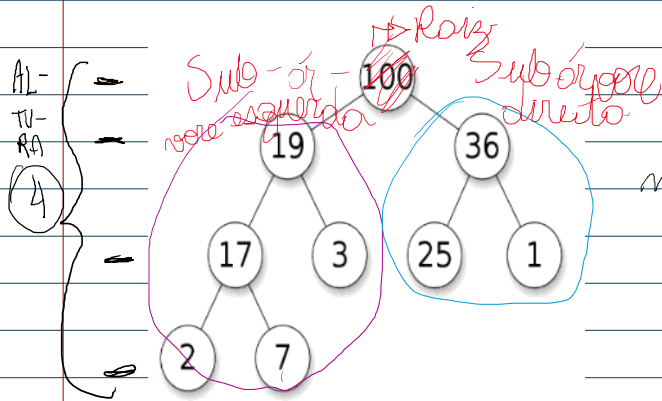


Árvore: grupo conexo e sem ciclos.  
 ↳ binária: tem 0, 1 ou 2 filhas.

Árvores binárias podem ser classificadas como:

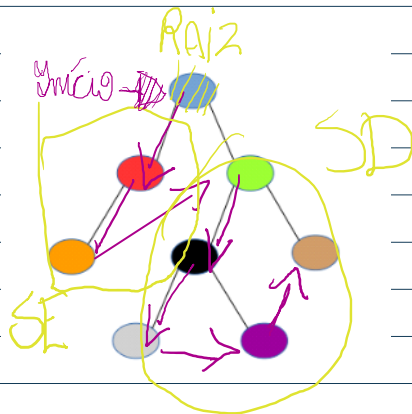
- Vazia (porque p/ a raíz não);
- no raíz contendo subárvores.  
 ↳ direita/esquerda.



Isso ocorre p/ todas as nós, não apenas a raíz.

↳ forma de percorrer  
 \* Procura em árvores binárias

- Pré-ordem: Acessa a raíz, percorre a SE e depois a SD.



Pré-ordem

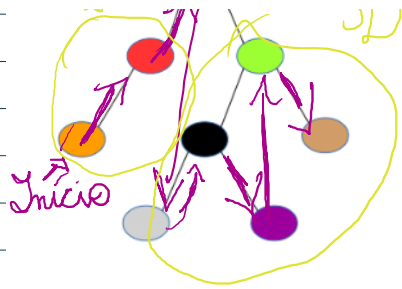


- Em-ordem: Acessa a SE, a raíz e depois a SD.



Em-ordem

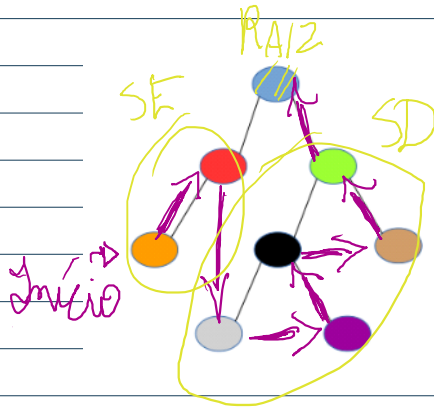




Em-ordem



• Pós-ordem: SE, SD e raiz.

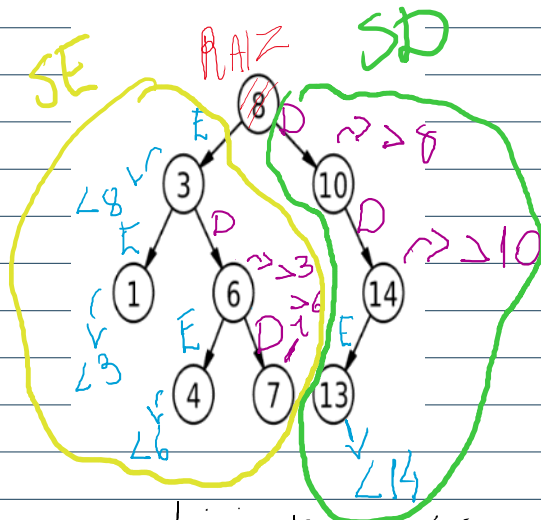


Pós-ordem



\* Como é uma ABB?

Em uma ABB, p/ cada nó, temos que elementos maiores serão filhos à direita, e menores serão filhos à esquerda.



Como pode ser visto, elementos situados à DIREITA sempre serão maiores que os pais. Já elementos que estão à ESQUERDA, menores.

P/ que buscas sejam eficientes, a SE e a SD devem ter um número de elementos igual ou muito próximo (1 elemento de diferença no máximo).

• Máximo de nós:  $2^{h+1} - 1$  nós • Mínimo de nós:  $\Theta(\log n)$

altura = h  
↳ qtd de níveis  
(nós verticais)

altura = n  
↳ podemos alcançar qualquer nó em  $\Theta(\log n)$  passos

