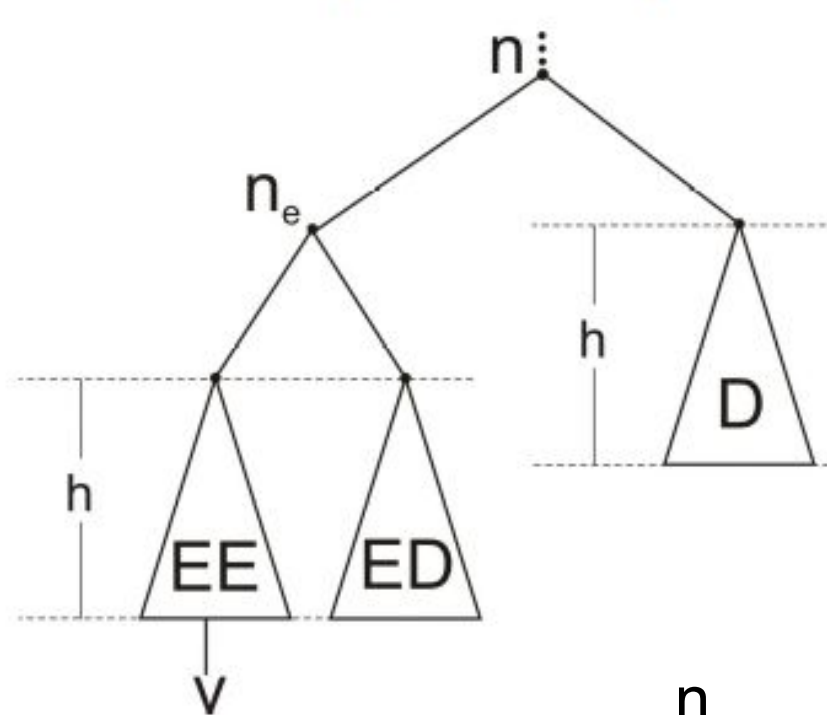
**n é sempre o
nó que está
desbalanceado**



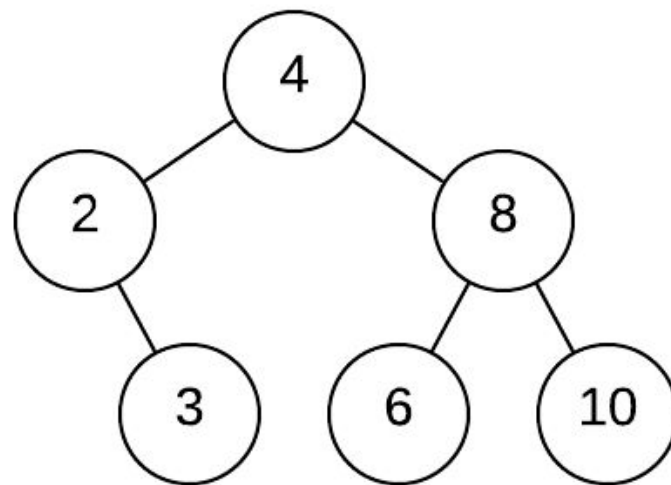
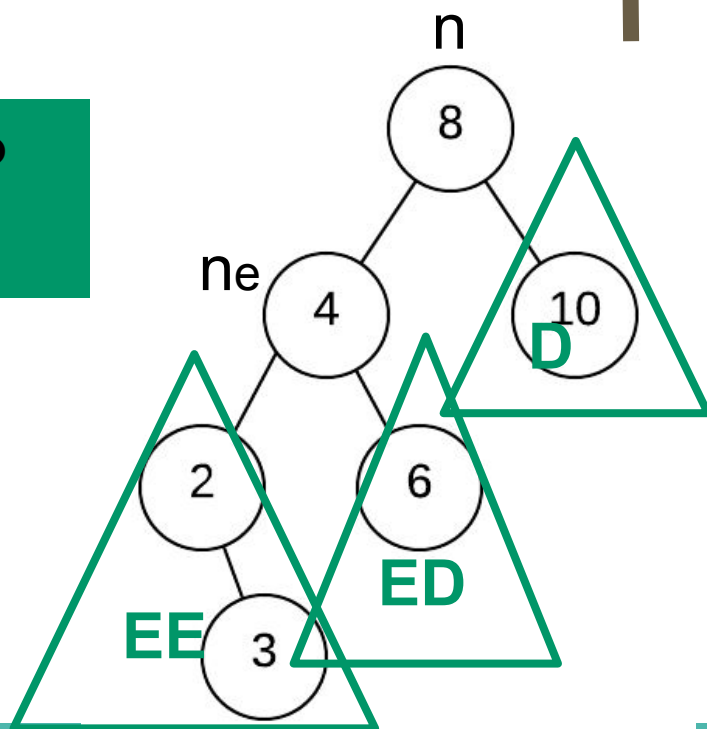


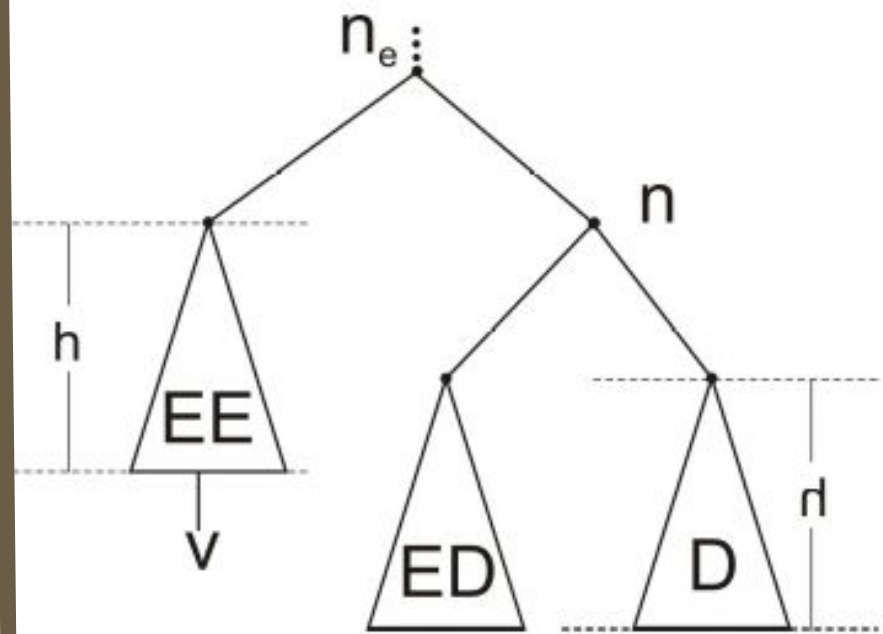
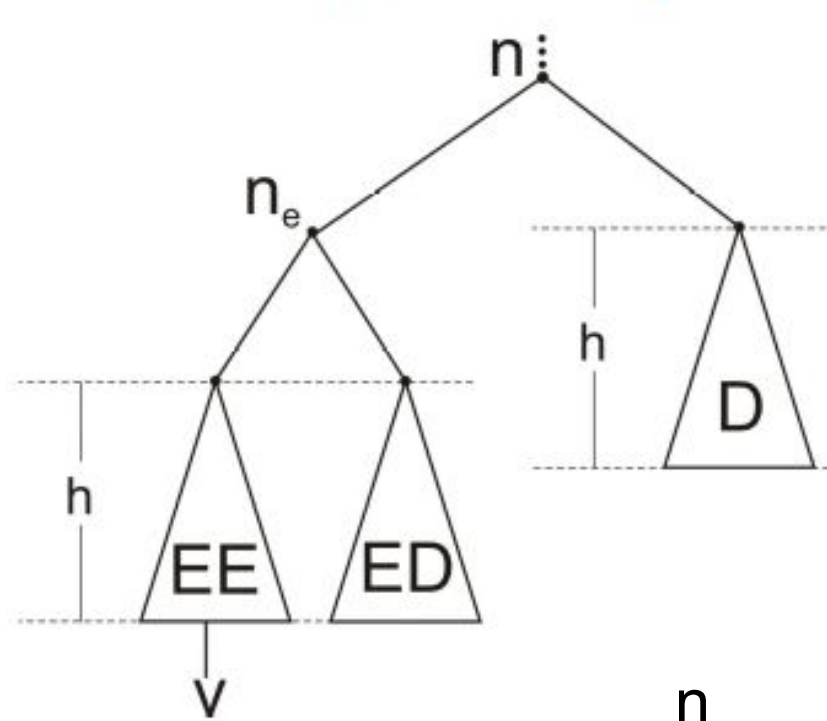
**n é sempre o
nó que está
desbalanceado**



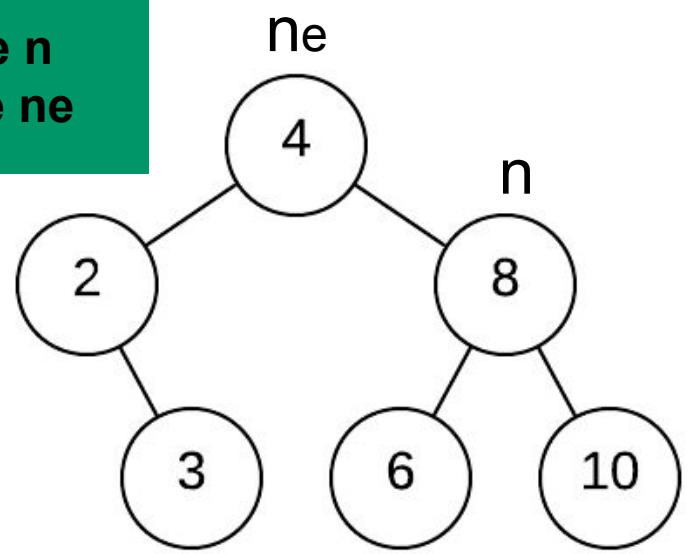
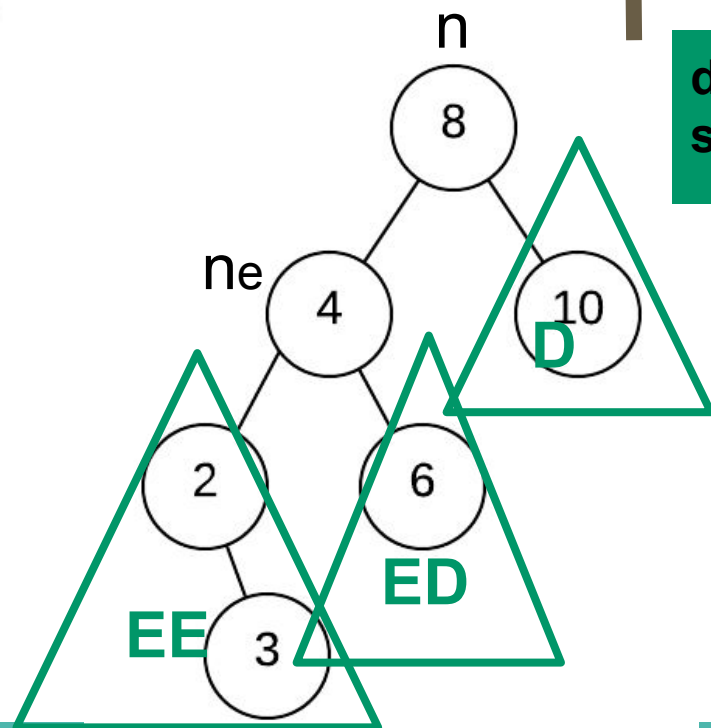


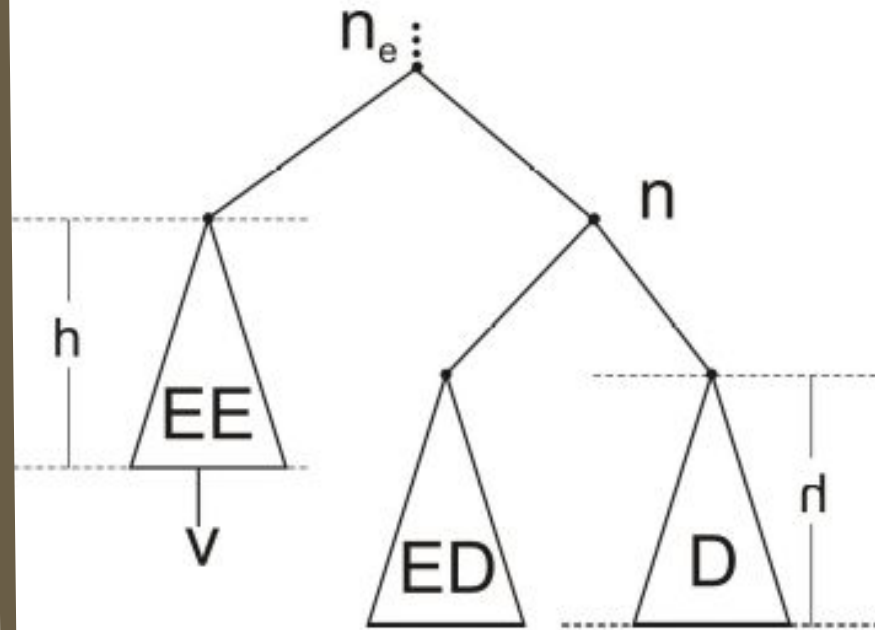
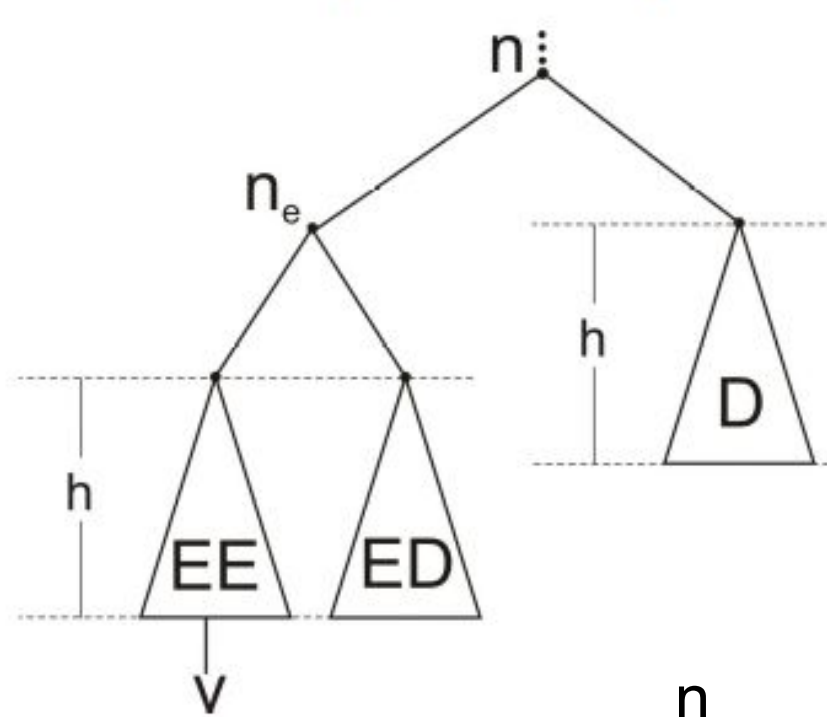
identificando
as árvores



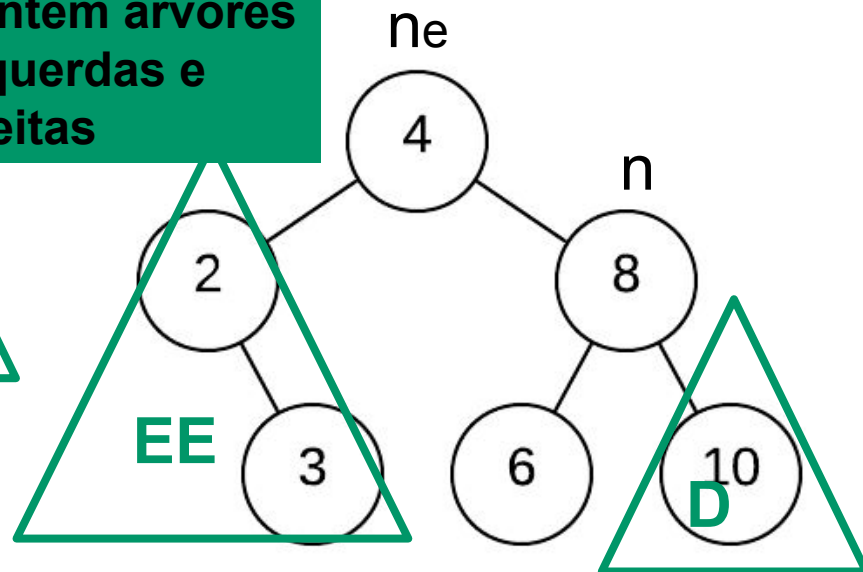
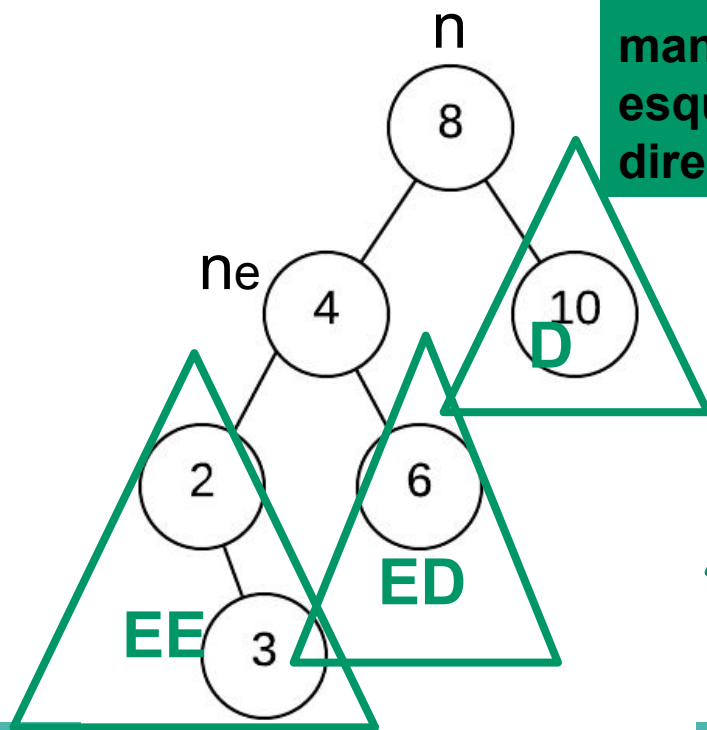


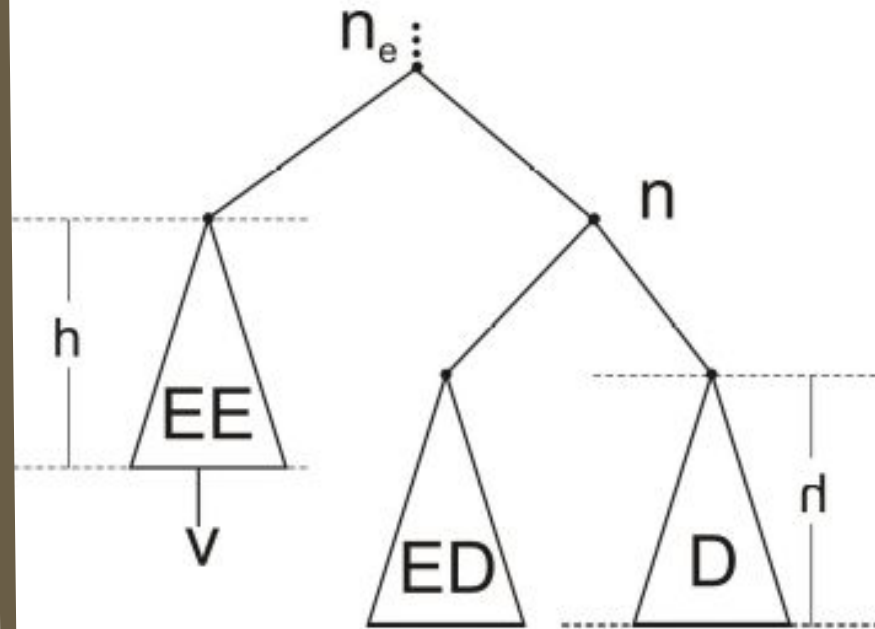
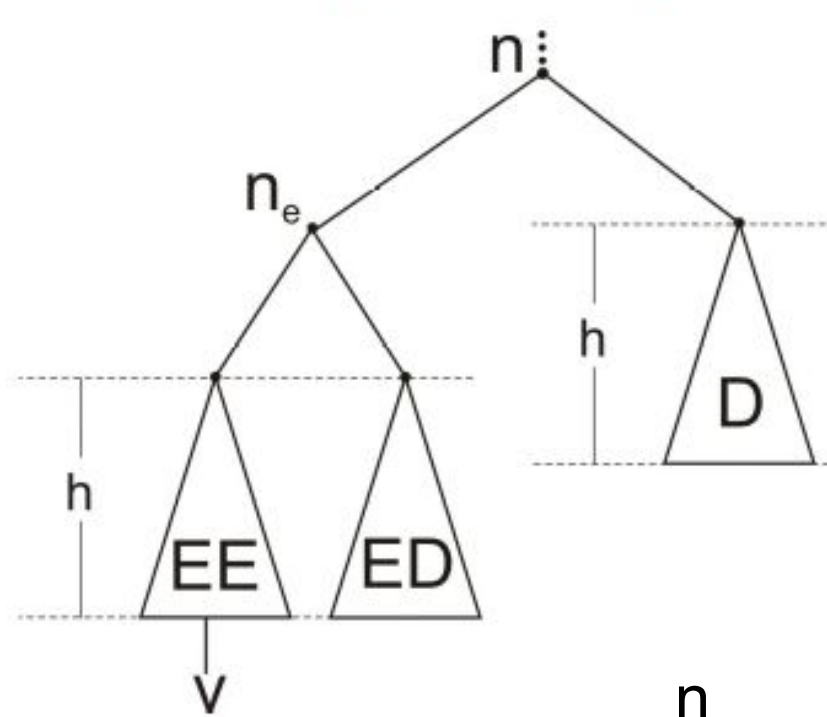
desce n
sobre n_e



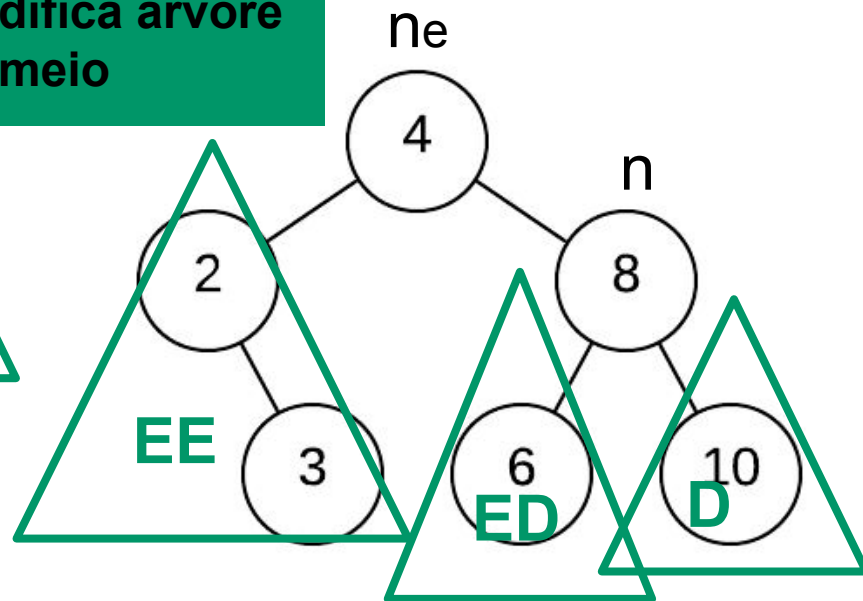
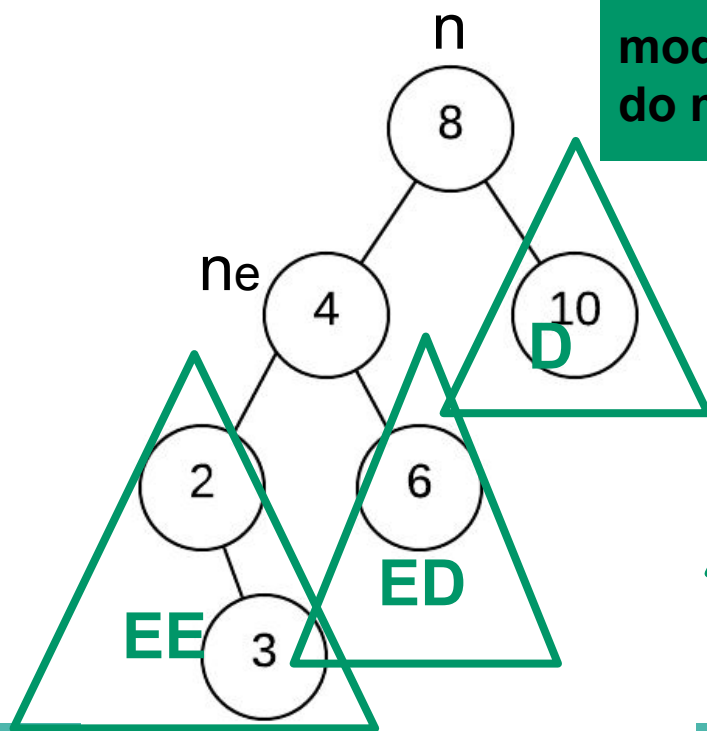


**mantém árvores
esquerdas e
direitas**

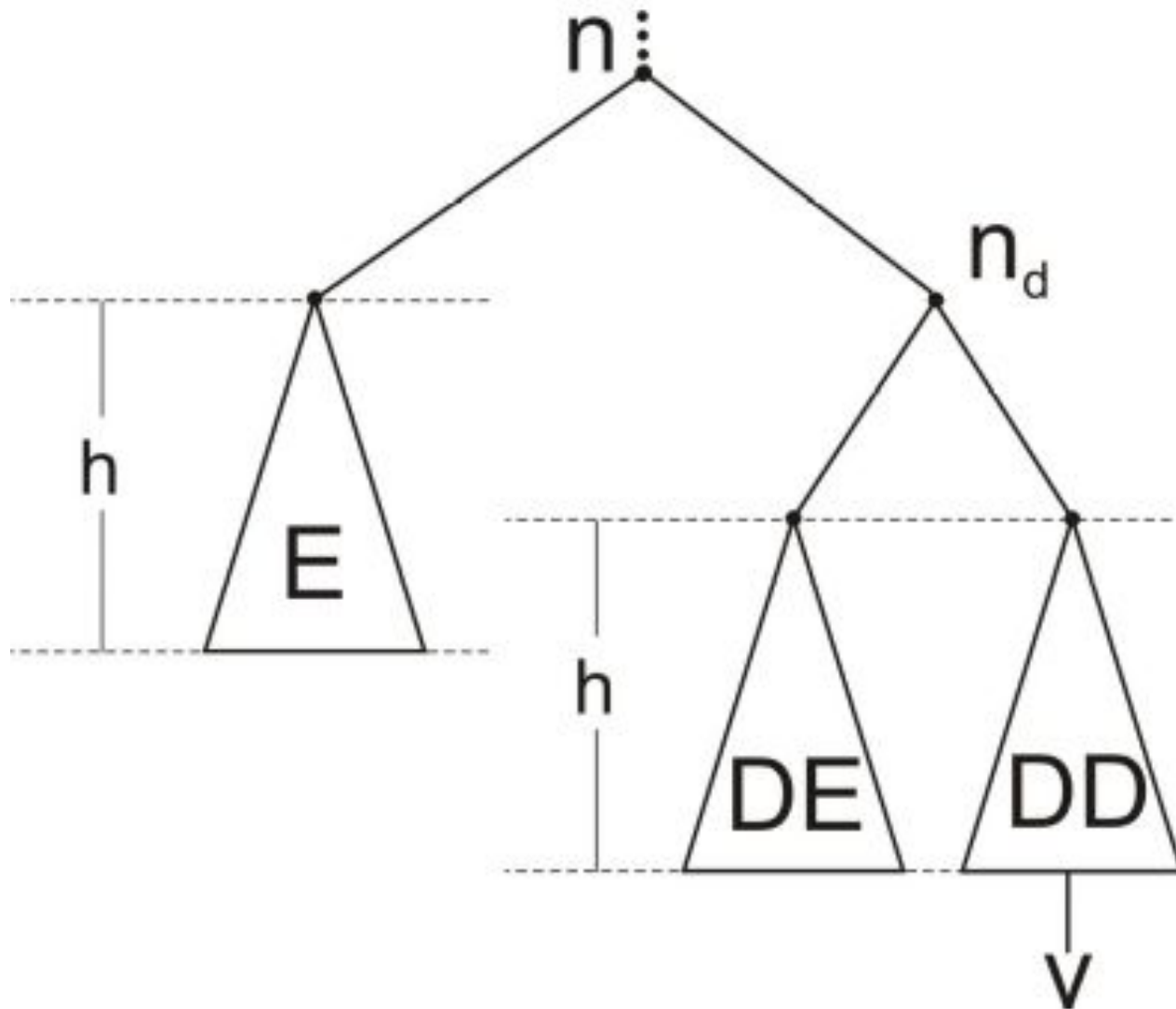




modifica árvore
do meio

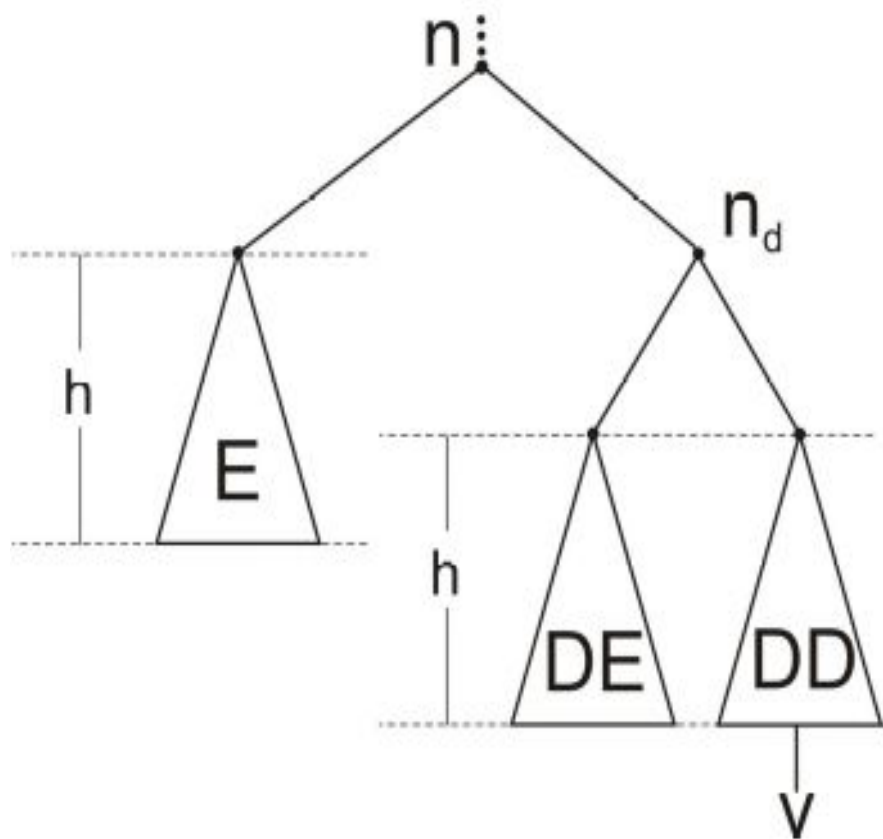


Caso C - DD - simétrico ao Caso A

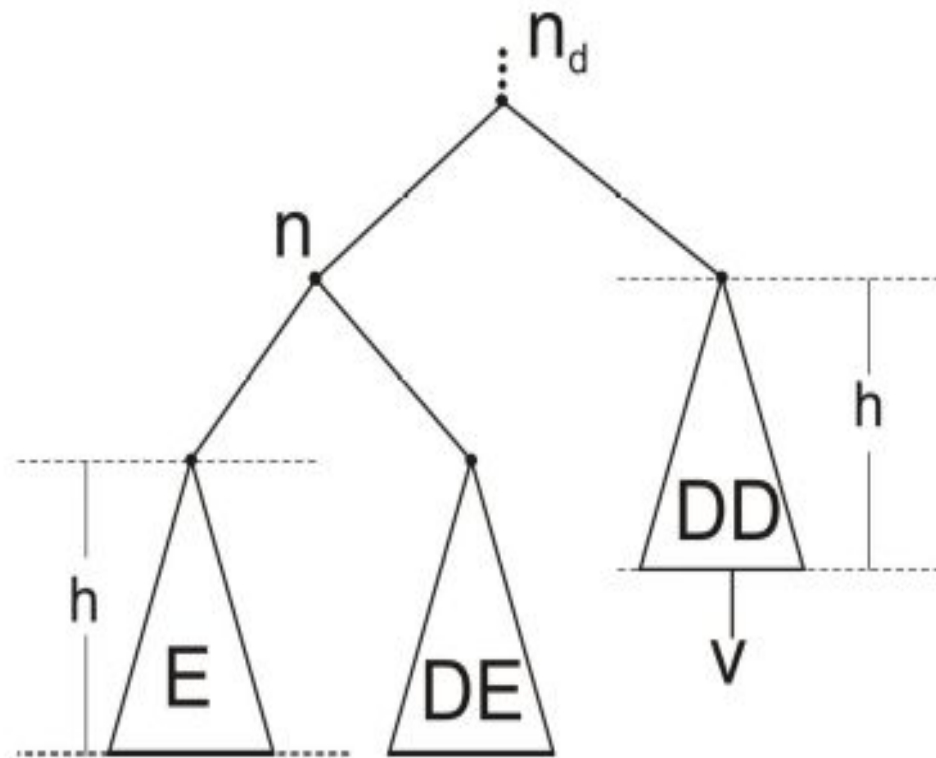


Rotação para esquerda (simples)

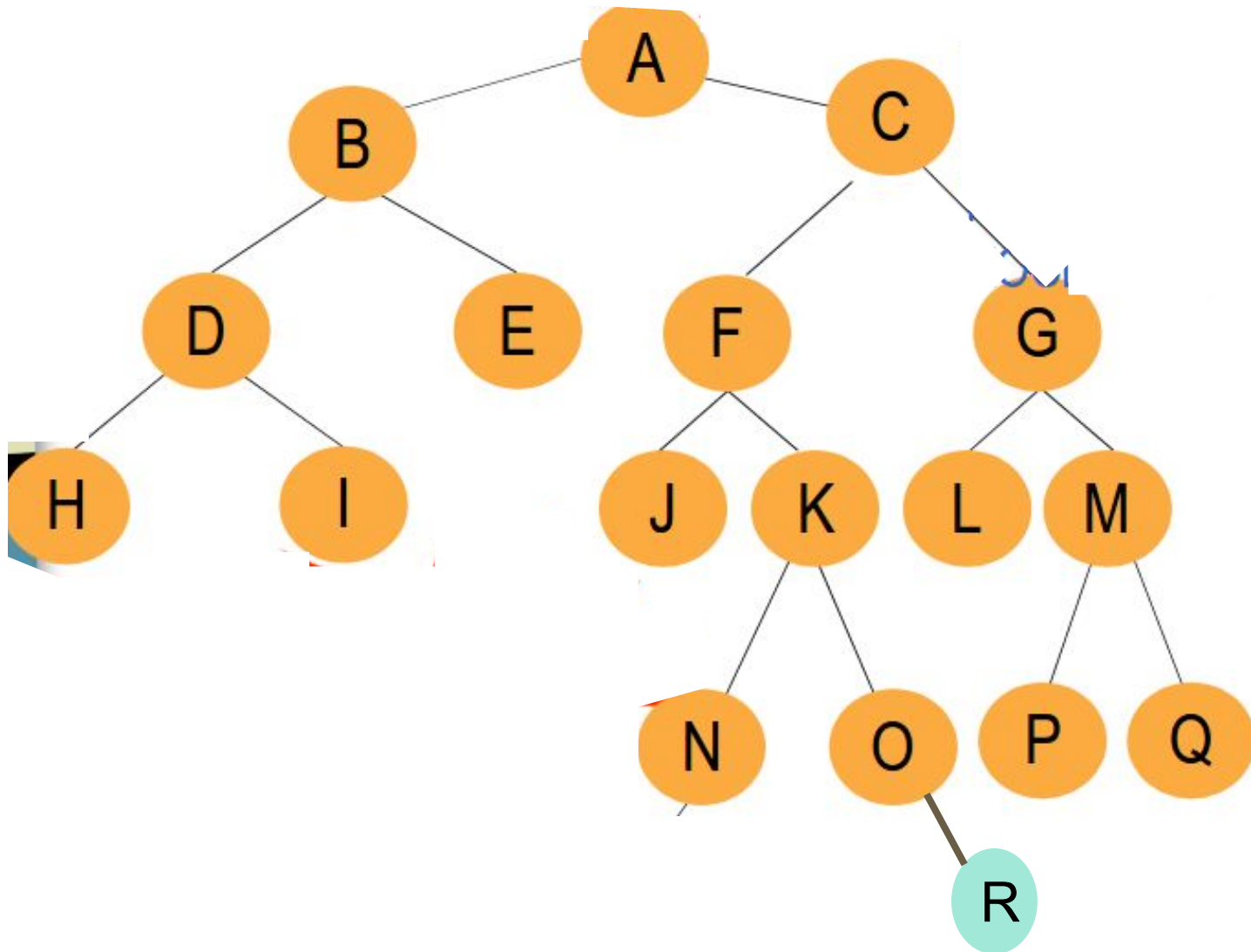
antes



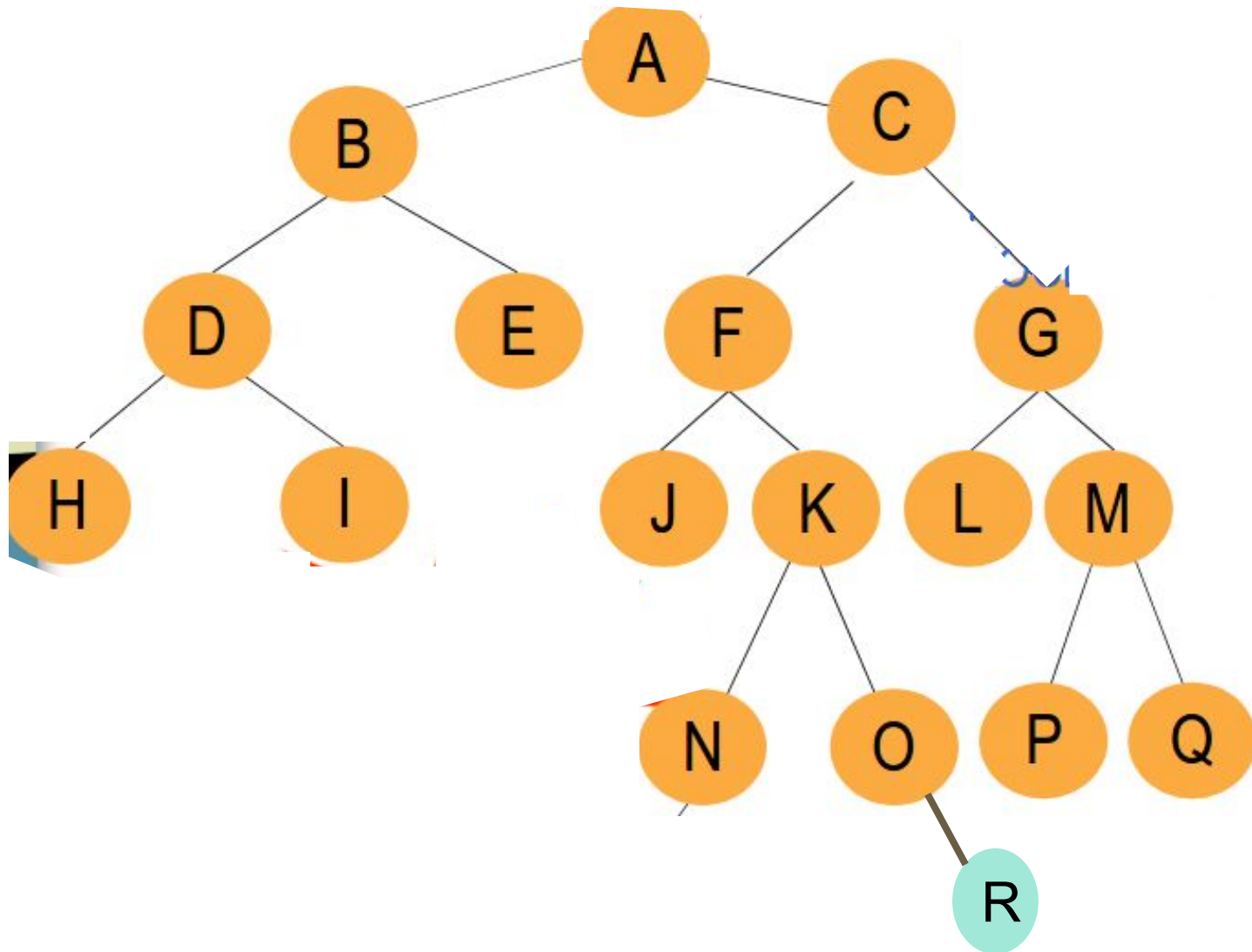
depois



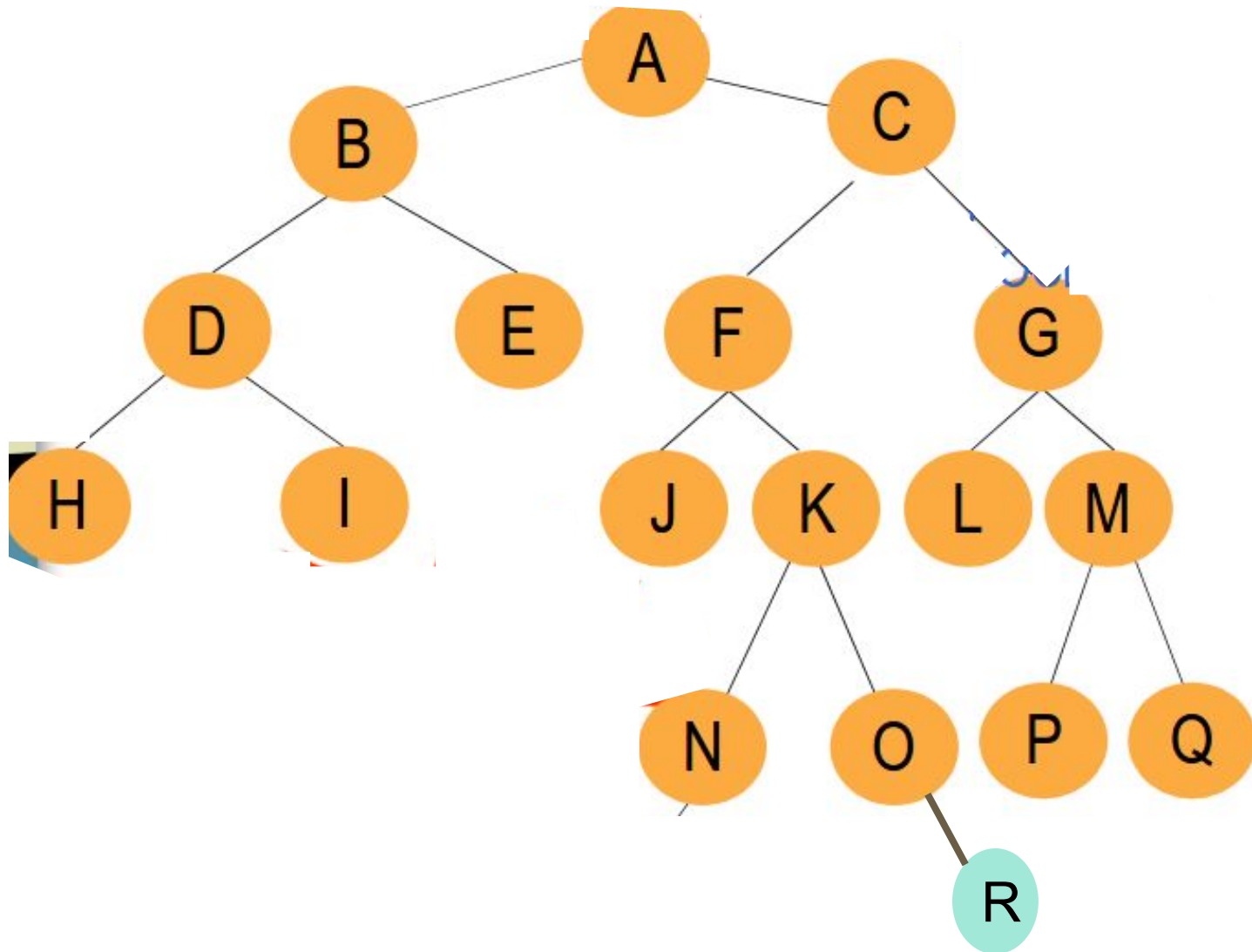
Exemplo - Qual nó está desbalanceado?



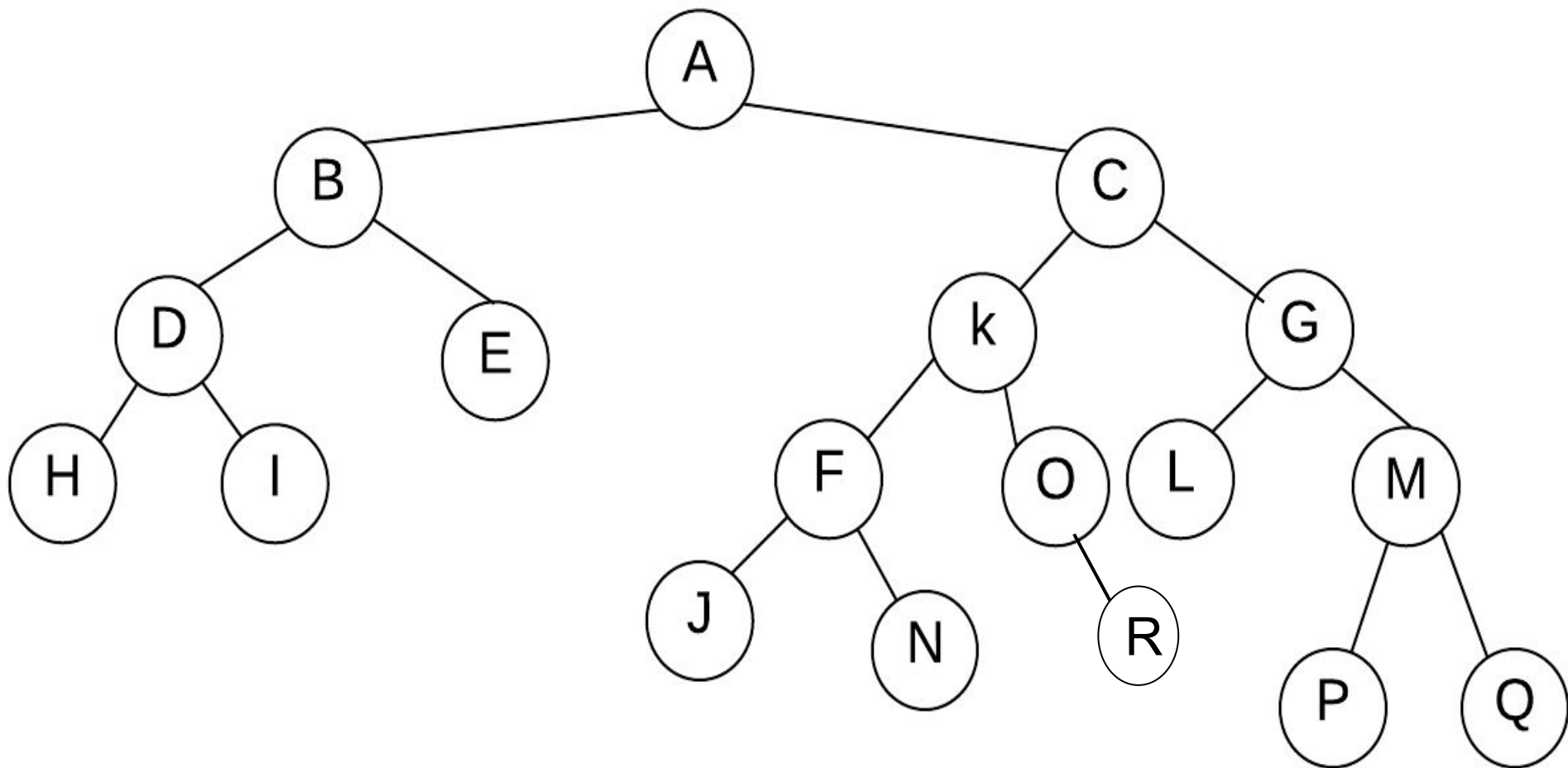
Exemplo - Rotação a direita ou esquerda?



Exemplo - Rotação a esquerda (em F)



Exemplo - Rotação a esquerda



Código - rotação à esquerda

```
void rotacaoEsq(p_no n){  
    p_no nd;  
    nd = n->dir;  
    n->dir = nd->esq;  
    nd->esq = n;  
}
```

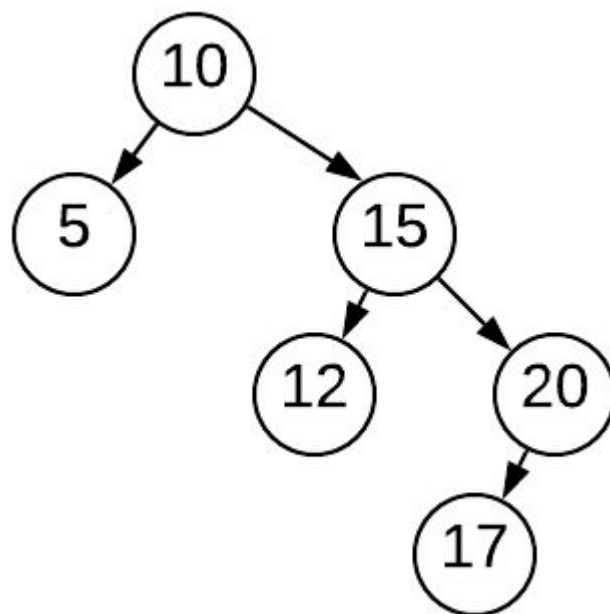
Passo a passo no quadro

Rotação à direita é simétrica!

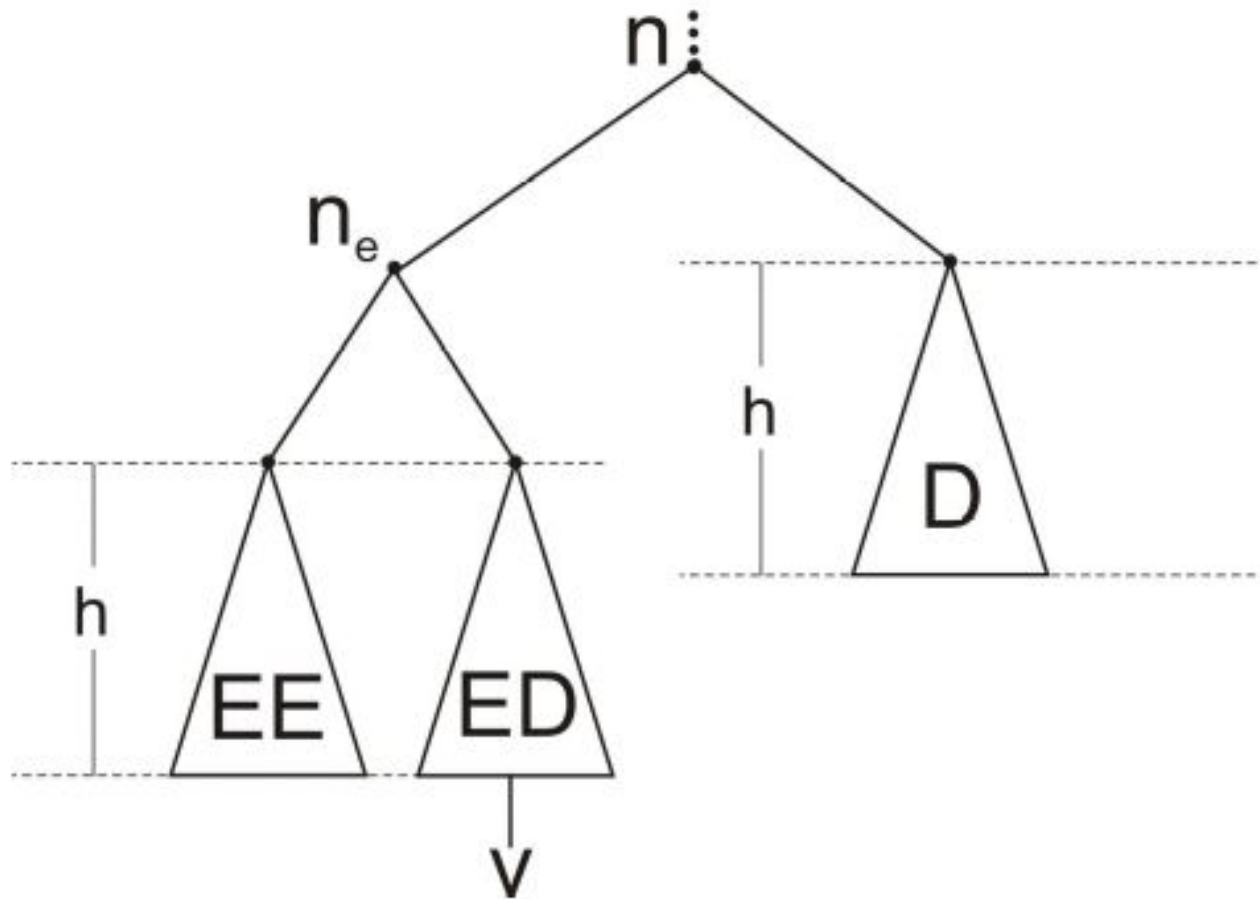
Exercício em sala

Que nó está desbalanceado?

Rotação a esquerda ou direita? Faça!

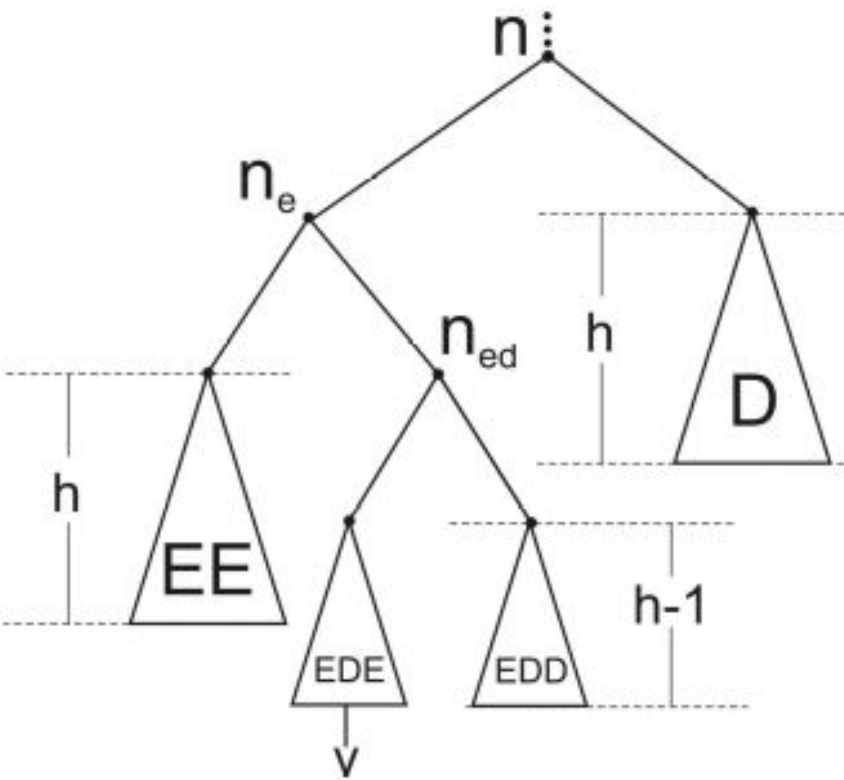


Caso B - ED

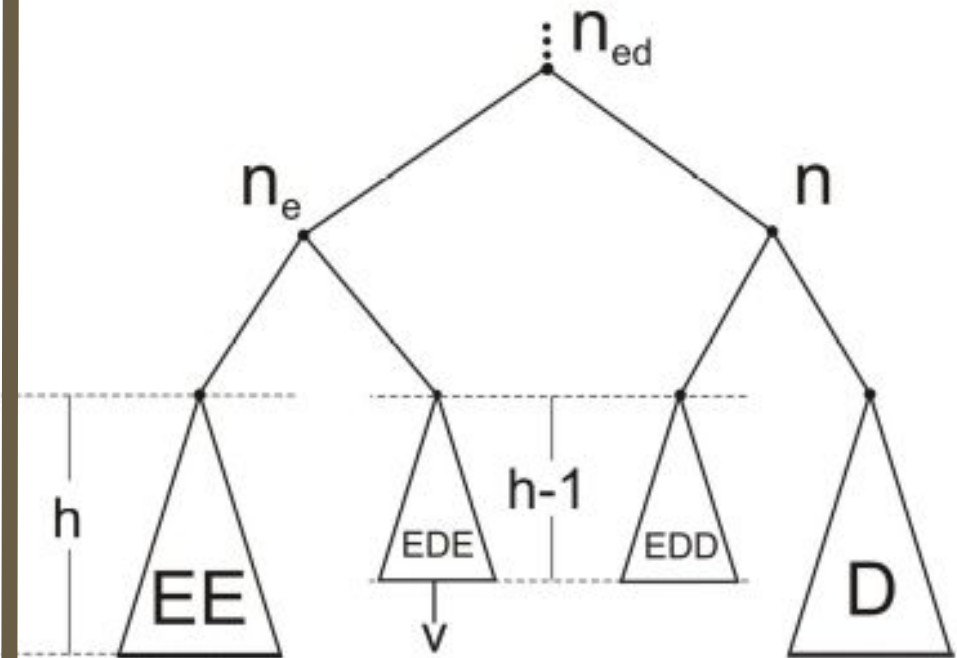


Caso B - Rotação dupla à direita (fórmula direta)

antes



depois



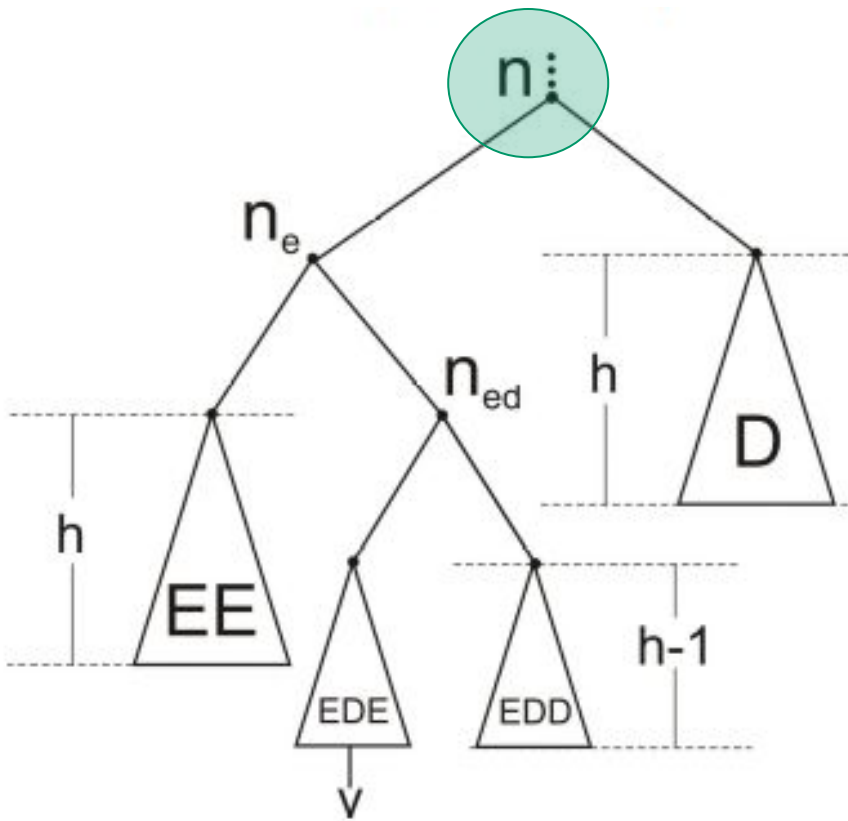
Caso B - Rotação dupla à direita

Passos:

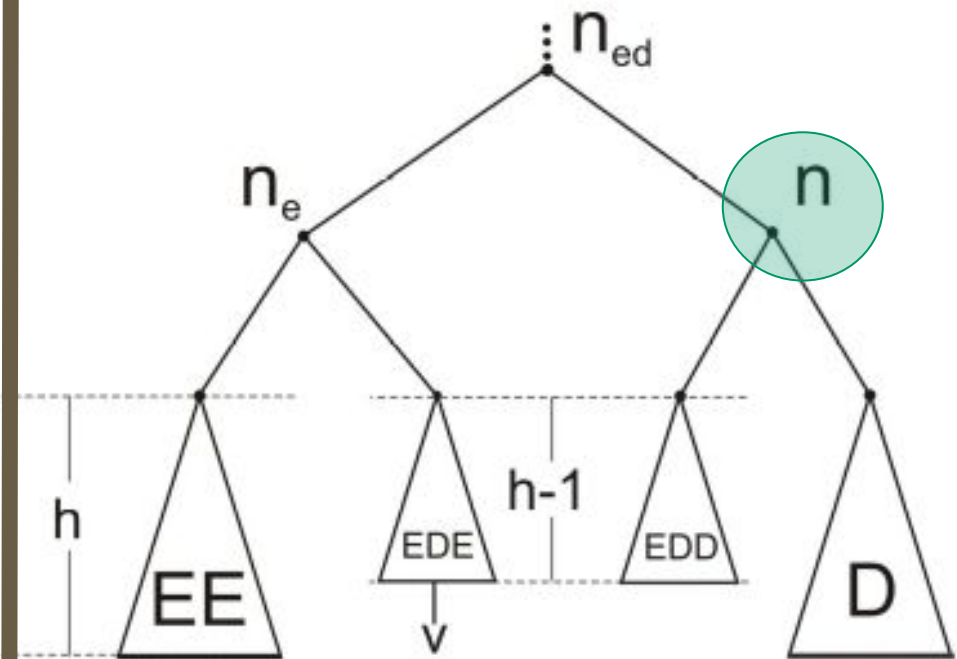
- Modifica nós centrais:
 - Sobe ned
 - n desce para direita

Caso B - Rotação dupla à direita (fórmula direta)

antes

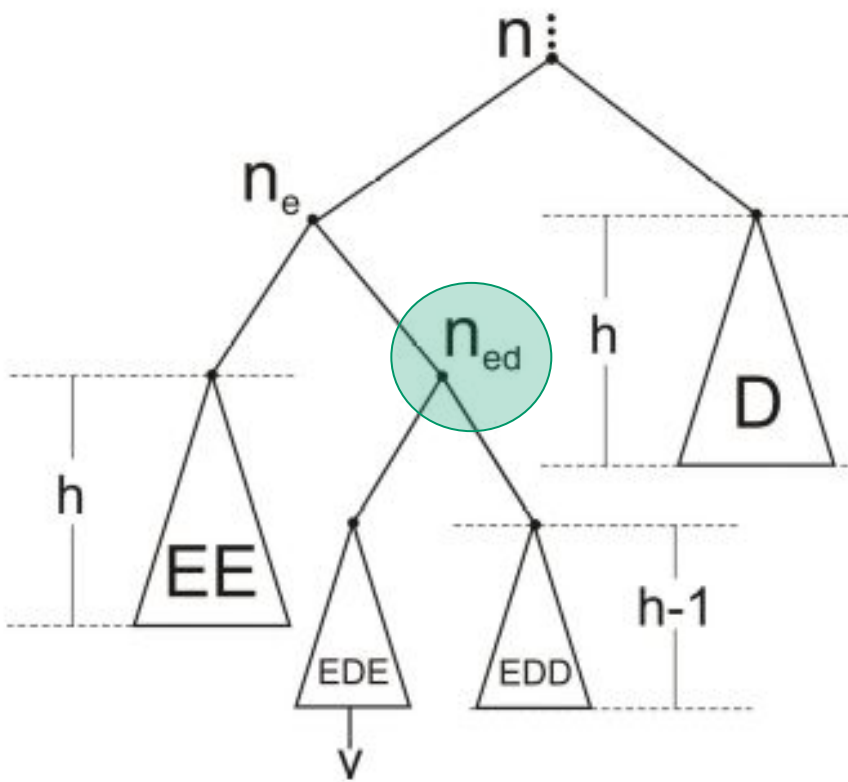


depois

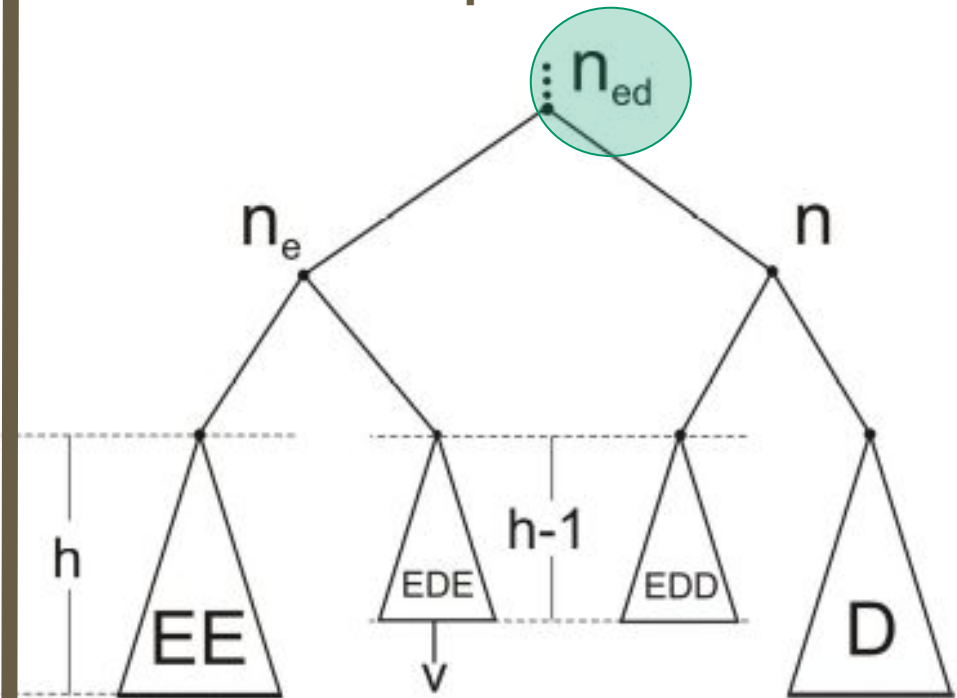


Caso B - Rotação dupla à direita (fórmula direta)

antes



depois



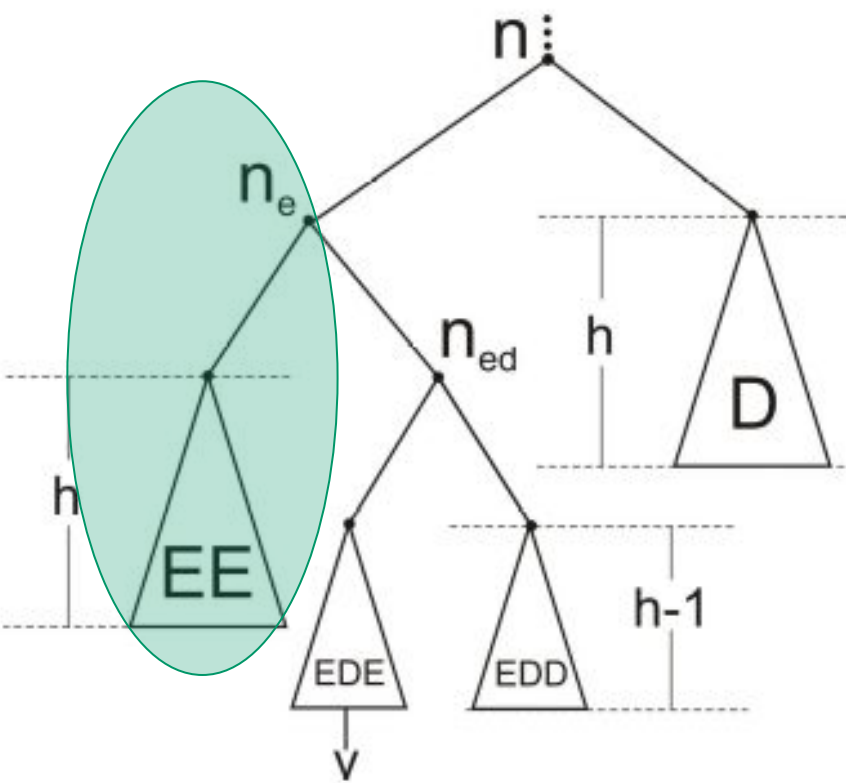
Caso B - Rotação dupla à direita

Passos:

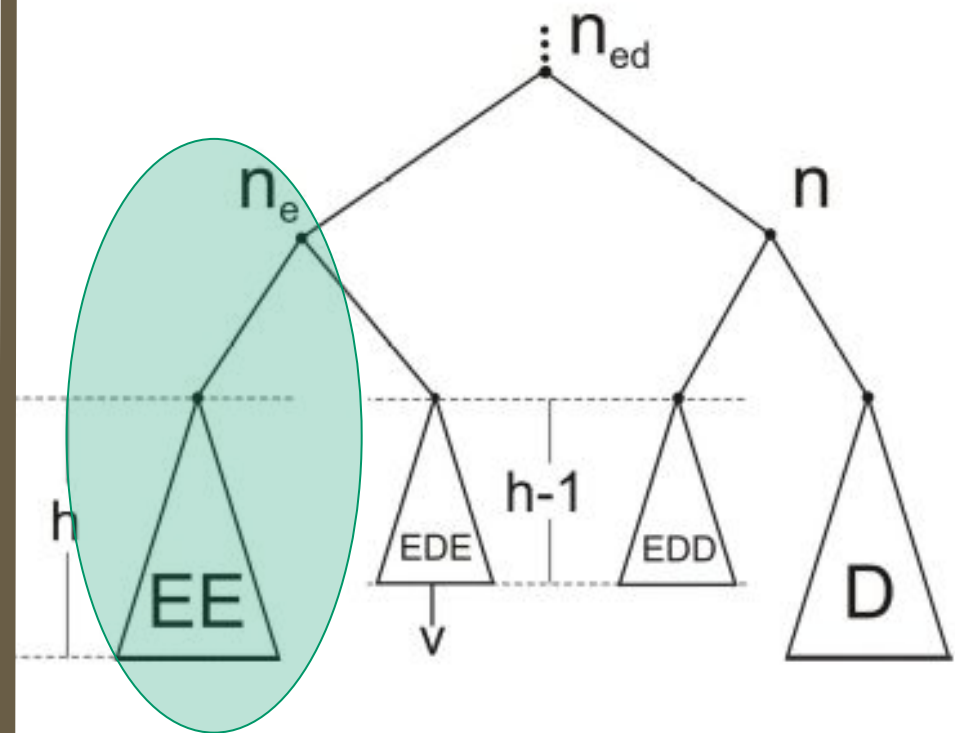
- Modifica nós centrais:
 - Sobe ned
 - n desce para direita
- Mantém árvores das extremidades:
 - esquerdas e direitas (EE e D)

Caso B - Rotação dupla à direita (fórmula direta)

antes

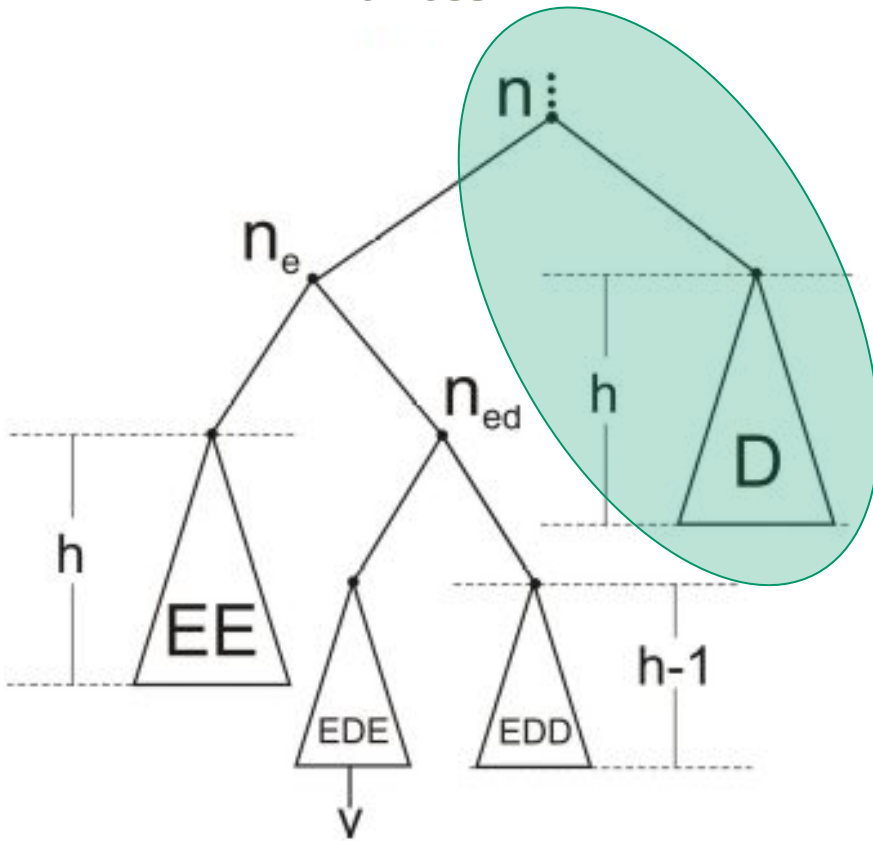


depois

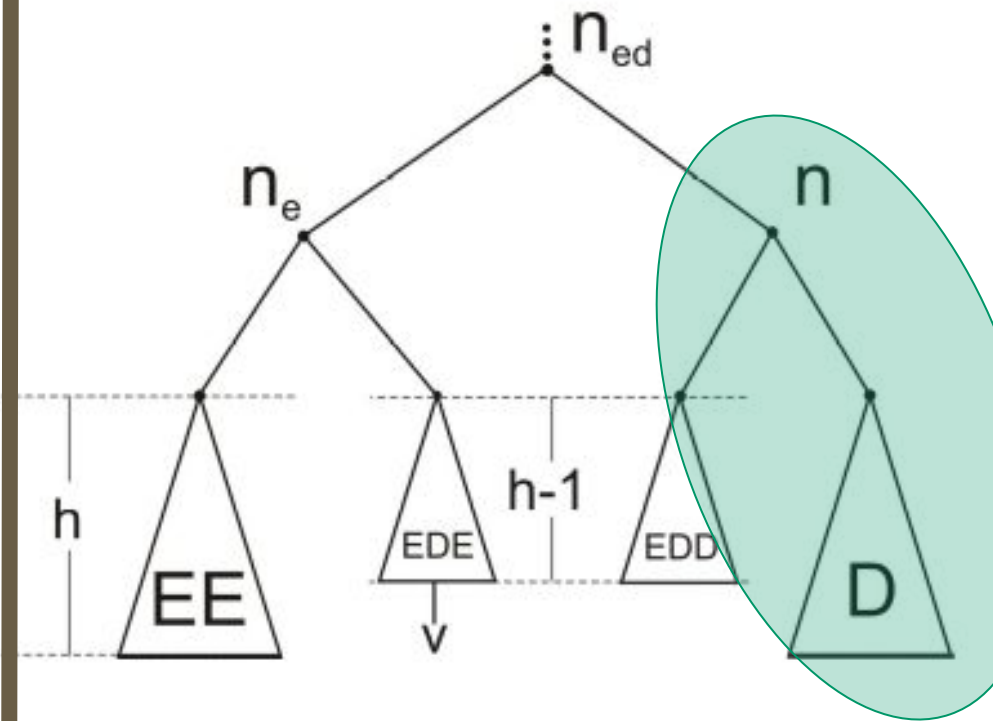


Caso B - Rotação dupla à direita (fórmula direta)

antes



depois



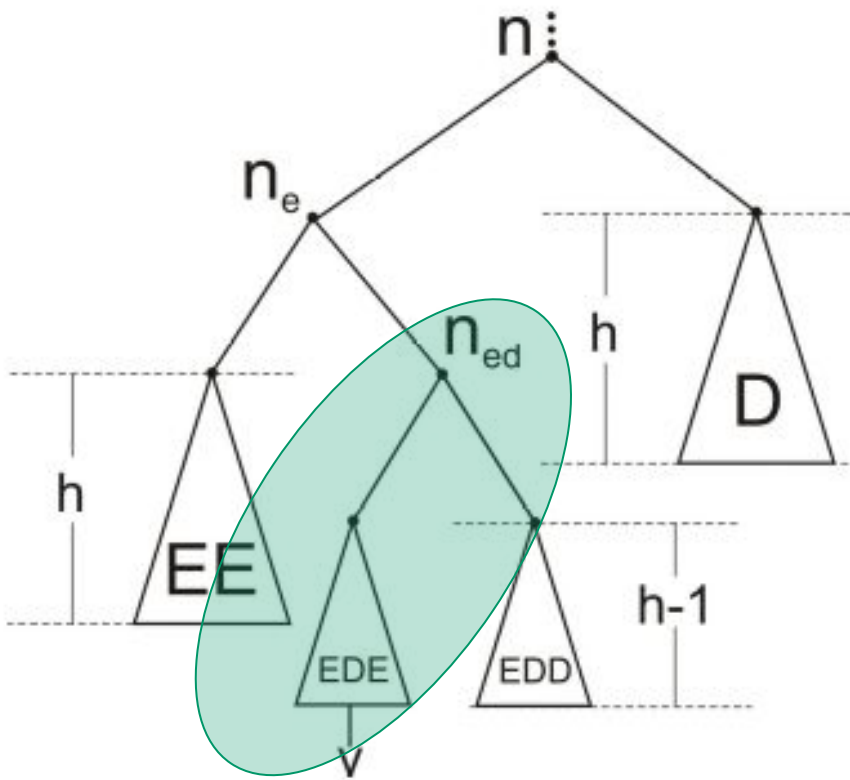
Caso B - Rotação dupla à direita (fórmula direta)

Passos:

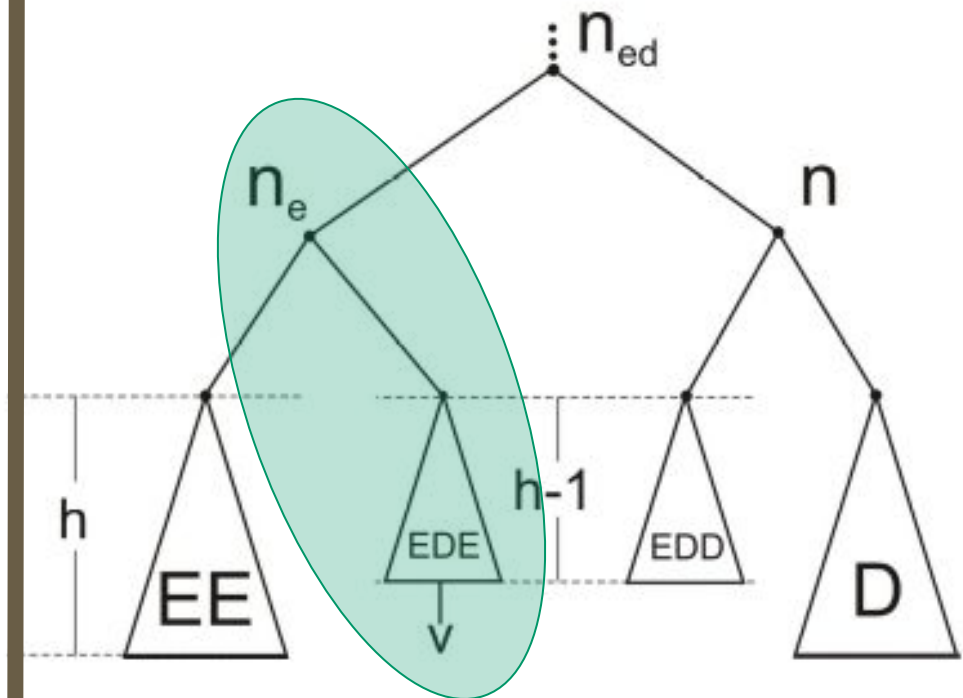
- Modifica nós centrais:
 - Sobe ned
 - n desce para direita
- Mantém árvores das extremidades:
 - esquerdas e direitas (EE e D)
- Distribui árvore do meio (EDE e EDD)
 - árvores do meio eram filhas de ned, porém este nó não pode mais ter filhos, já tem 2 (n e ne).
 - árvore EDE será filha direita de ne
 - árvore EDD será filha esquerda de n

Caso B - Rotação dupla à direita (fórmula direta)

antes

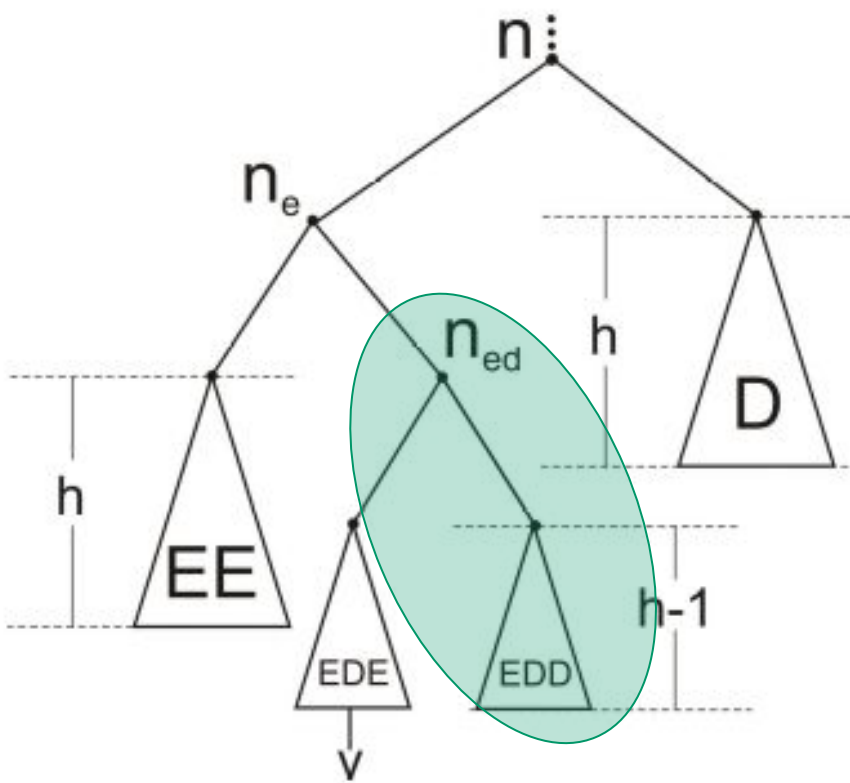


depois

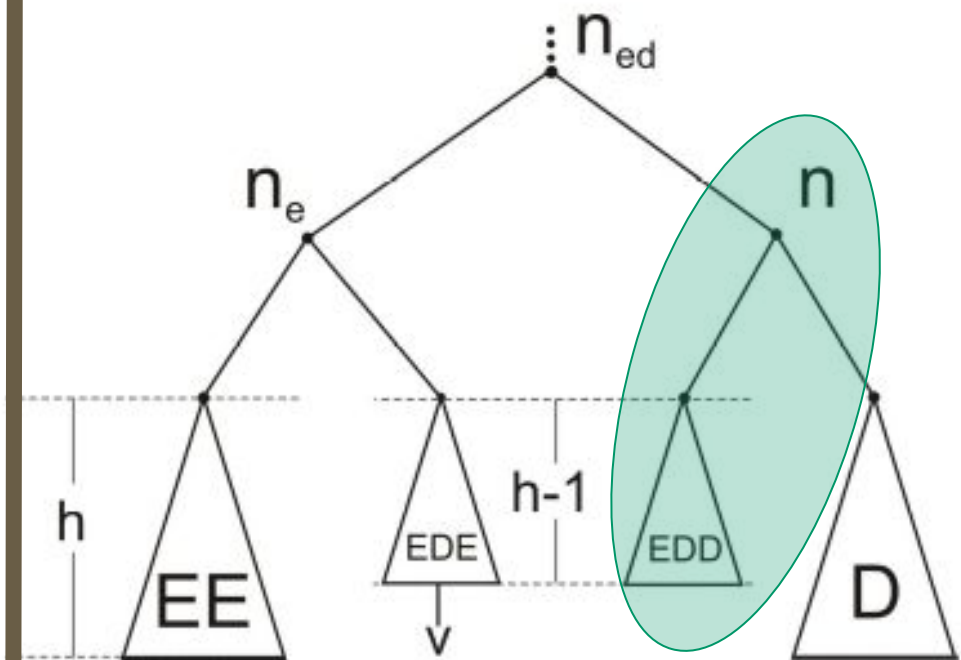


Caso B - Rotação dupla à direita (fórmula direta)

antes



depois



Caso B - Rotação dupla à direita (fórmula direta)

Pode-se decorar a **fórmula direta** de se fazer a rotação dupla.

EU NÃO RECOMENDO!!!

MUITA COISA PARA DECORAR

Caso B - Rotação dupla à direita (2 rotações)

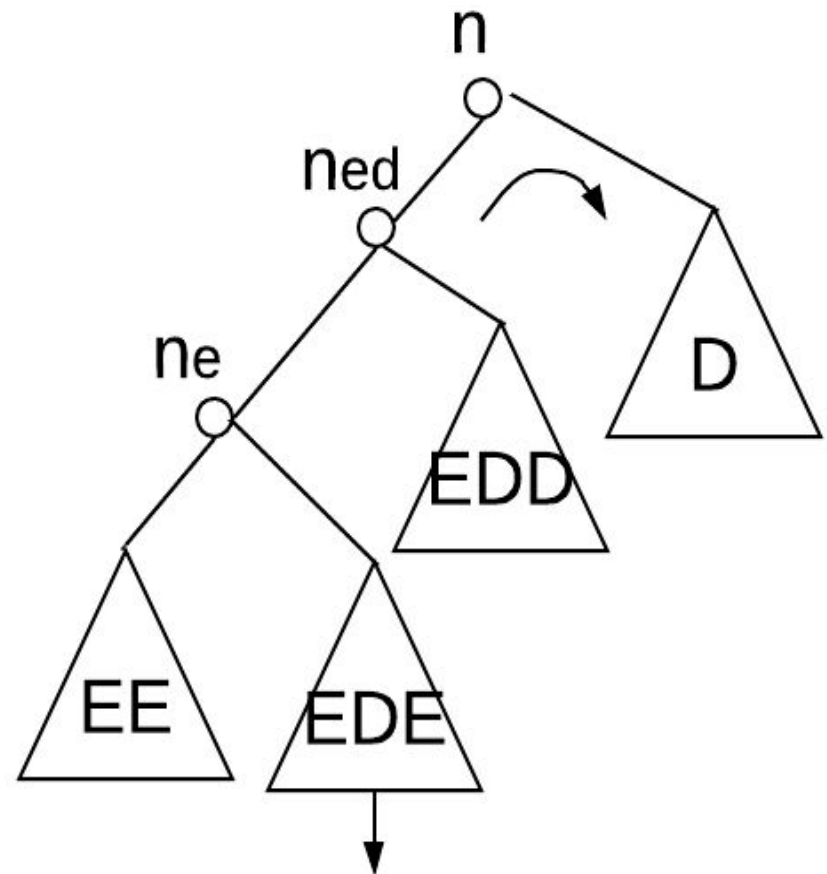
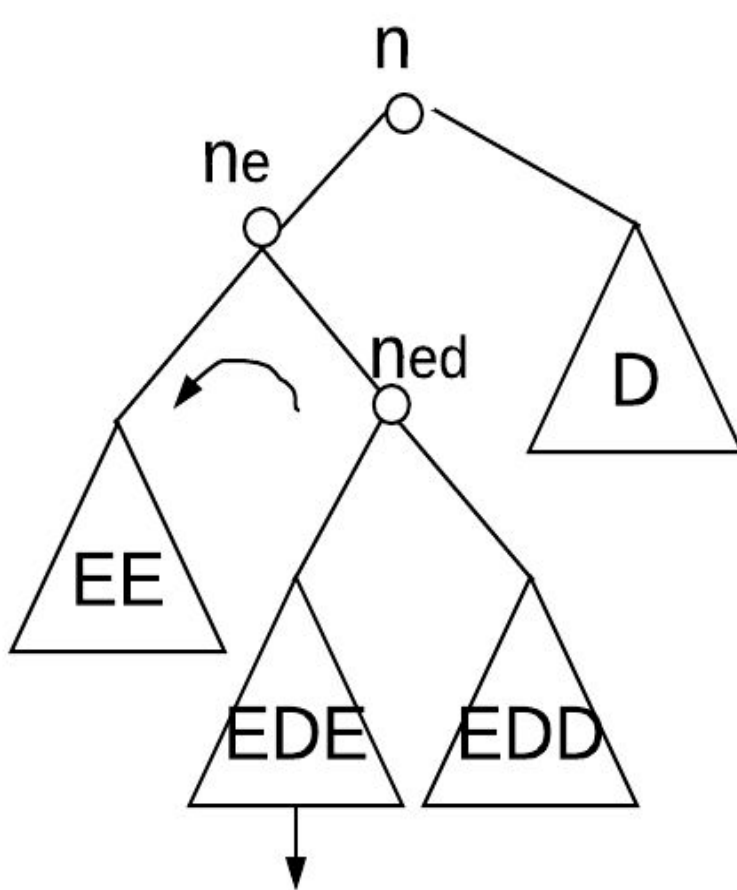
Pode-se decorar a **fórmula direta** de se fazer a rotação dupla.

Ou (**MELHOR SOLUÇÃO**) pode-se desmembrar nas **duas rotações simples** já conhecidas:

- rotação para esquerda em Ne
- rotação para direita em N

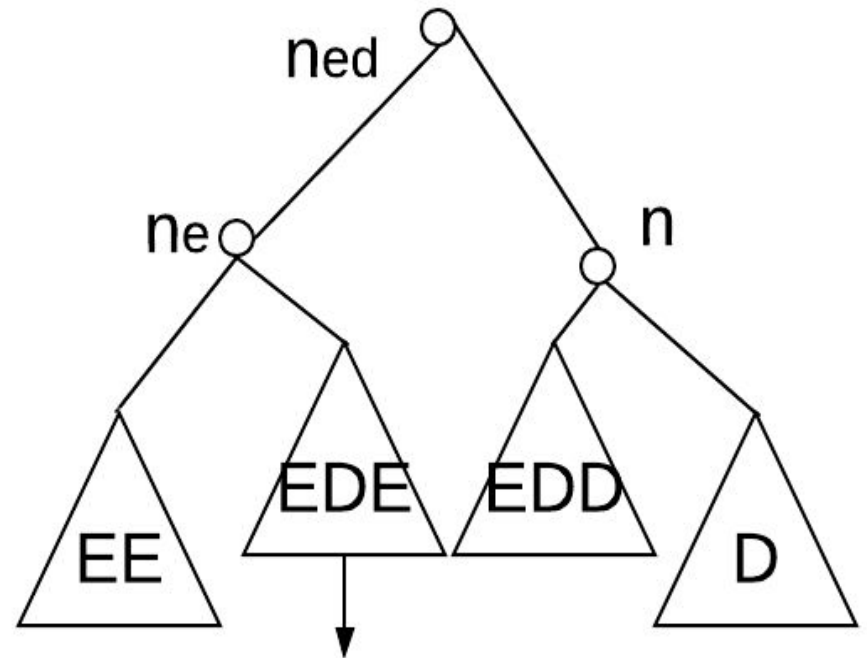
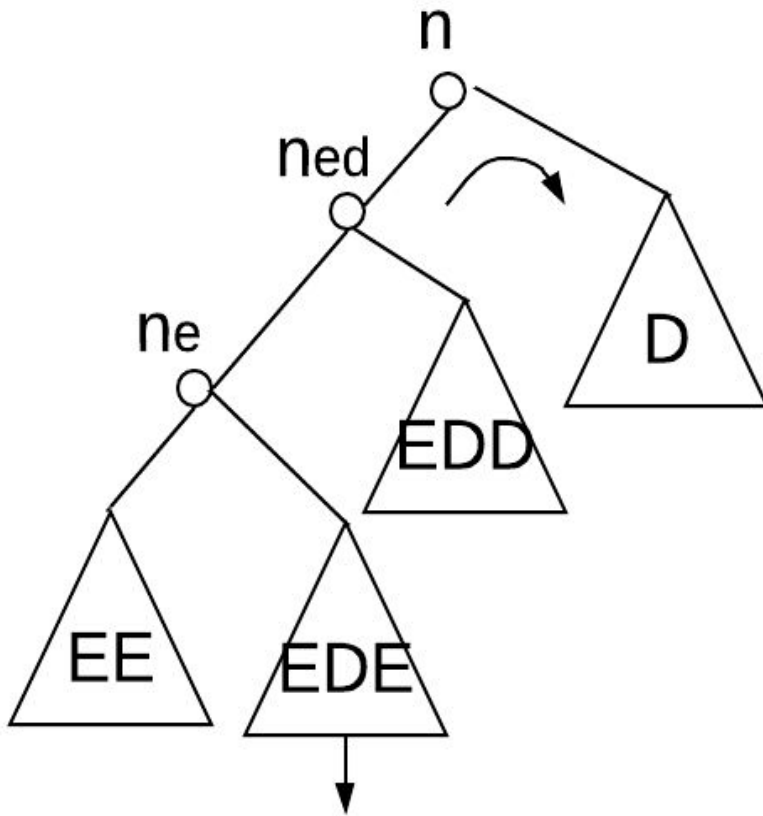
Caso B - Rotação dupla à direita (2 rotações)

- Equivalente à 2 rotações simples:
- a primeira para esquerda em **ne**

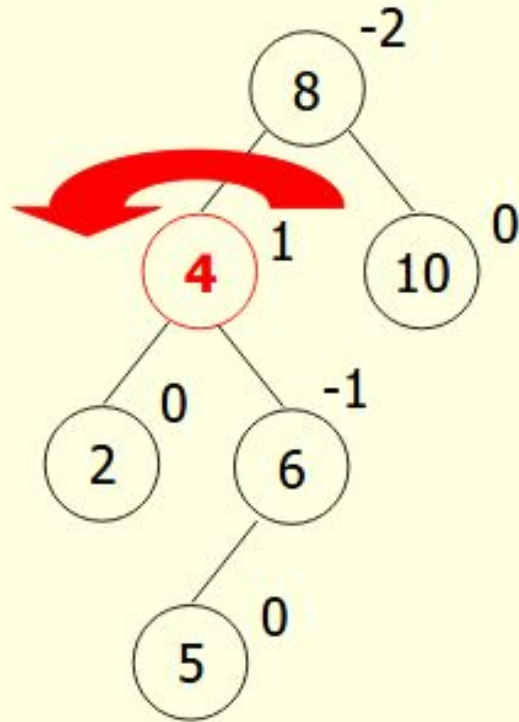


Caso B - Rotação dupla à direita (2 rotações)

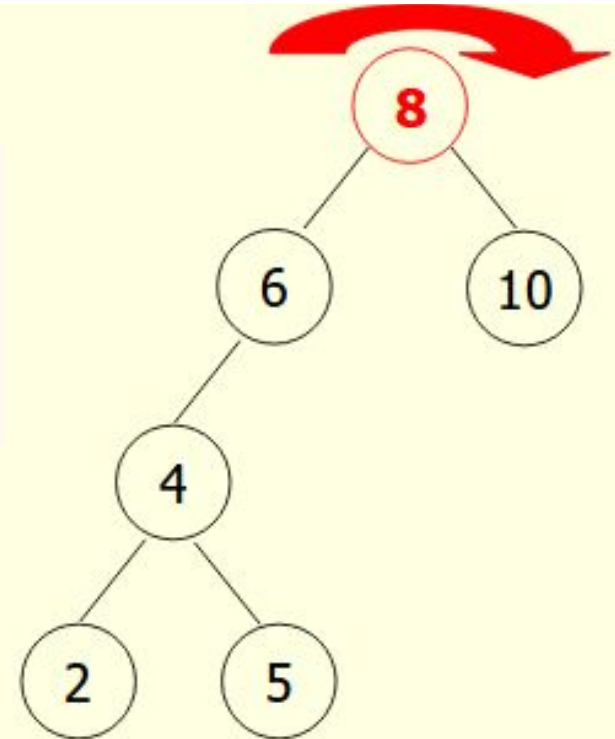
- e a segunda para direita em **n**



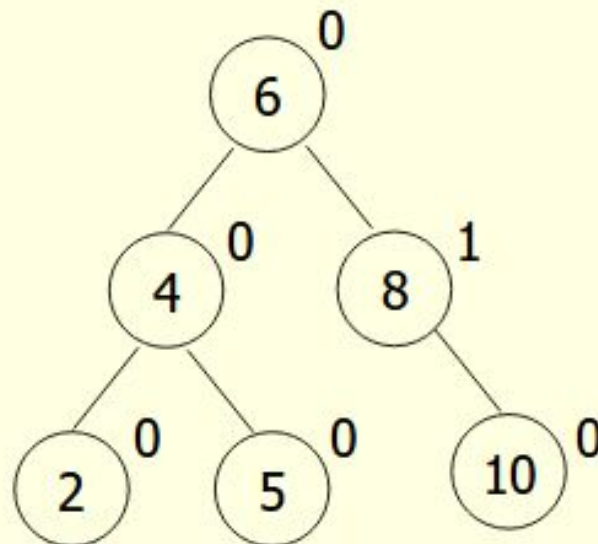
Caso B - Rotação dupla à direita (2 rotações)



Rotação em Ne para esquerda
→



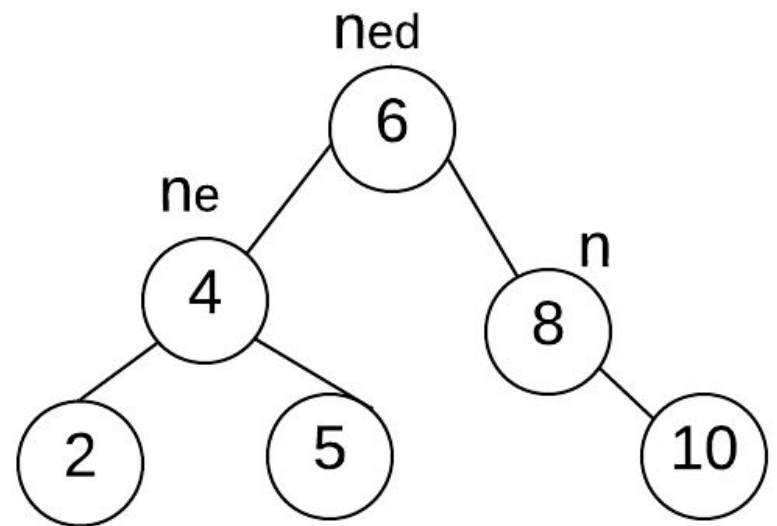
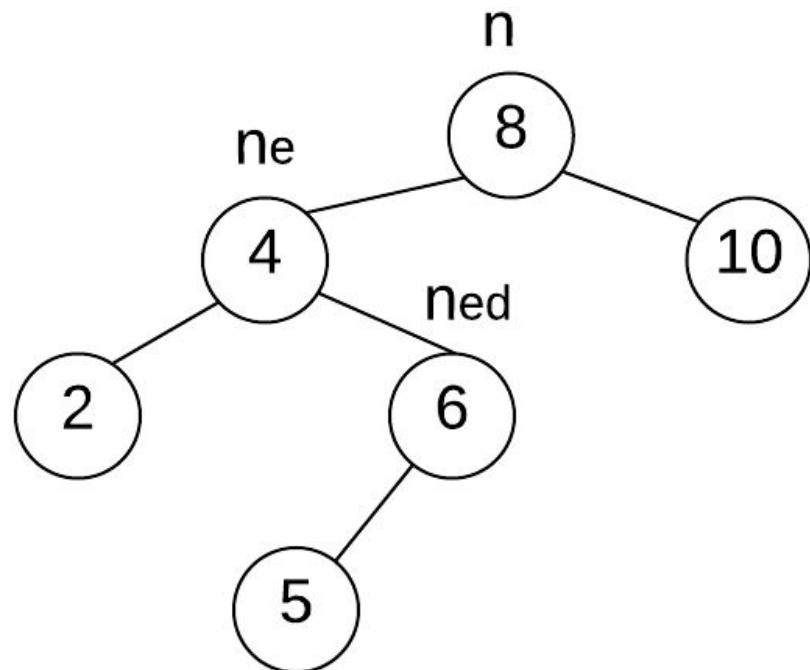
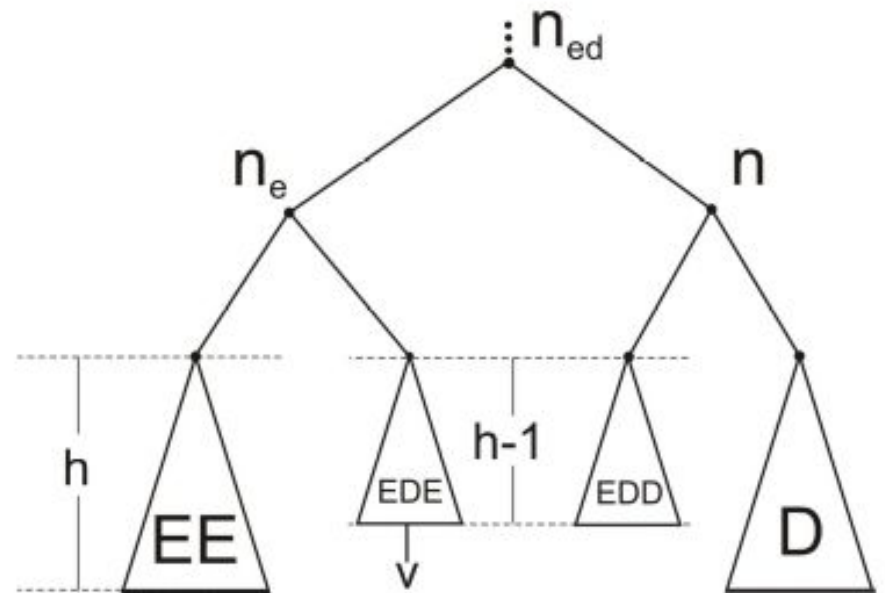
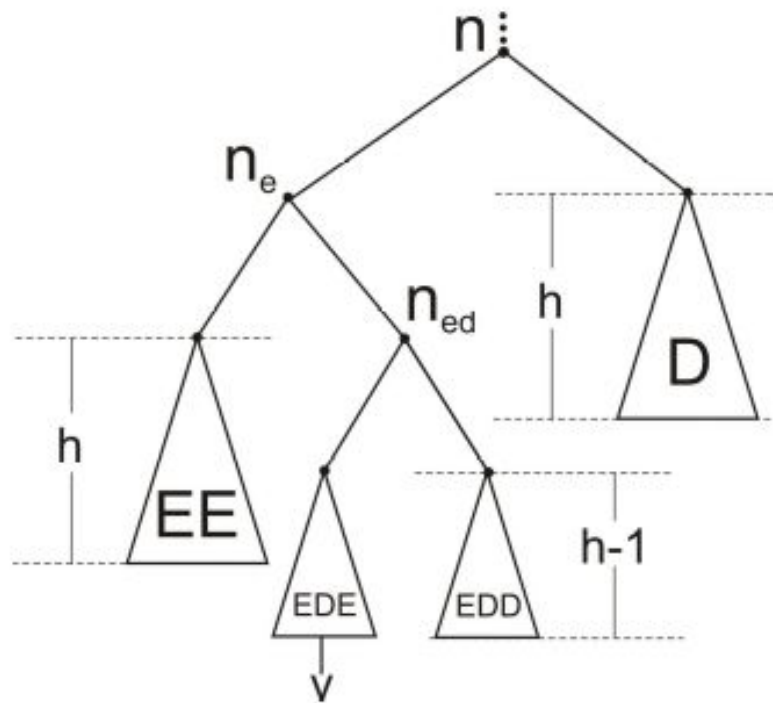
Rotação em N para direita
→

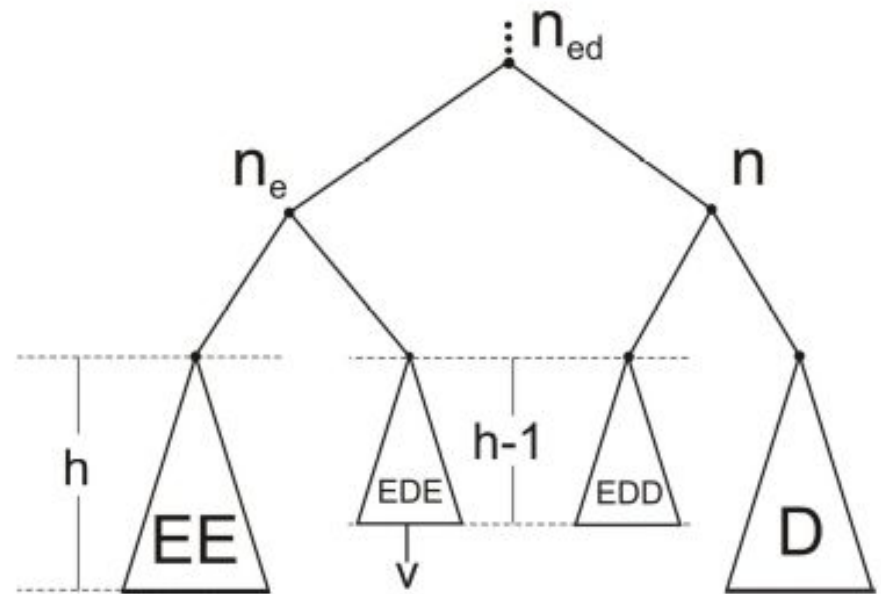
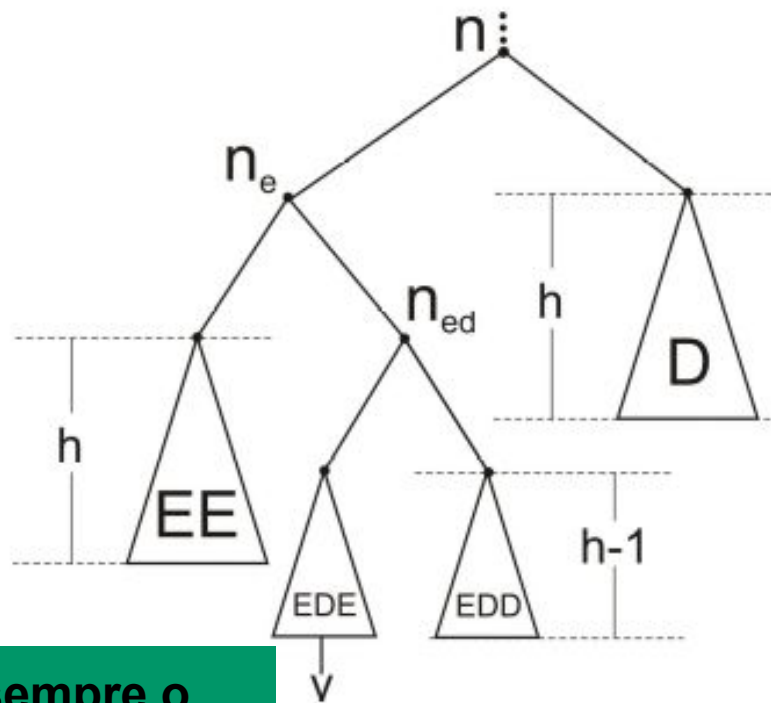


Árvore balanceada!!!

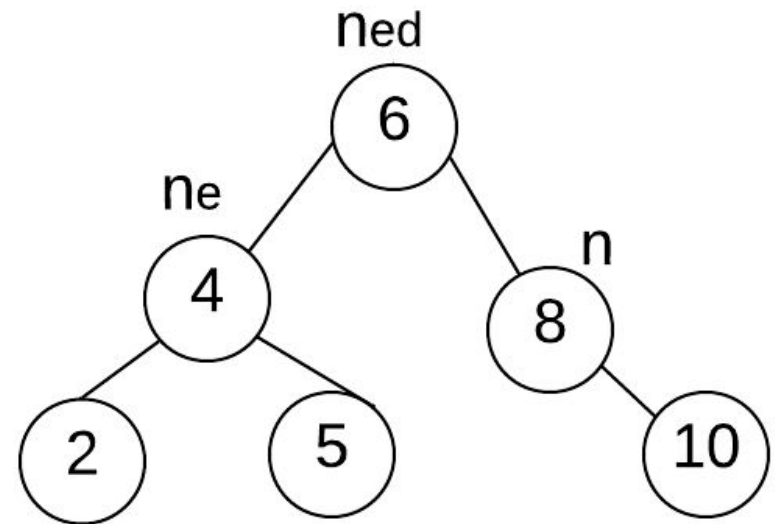
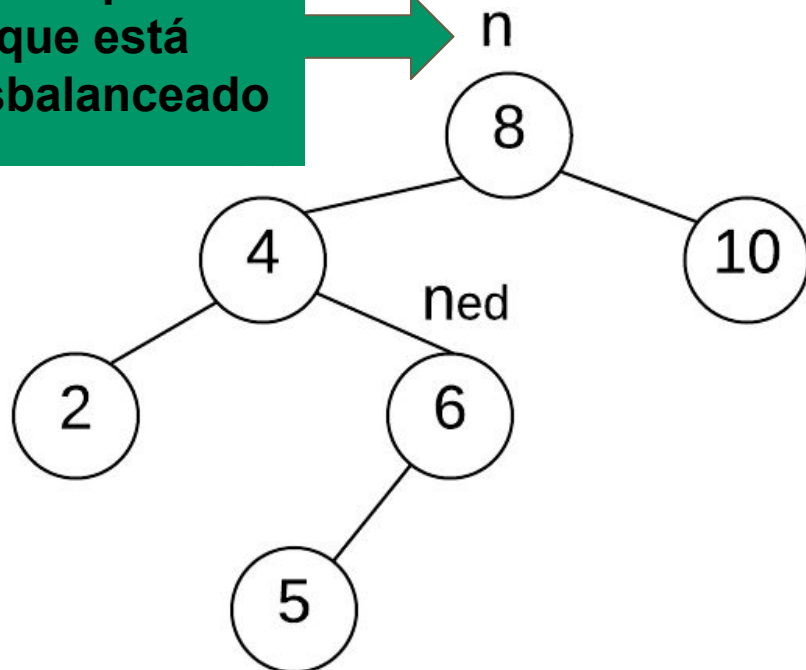
Exemplo com fórmula direta

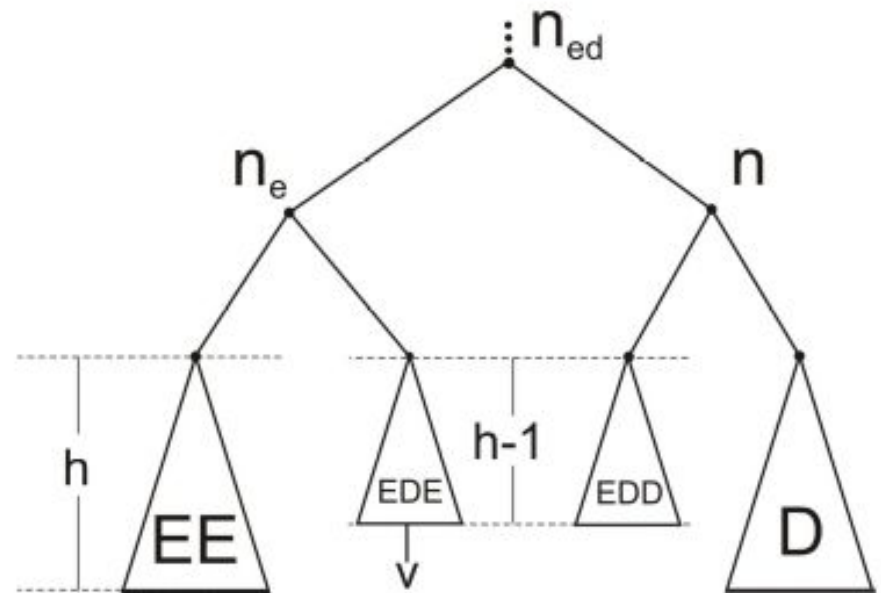
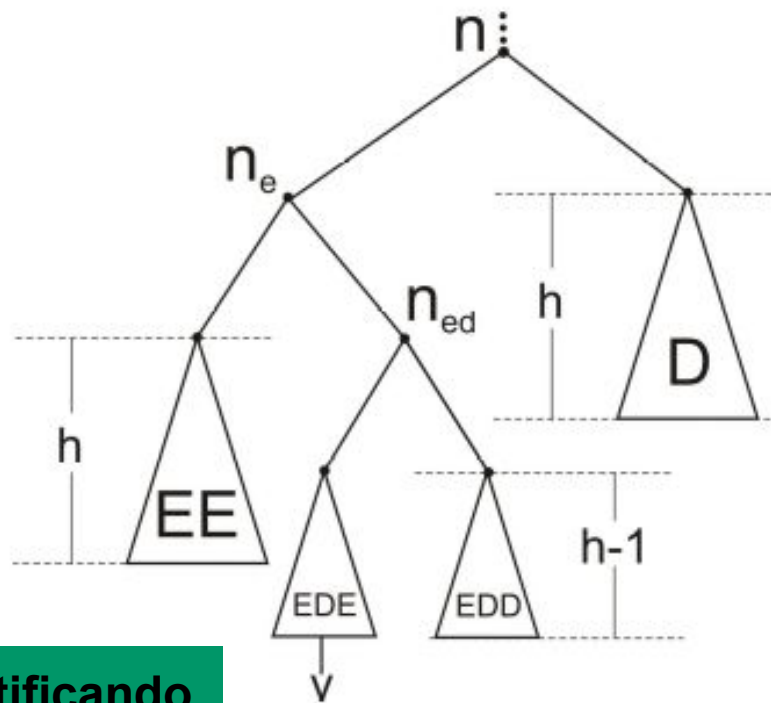
-



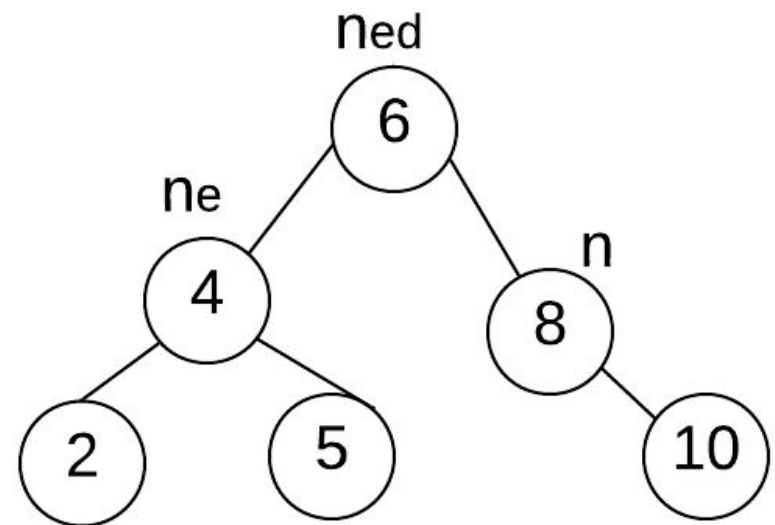
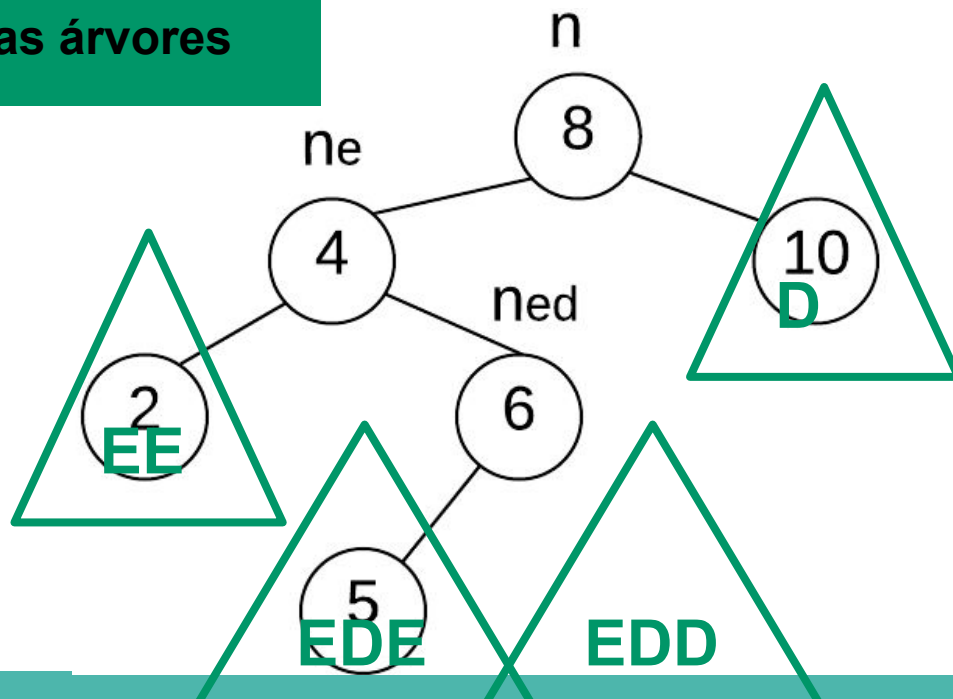


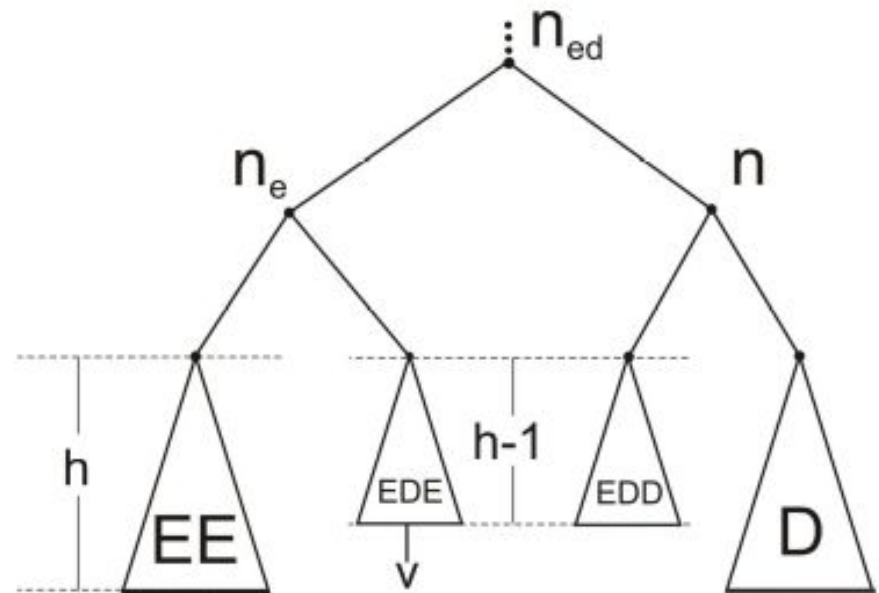
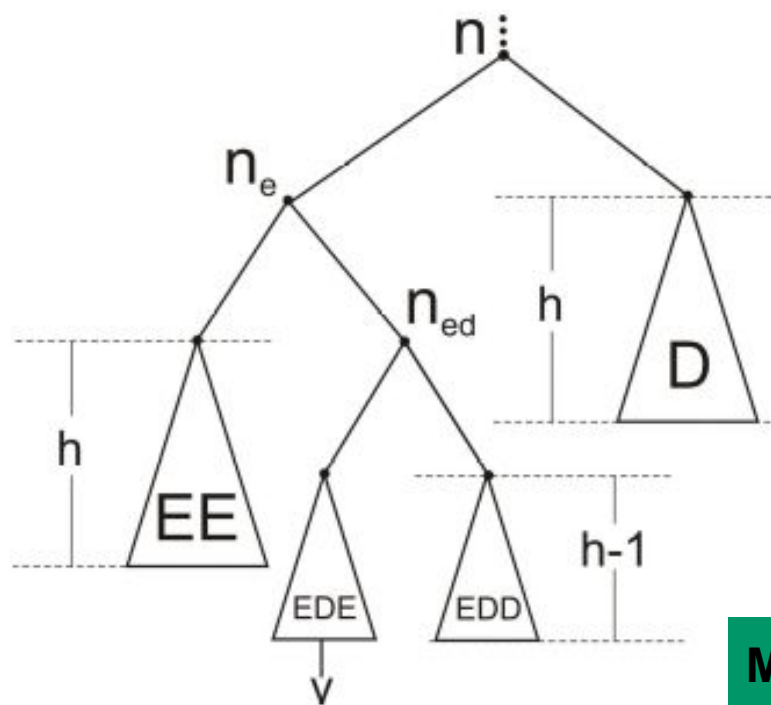
**n é sempre o
nó que está
desbalanceado**



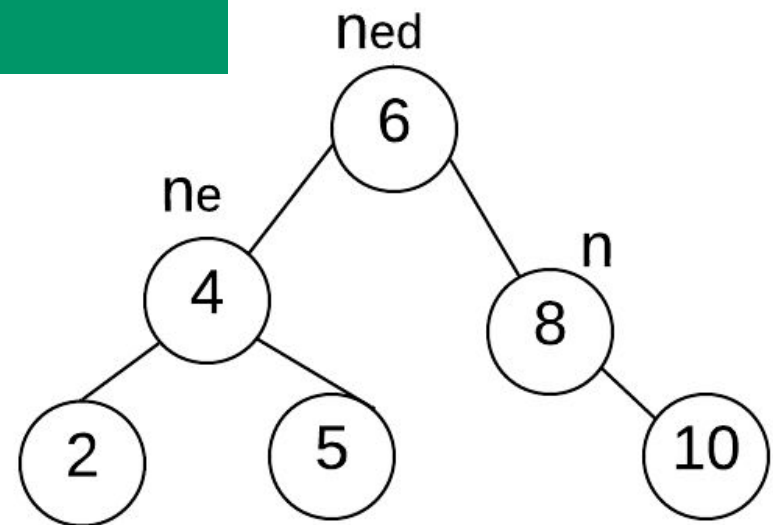
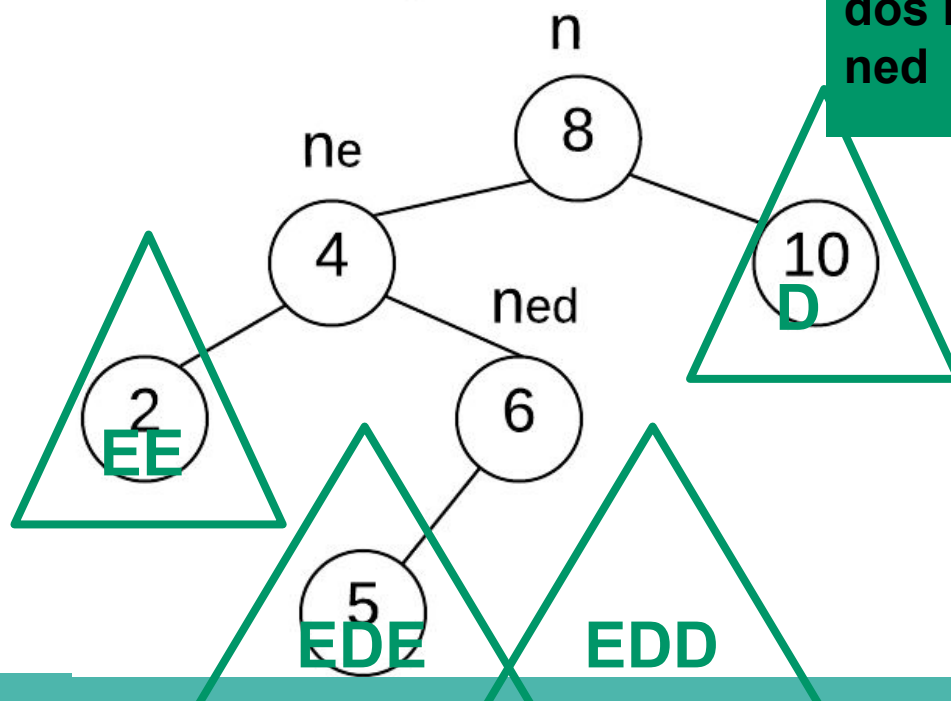


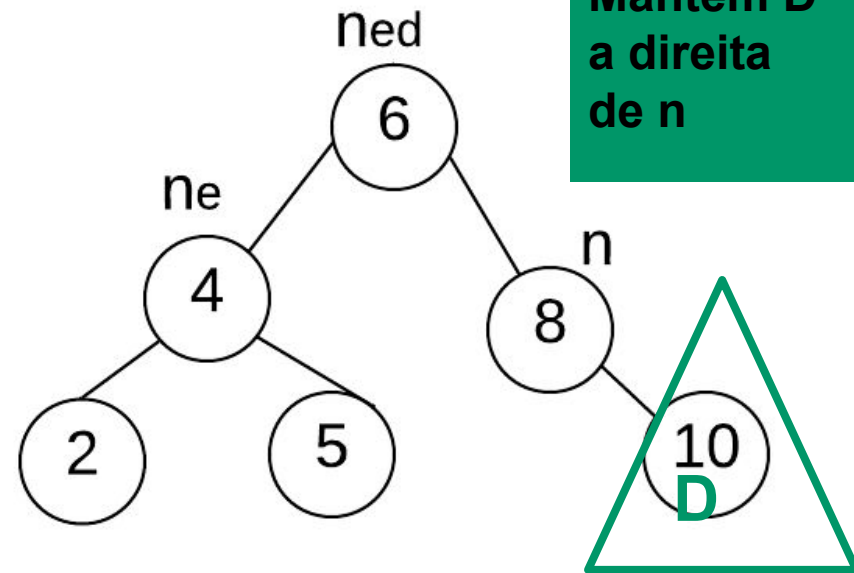
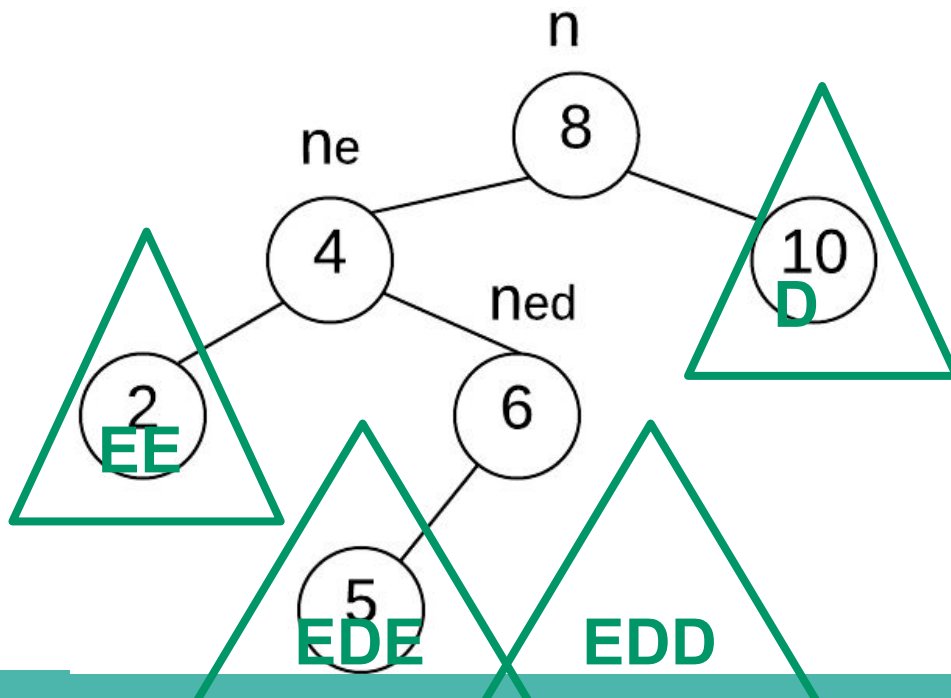
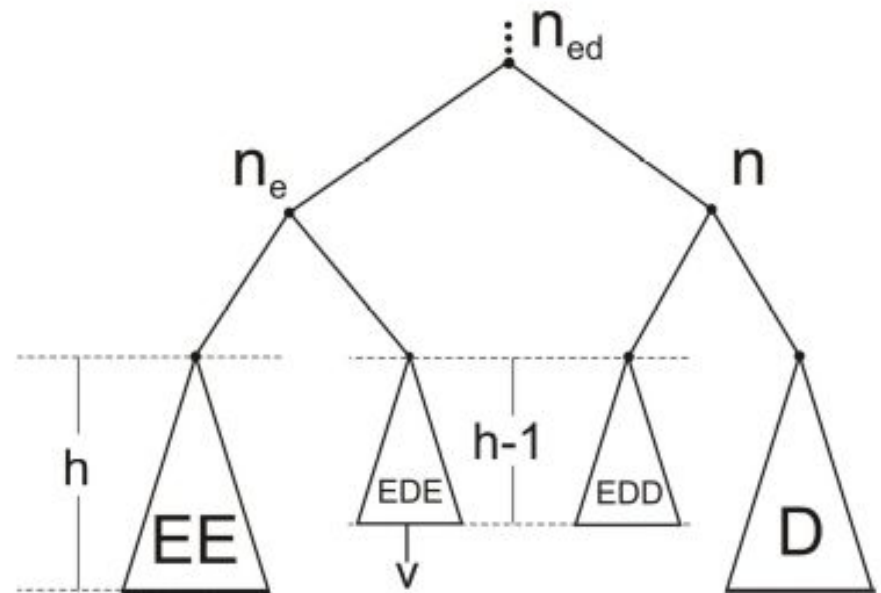
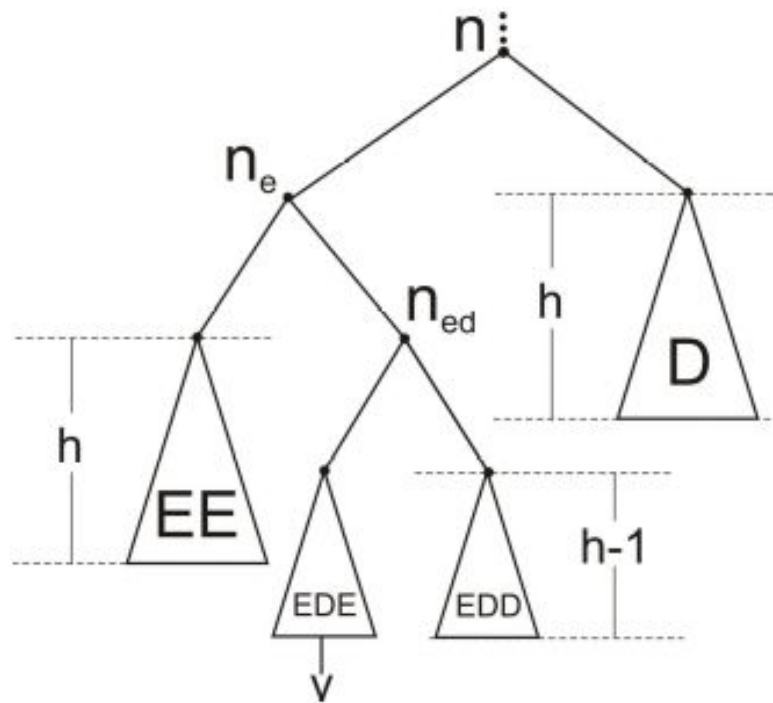
identificando
as árvores



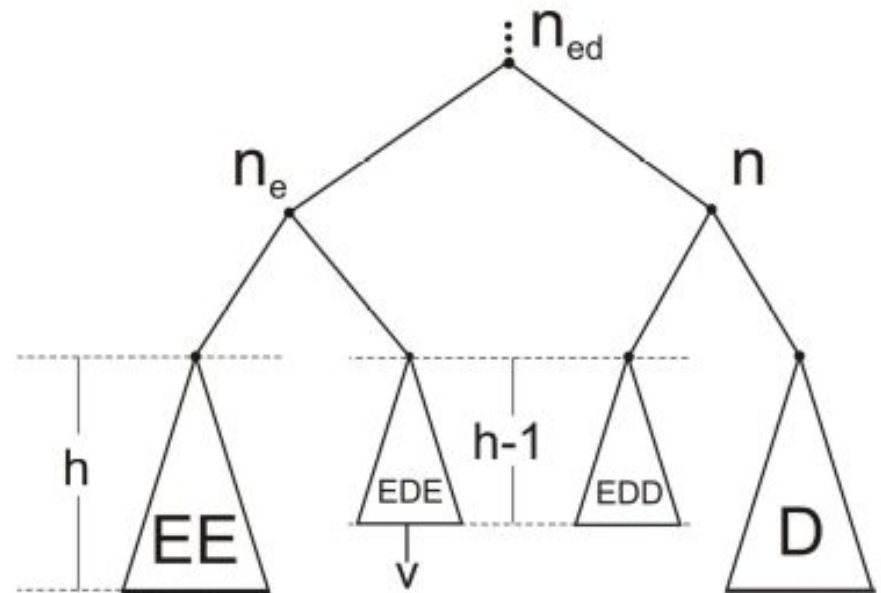
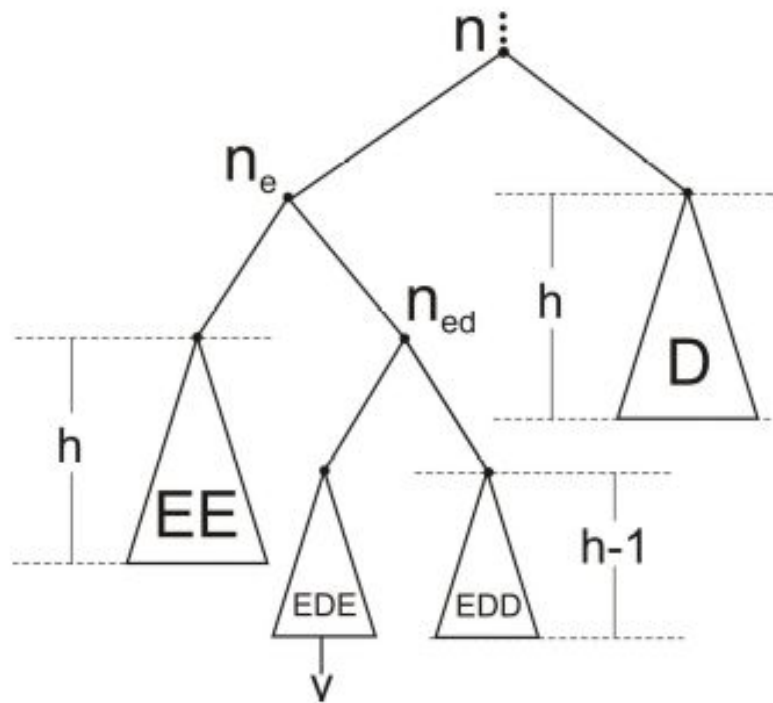


Muda ordem
dos nós n , n_e e
 n_{ed}

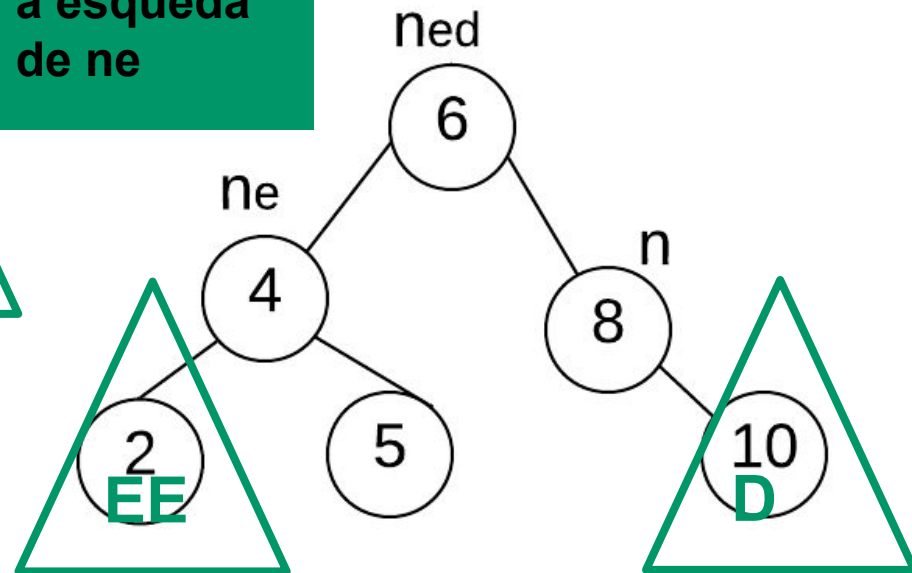
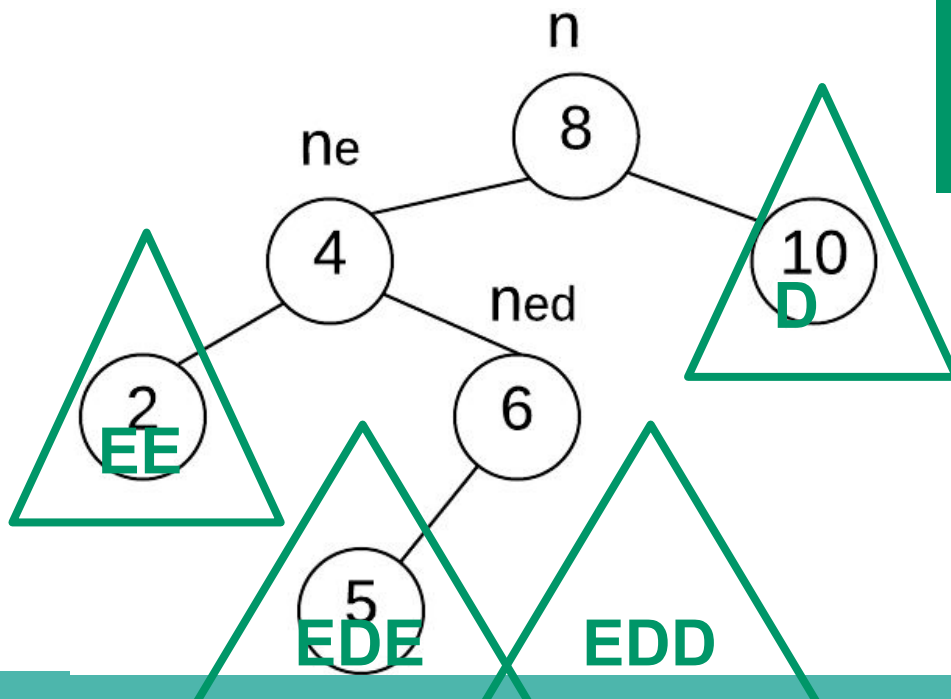


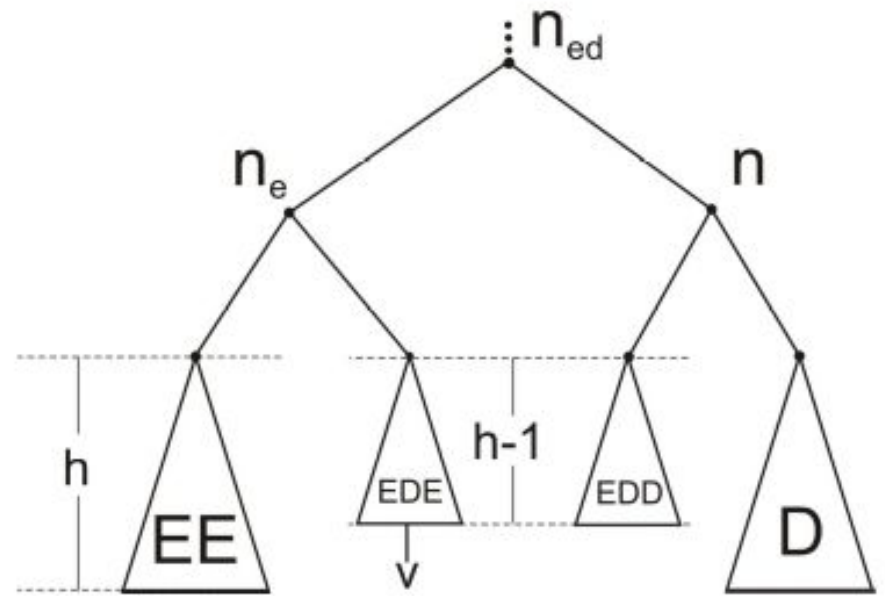


Mantém D
a direita
de n

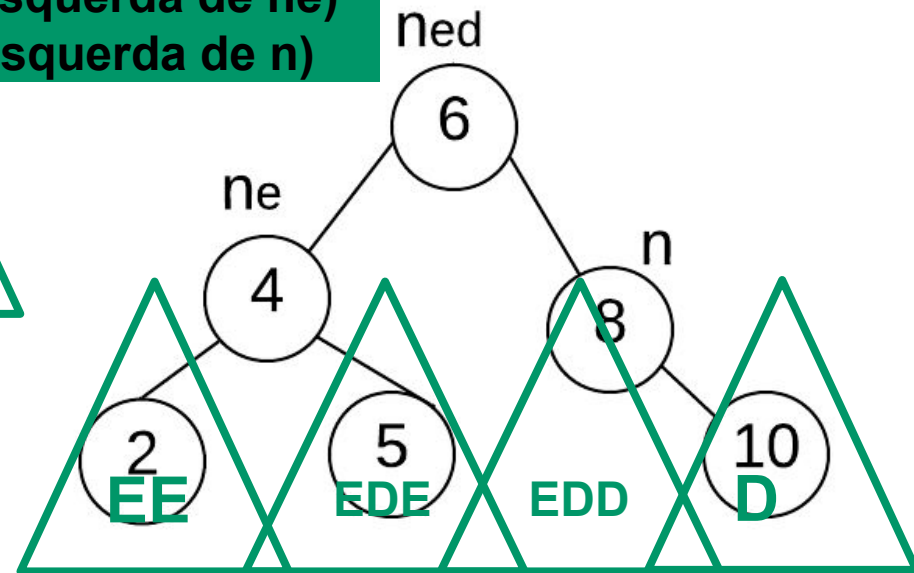
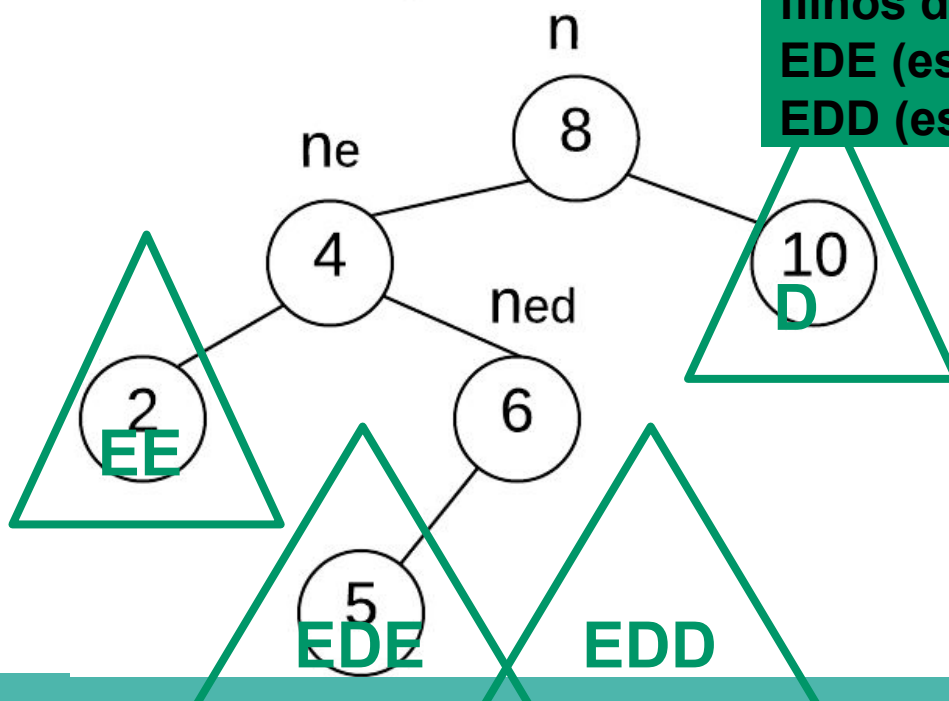


Mantém EE
à esquerda
de ne



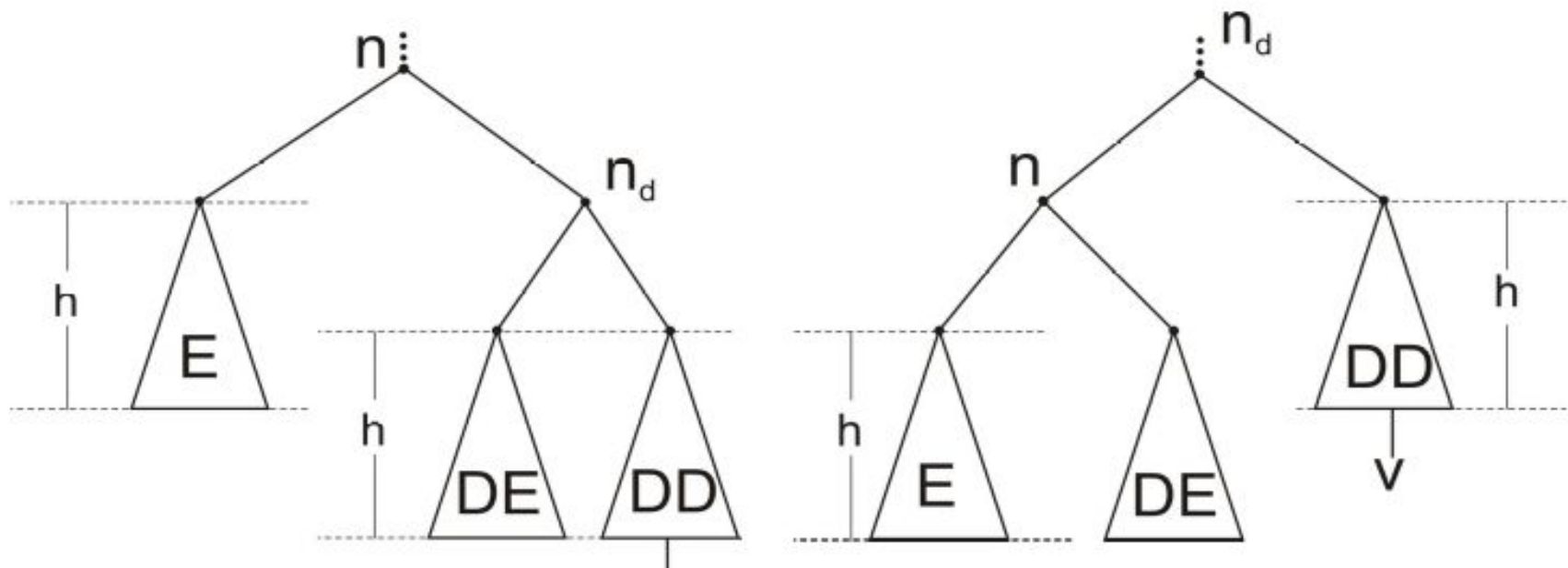


**Modifica todos os
filhos de ned:
EDE (esquerda de ne)
EDD (esquerda de n)**

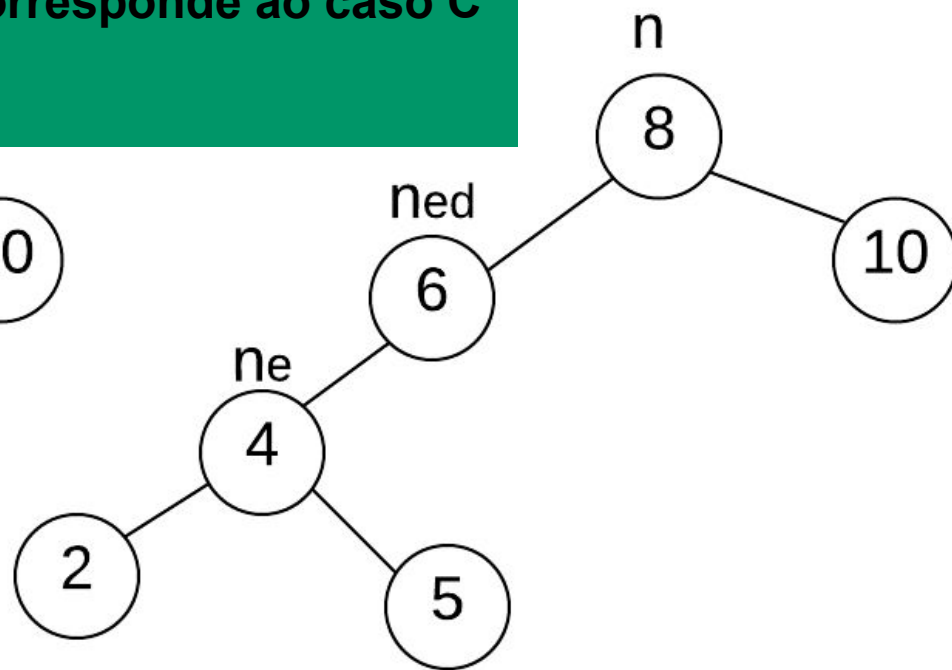
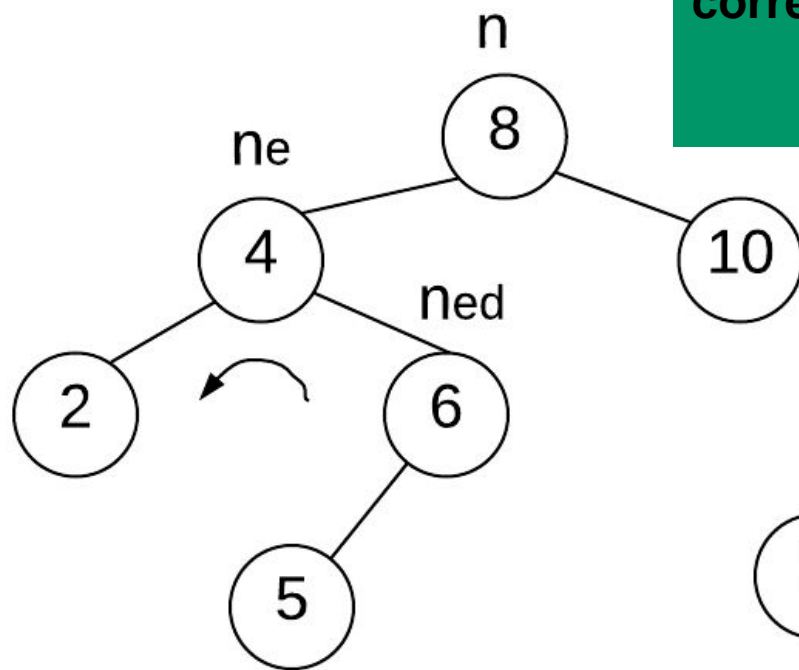


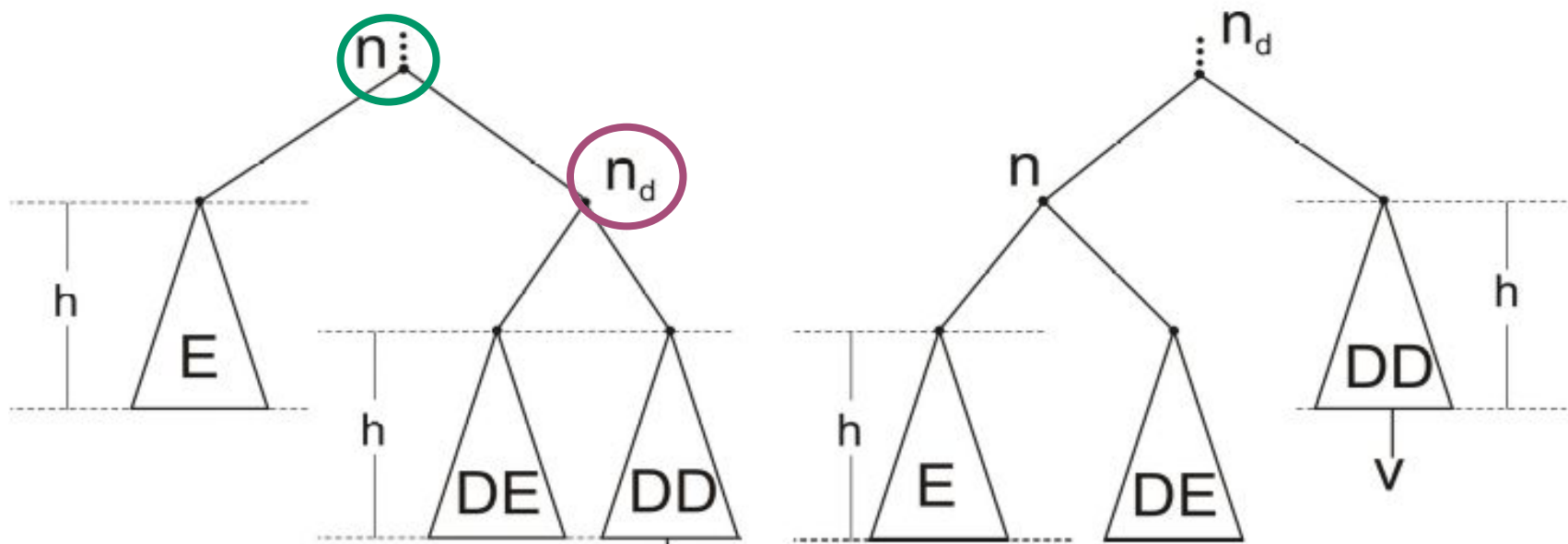
Exemplo - não direto (2 rotações)

Rotação à esquerda

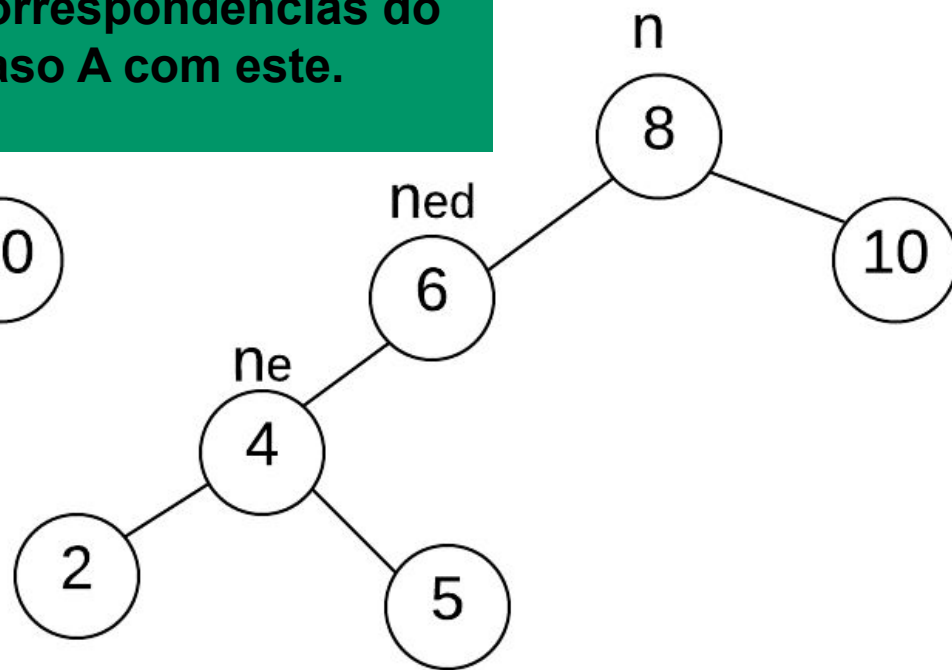
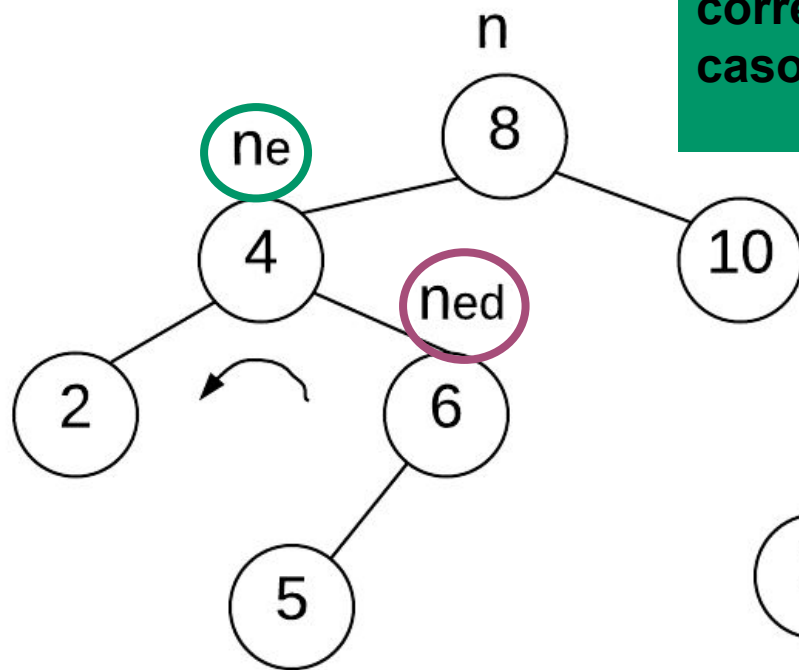


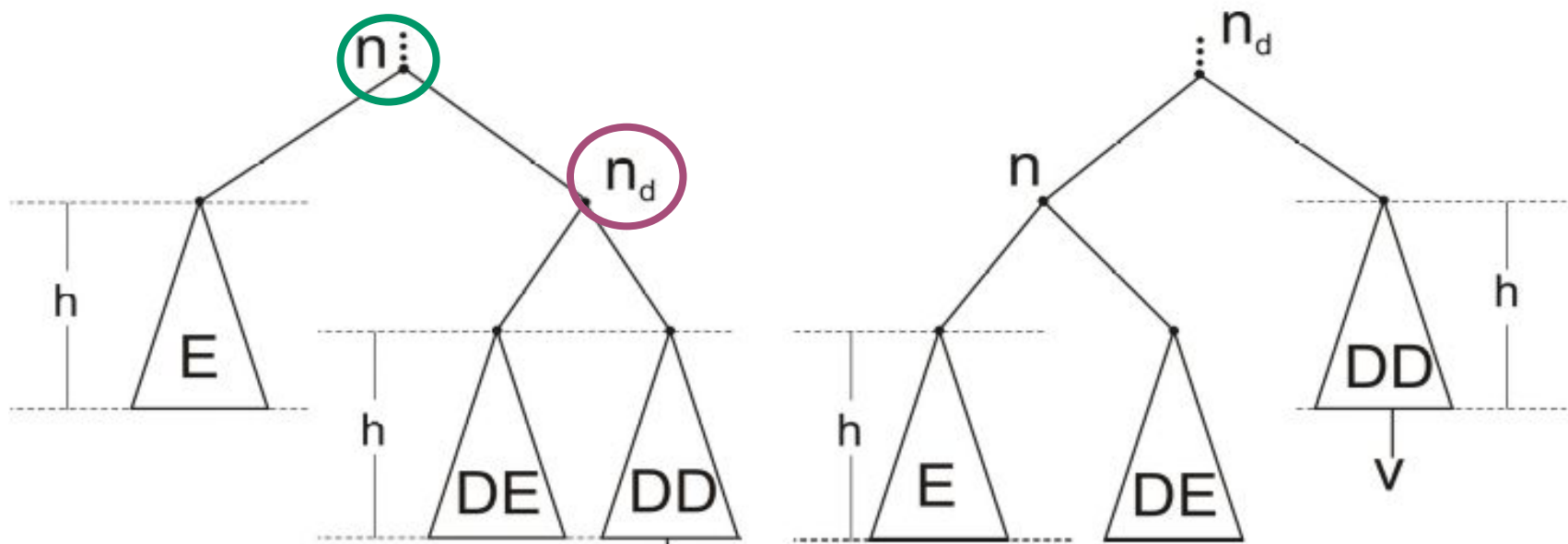
**Rotação à esquerda
corresponde ao caso C**



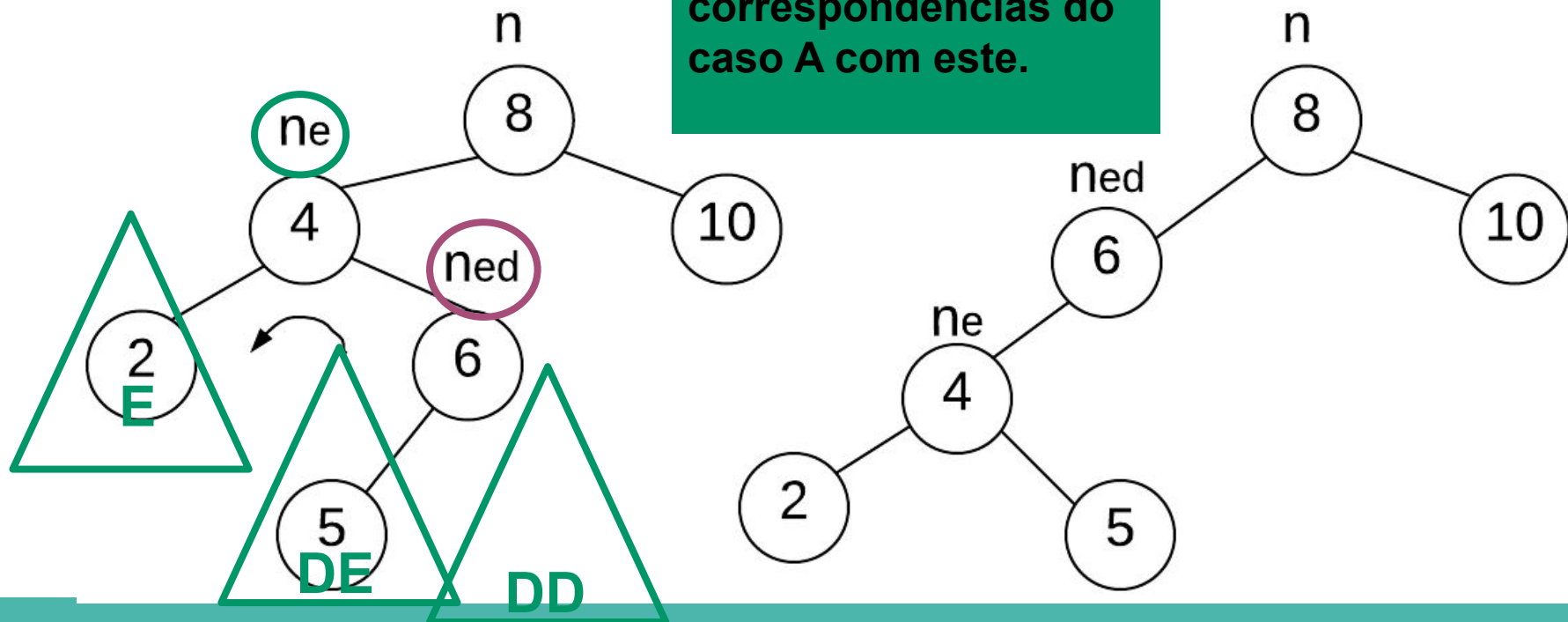


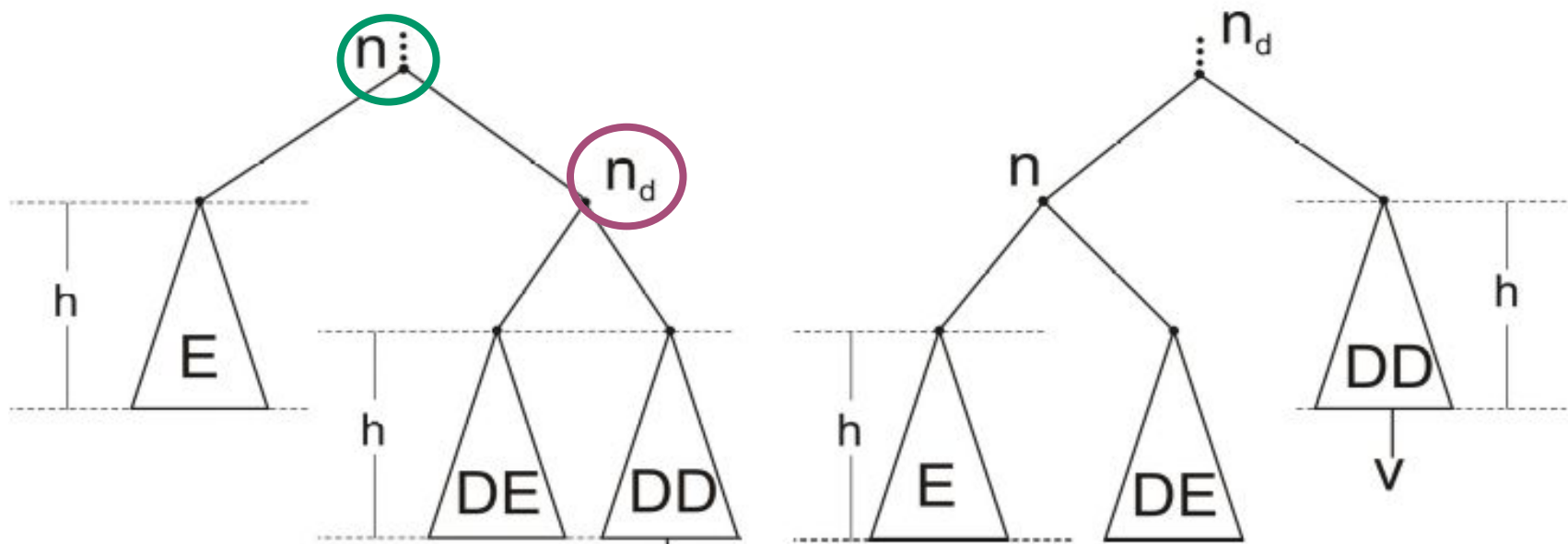
Achando as correspondências do caso A com este.



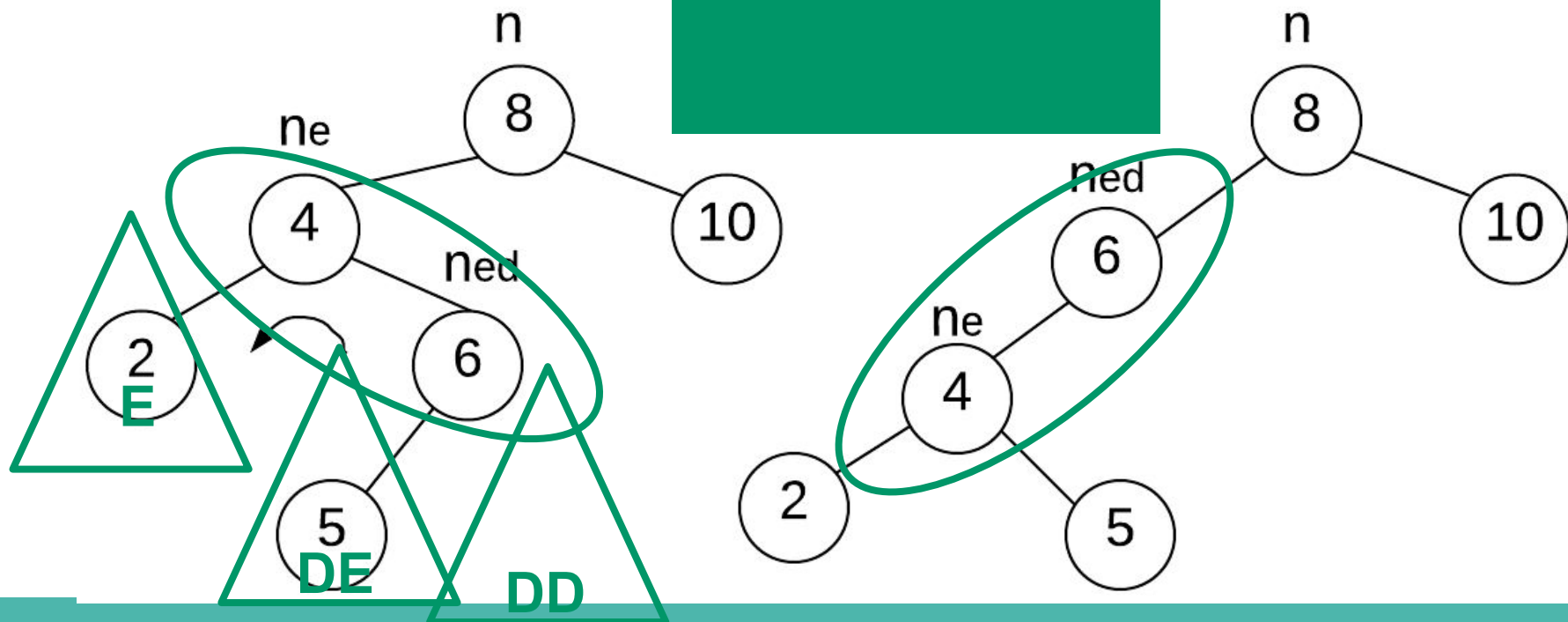


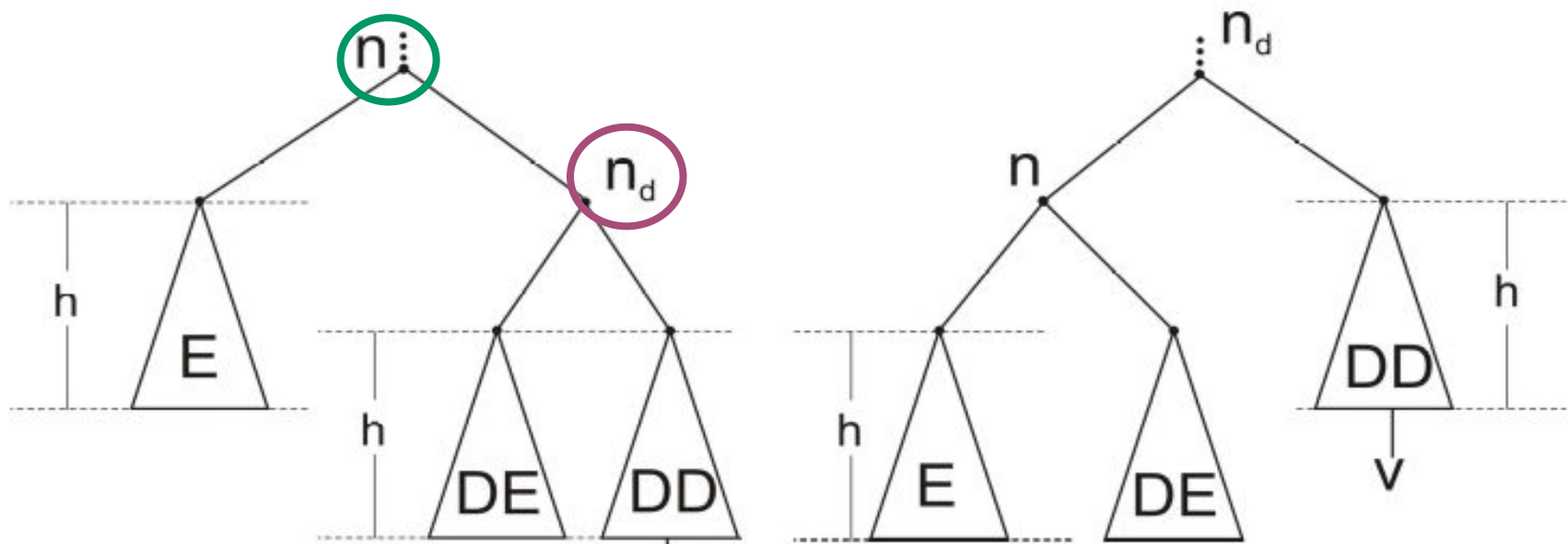
Achando as correspondências do caso A com este.



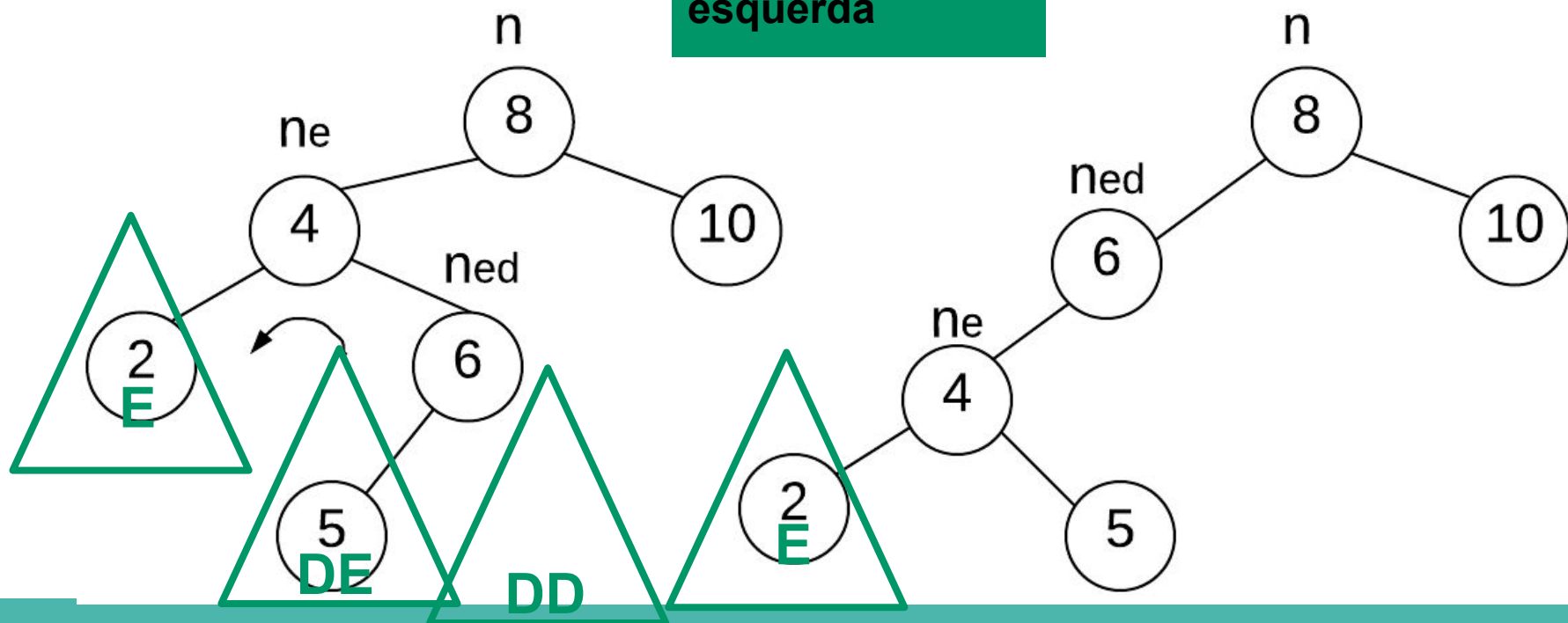


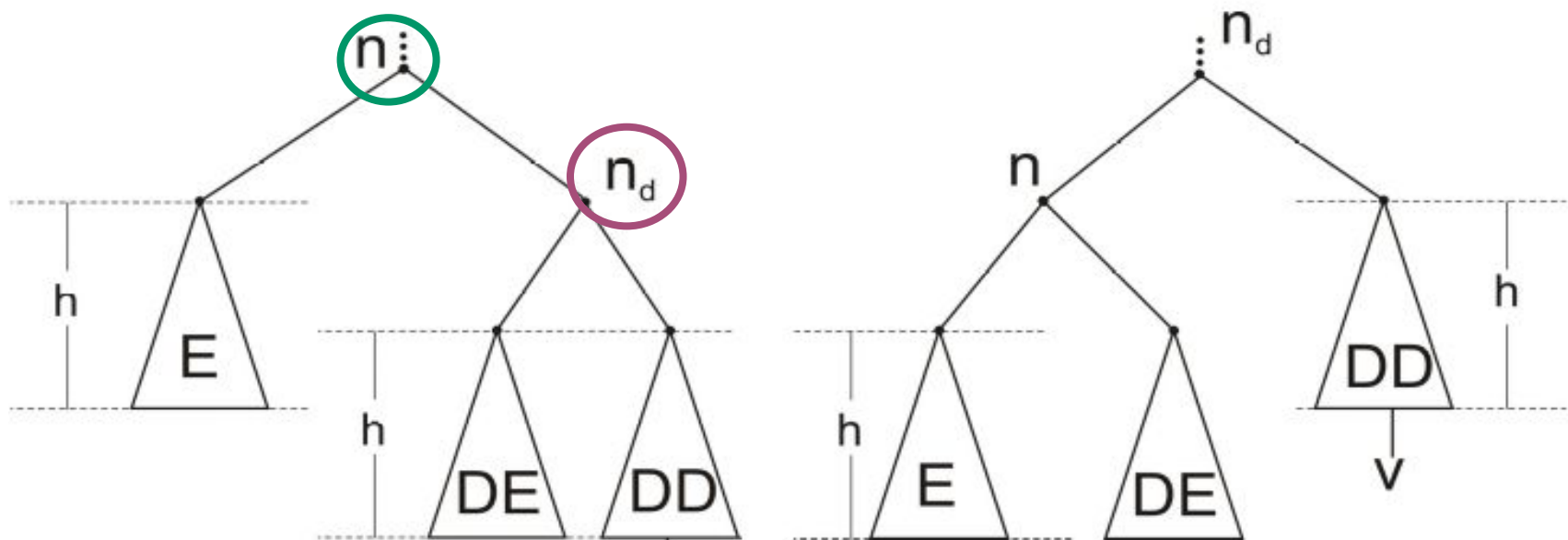
Modifica os nós



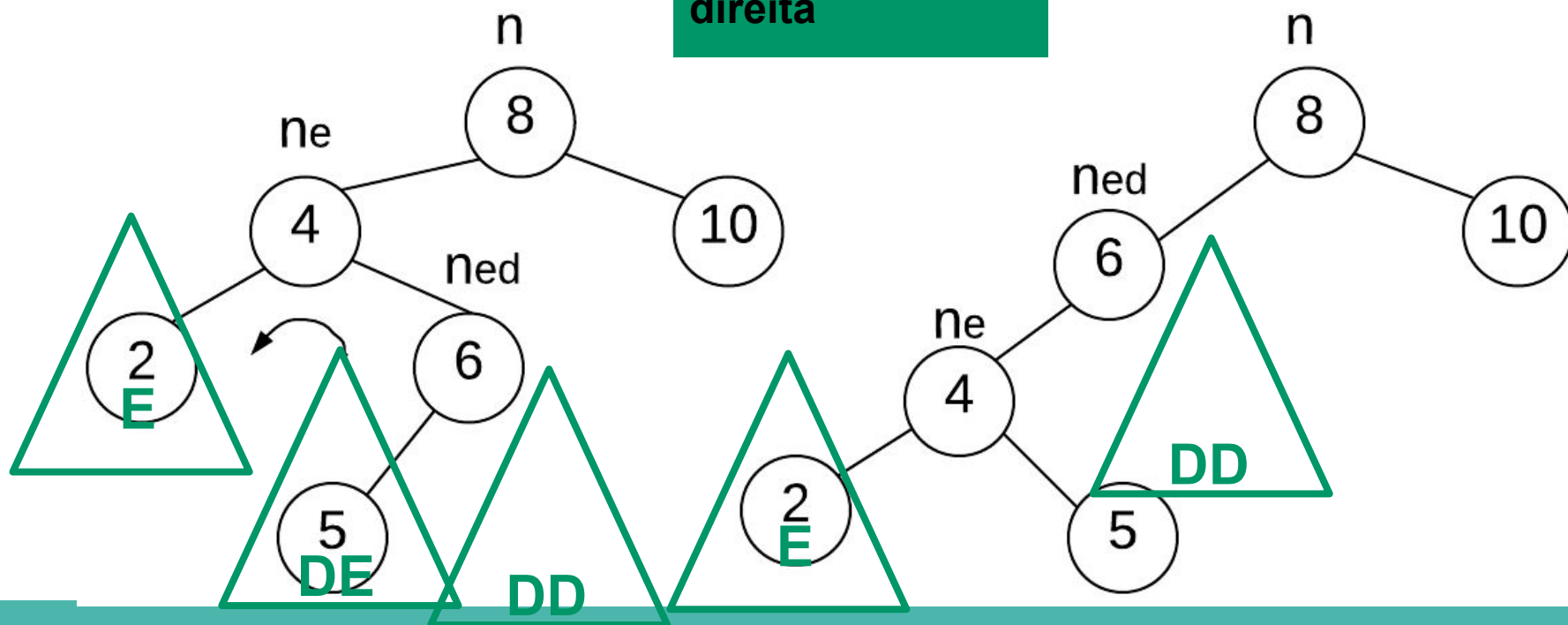


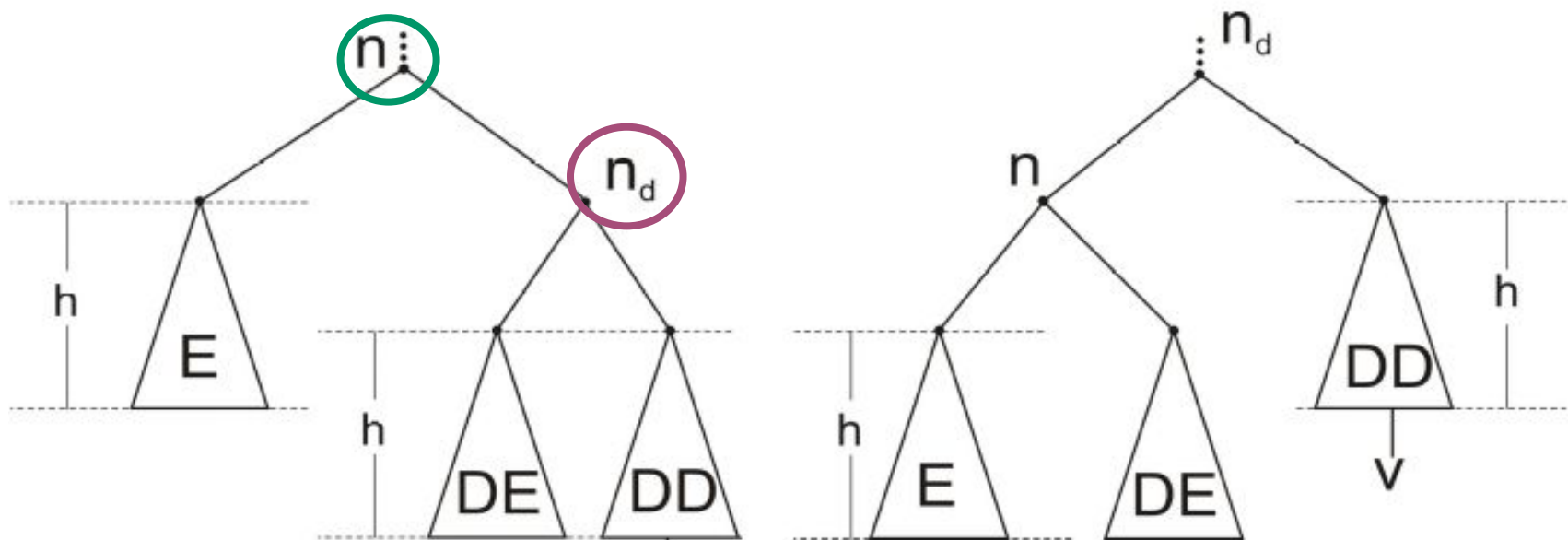
Mantém árvore esquerda



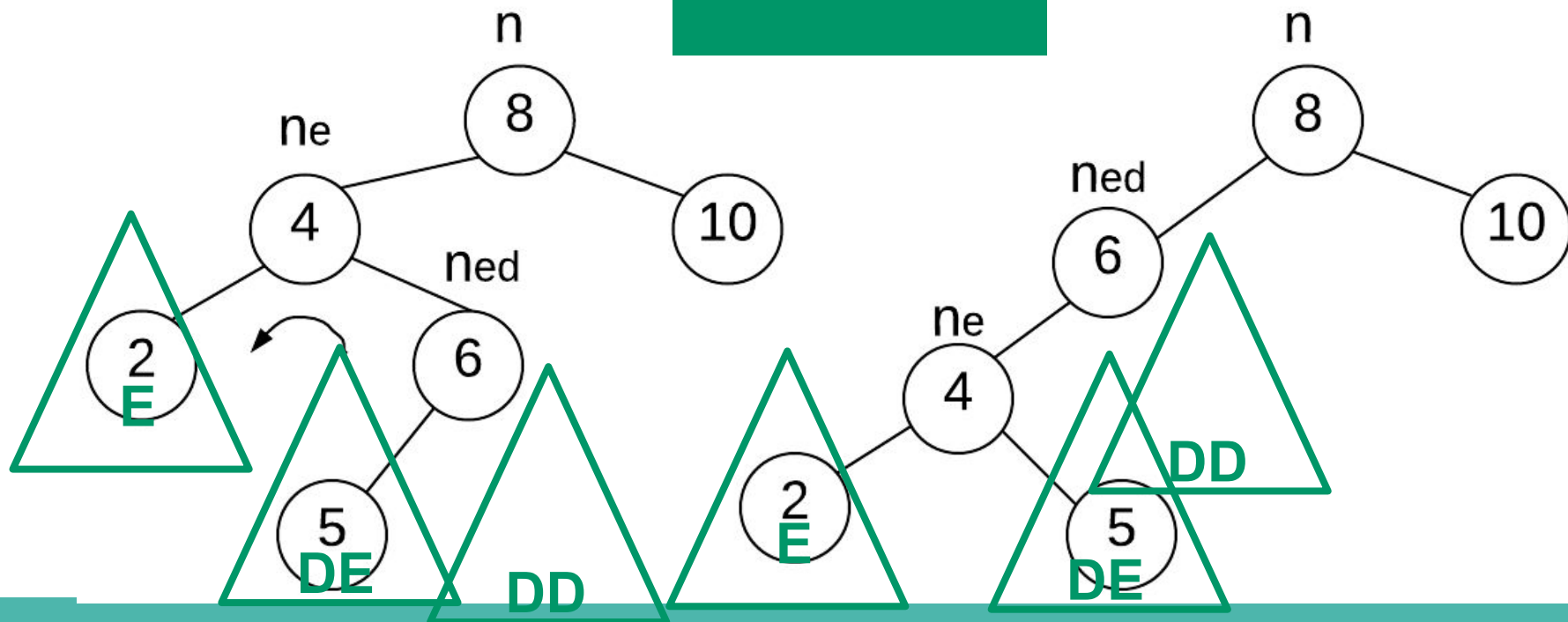


Mantém árvore
direita





Modifica DE

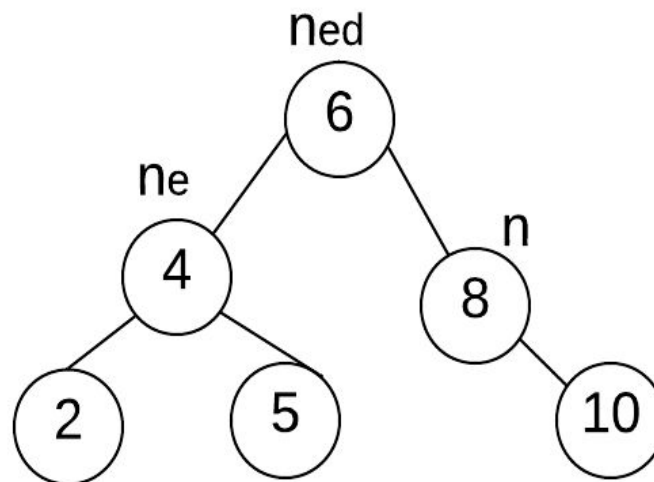
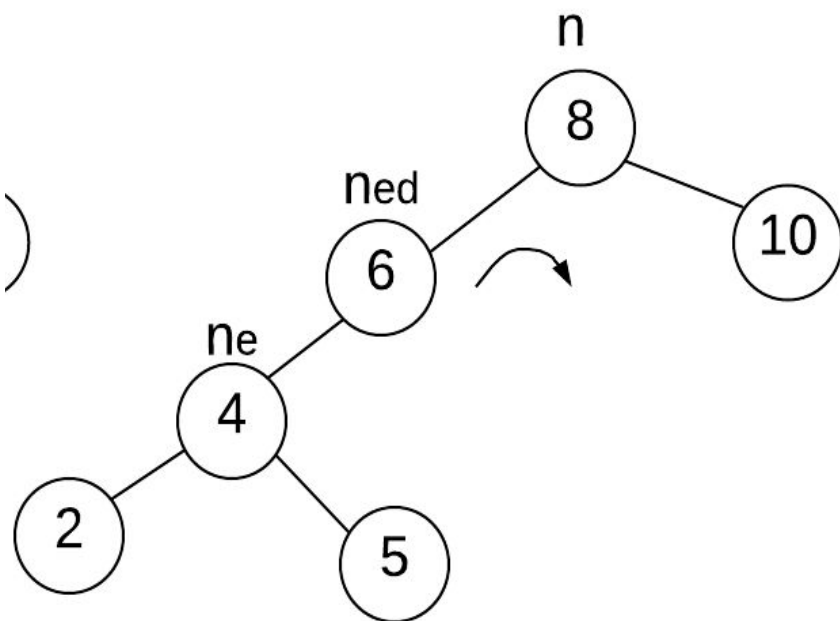


Exemplo - não direto (2 rotações)

Rotação à direita

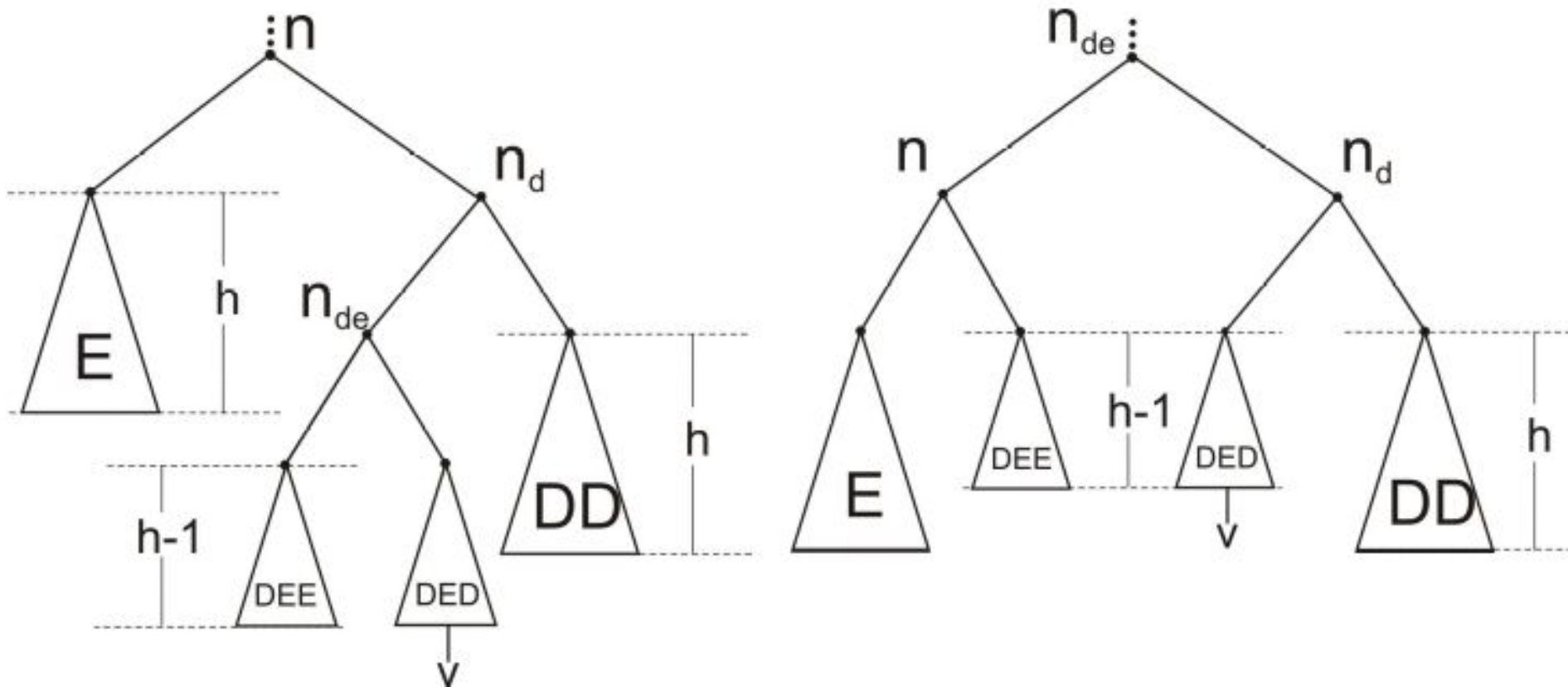
Fim do Exemplo - Mesma configuração da rotação direta

Rotação à direita => igual caso A



Caso D - Simétrico ao Caso B - Rotação dupla à esquerda

- Duas rotações simples: primeira em n_d para direita, segunda em n para esquerda
-

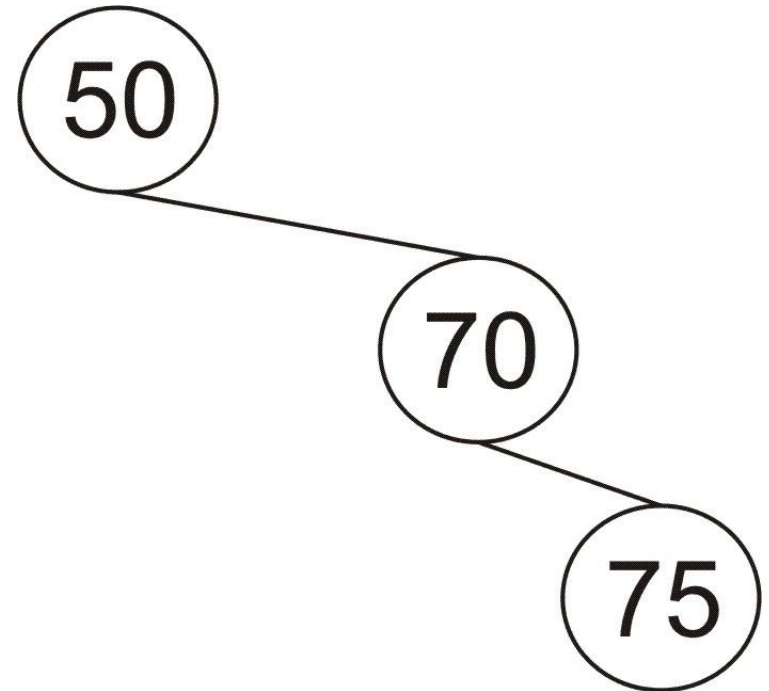
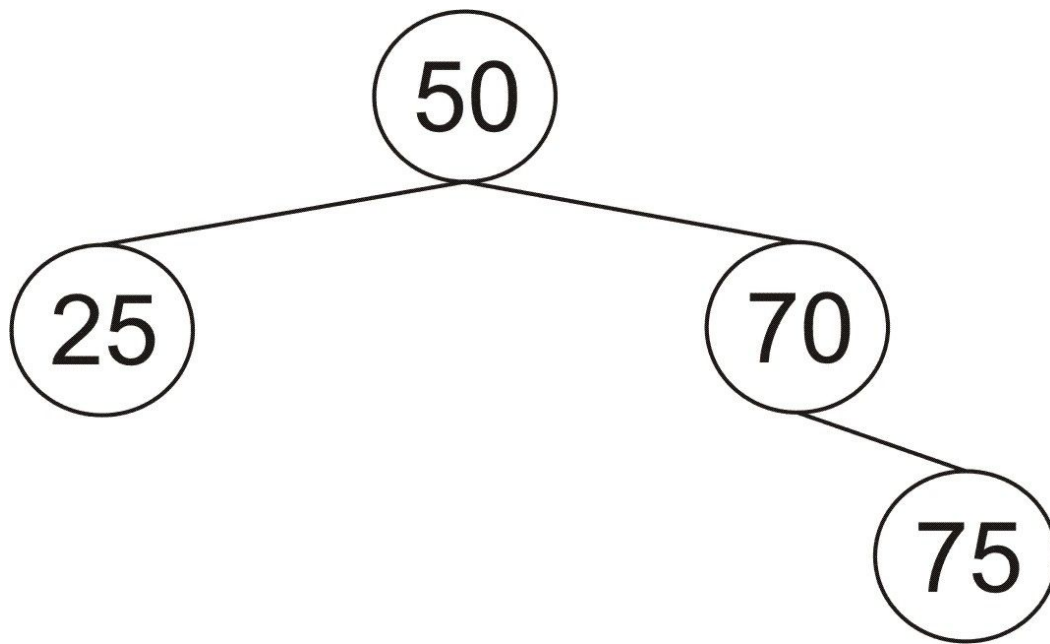


Relembrando o algoritmo para remoção em ABB

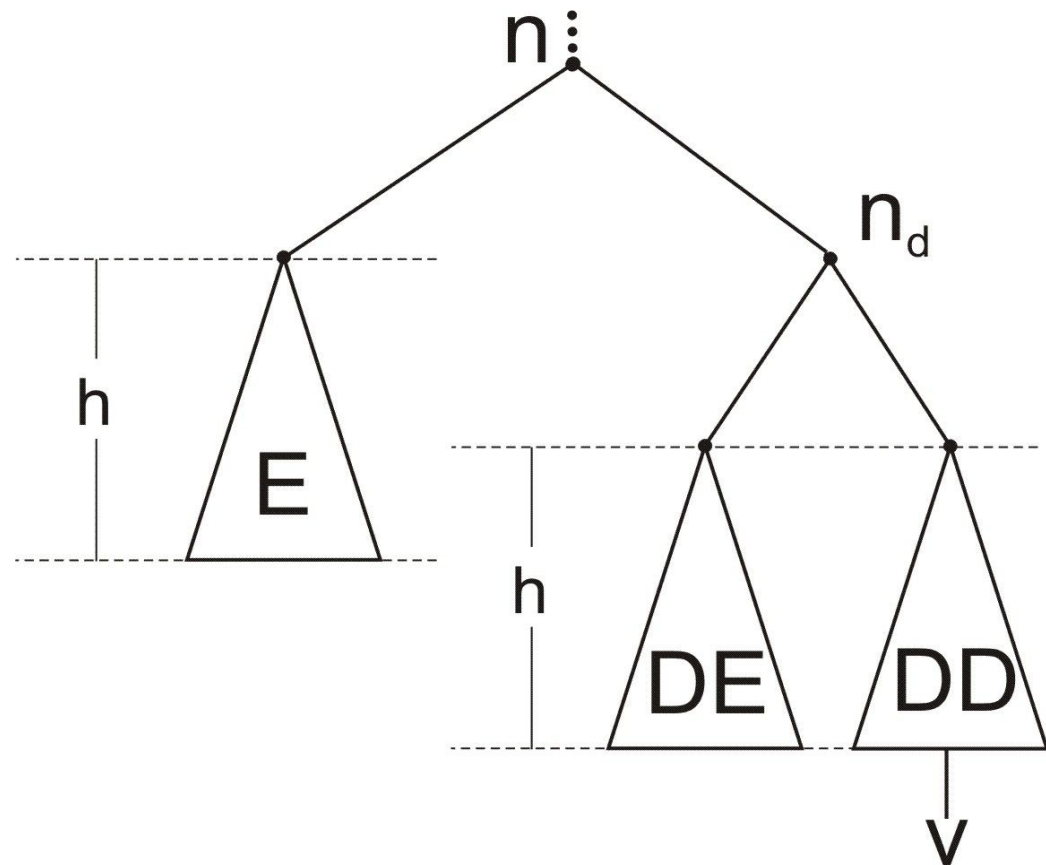
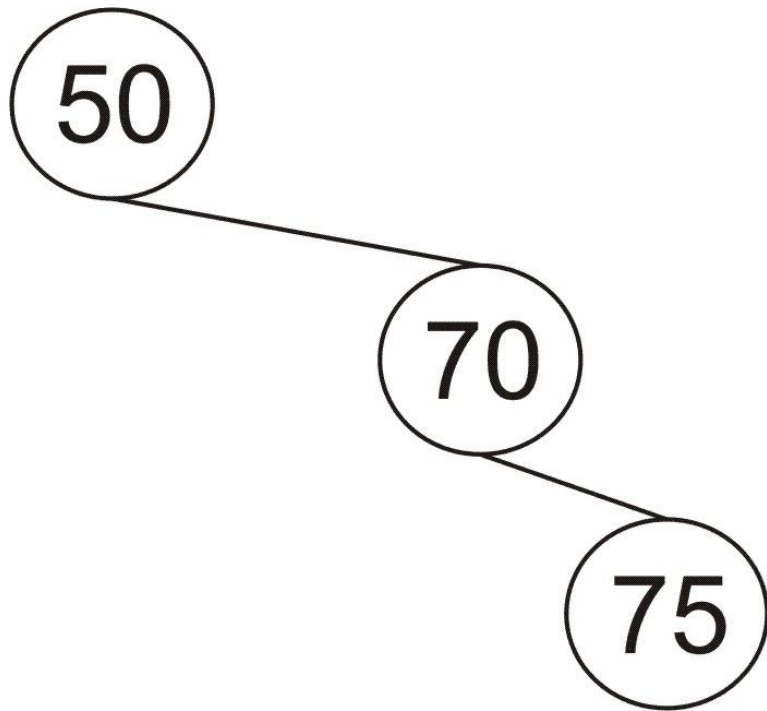
- Passos:
 - busca o nó a ser removido;
 - Se é folha, remove
 - Se possui apenas um filho, coloca este filho no seu lugar (o filho substitui o nó);
 - Se tem dois filhos, busca o **antecessor (ou sucessor)** copia os valores do nó pelo valores do sucessor; remove o antecessor.
- Para a árvore AVL: verifica se a remoção acarretou em desbalanceamento, faça as rotações necessárias para cada caso.

Exemplo 1 - remover o 25

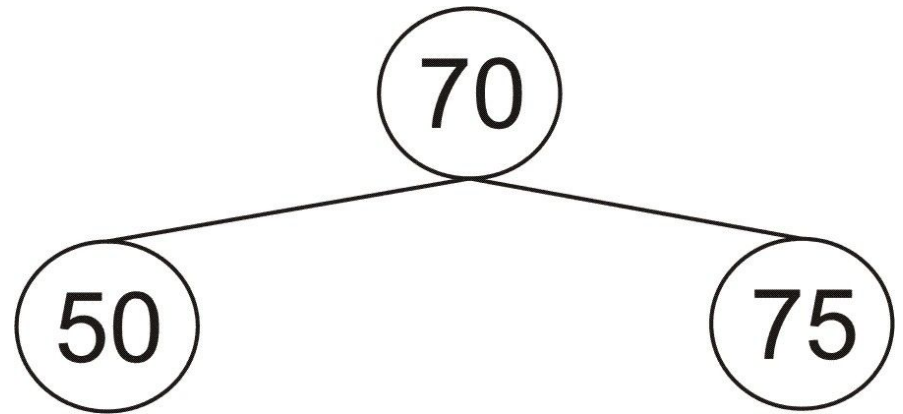
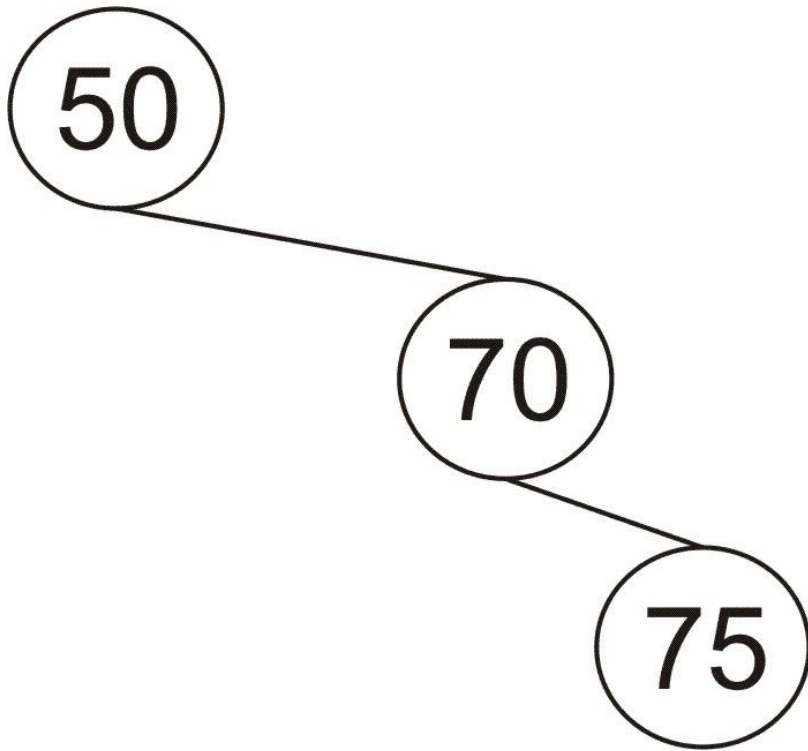
Desbalanceada!



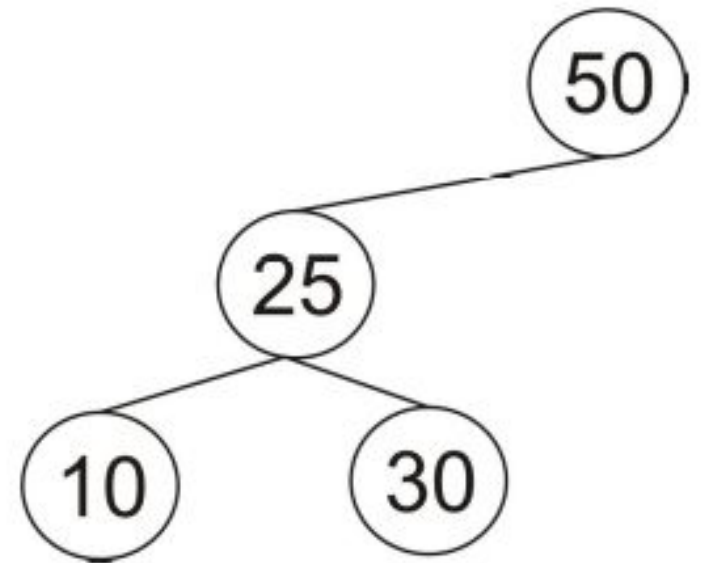
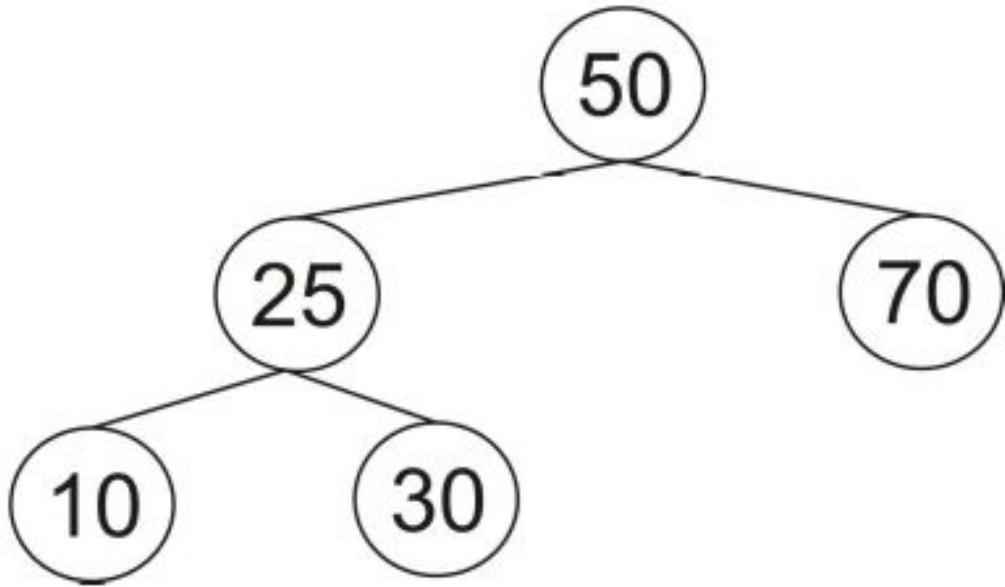
Caso C - Rotação a esquerda



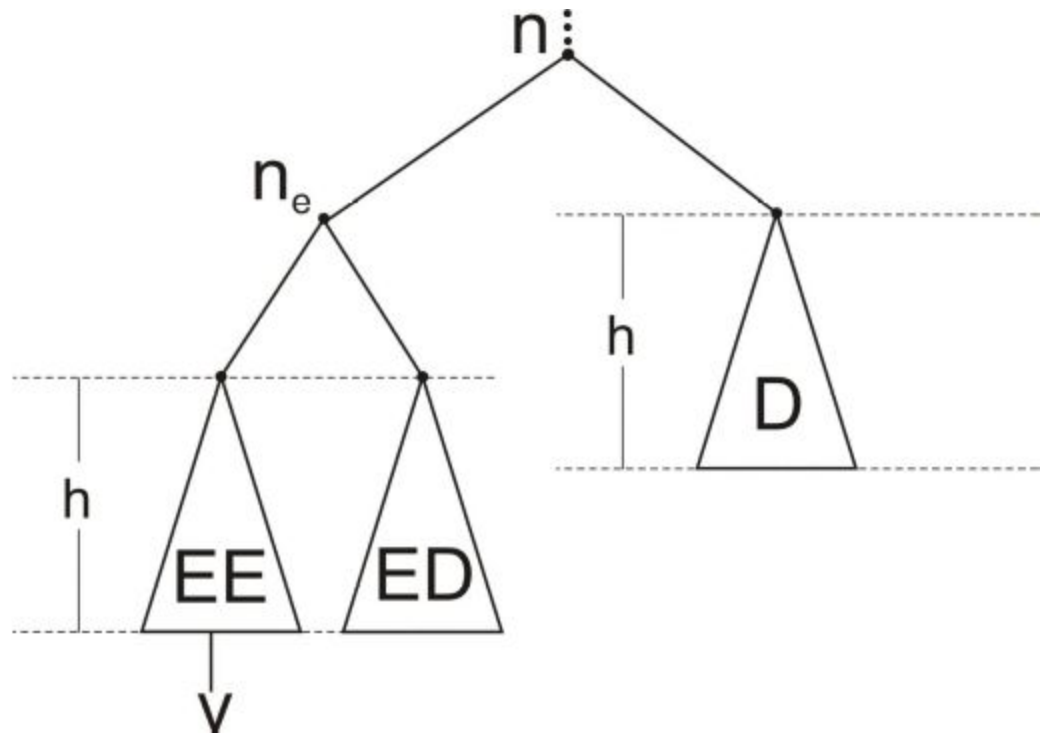
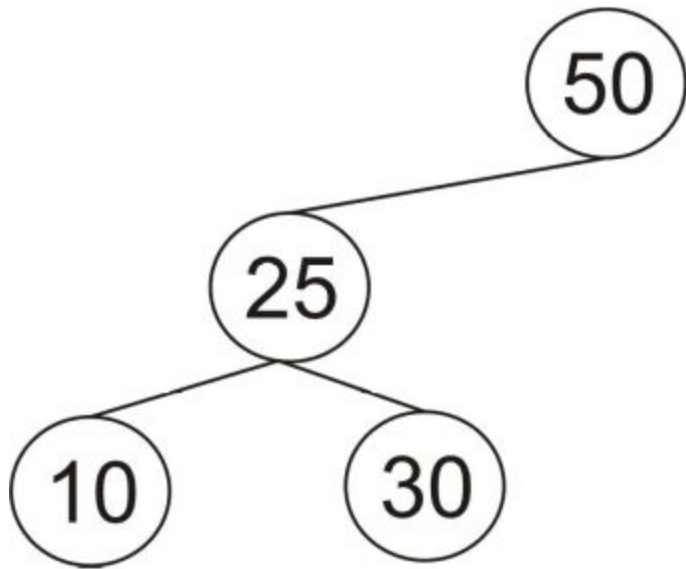
Balanceando...



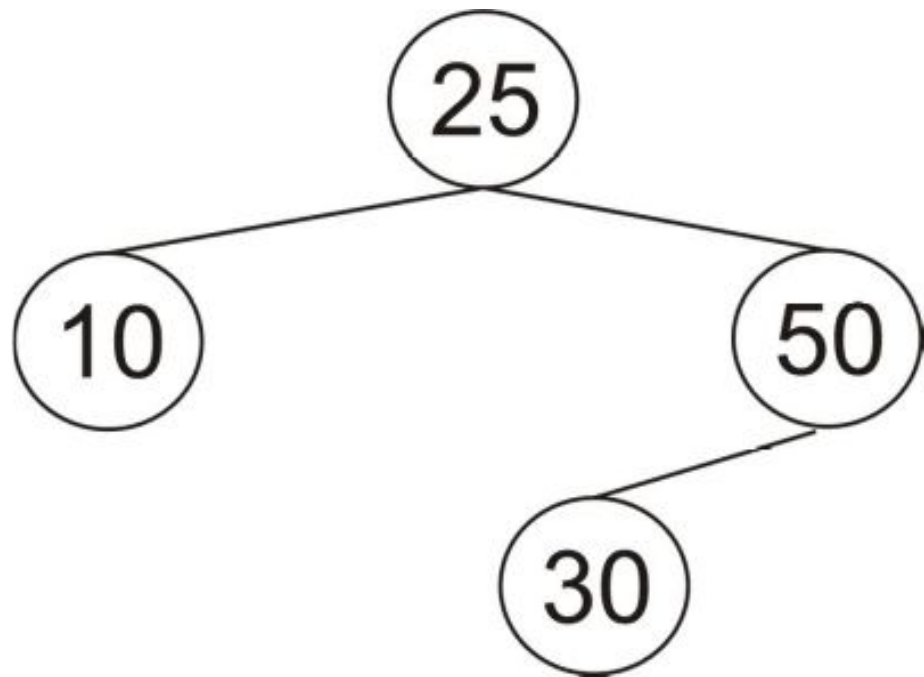
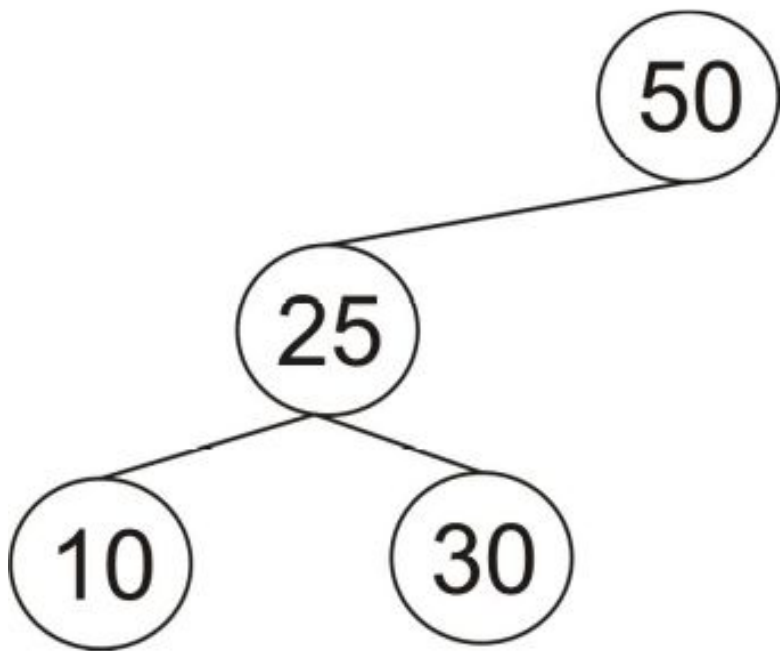
Exemplo 2 - Remover 70



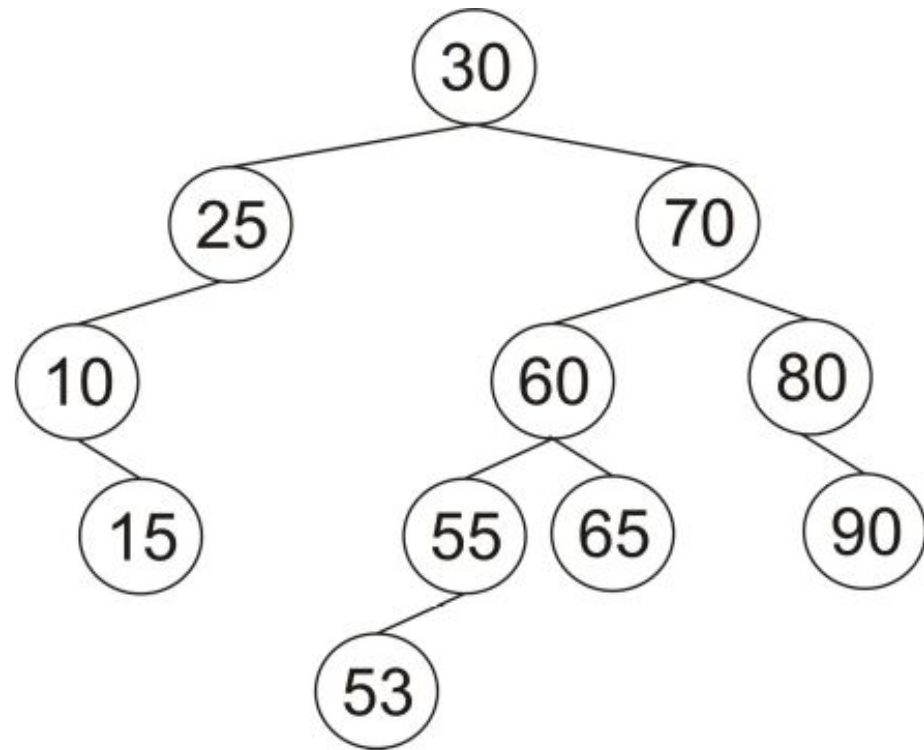
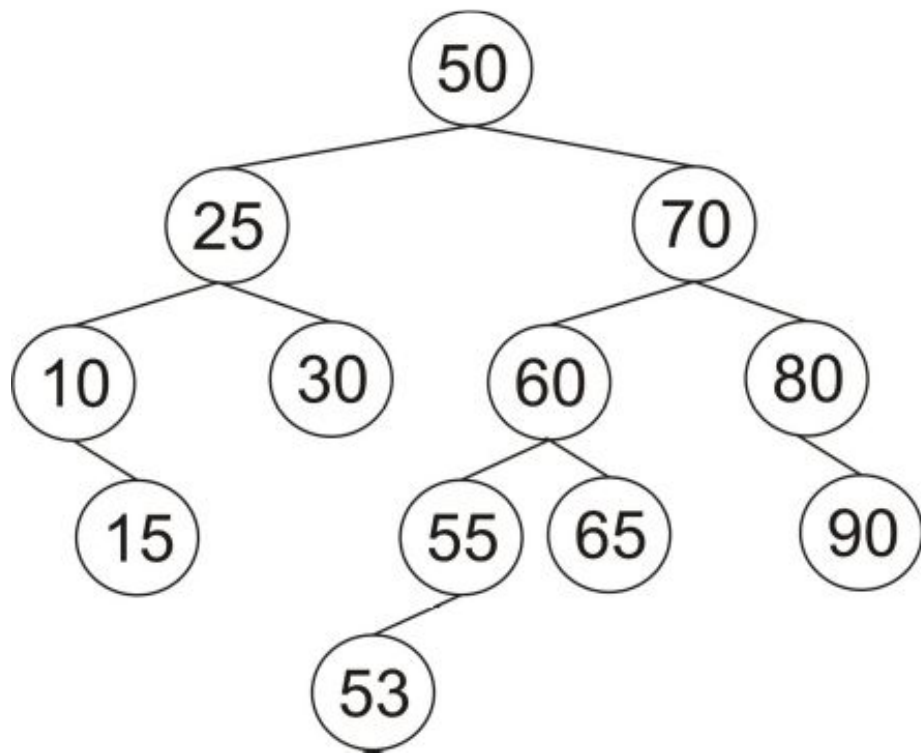
Exemplo 2 - Caso A - rotação à direita



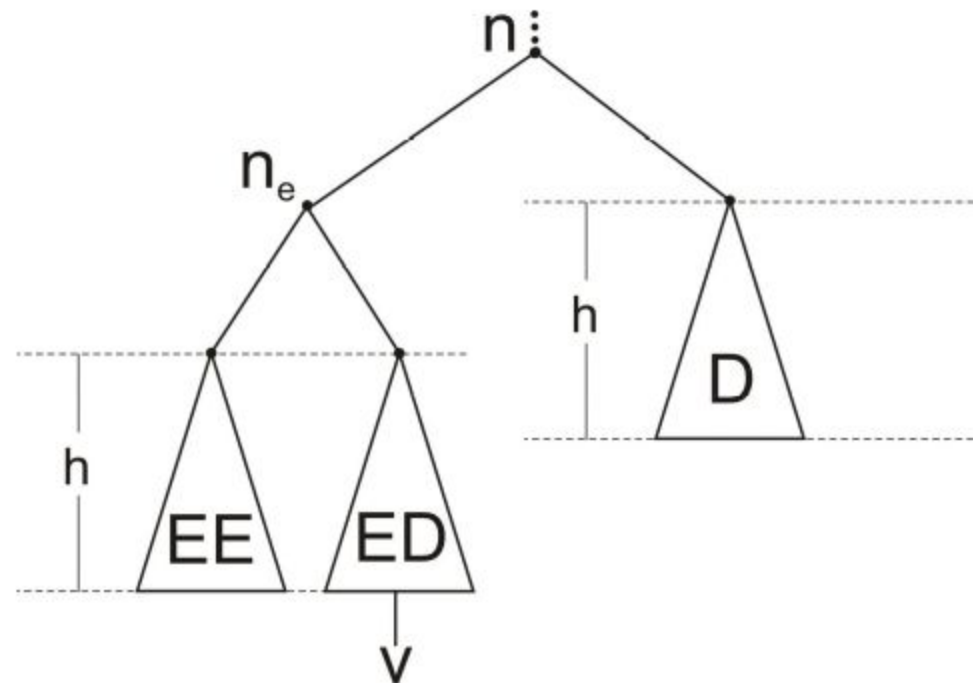
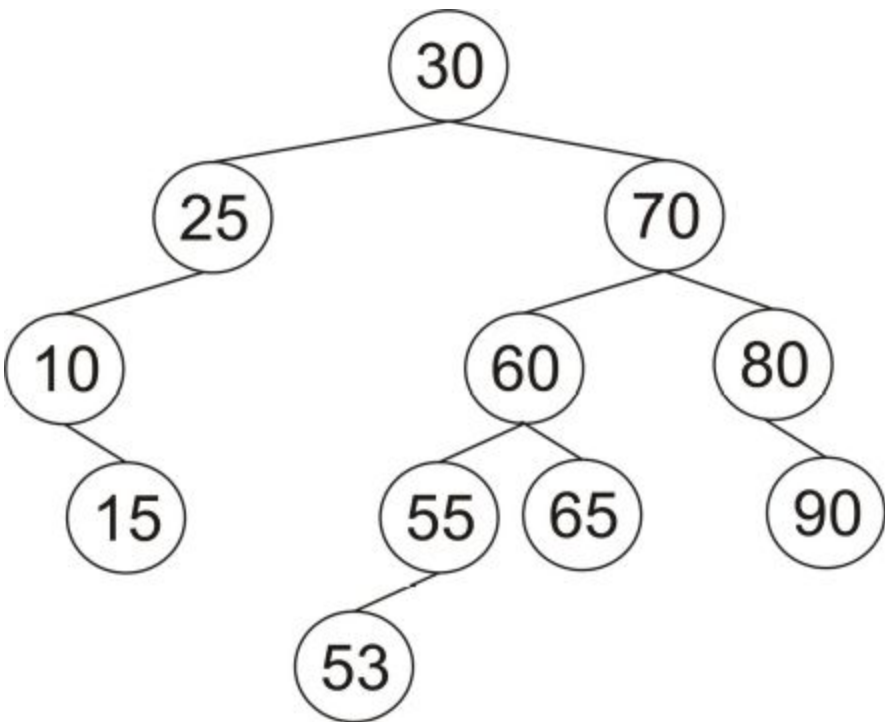
Balanceando...



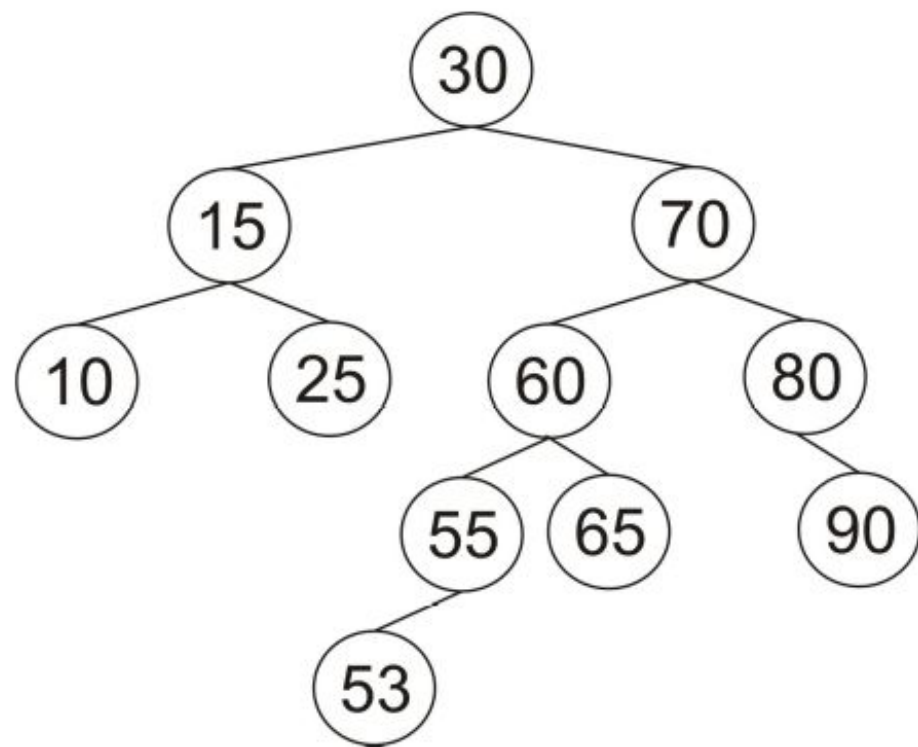
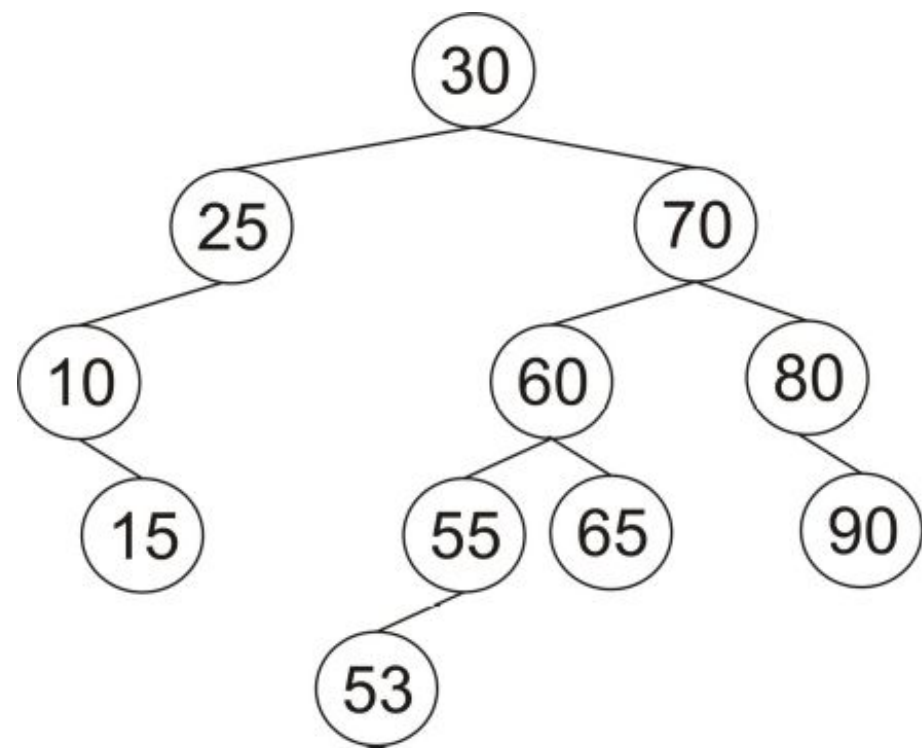
Exemplo 3 - Remover 50



Exemplo 3 - Caso B



Balanceando... duas rotações



Exercício - sala de aula

1. Desenhe o passo a passo da inserção dos seguintes valores na árvore AVL. Para cada rotação, mostre o estado anterior (desbalanceado) e o estado posterior (balanceado após rotação). Sequência: 20, 15, 10, 25, 30, 22, 21.
2. Para a árvore resultado do exercício 2, desenhe a a árvore após a remoção de cada um dos seguintes elementos: 30, 20 e 10. Faça rotação quando precisar.
3. Tente fazer sozinho primeiro. Confira passo a passo aqui: [AVL Tree Visualization](#)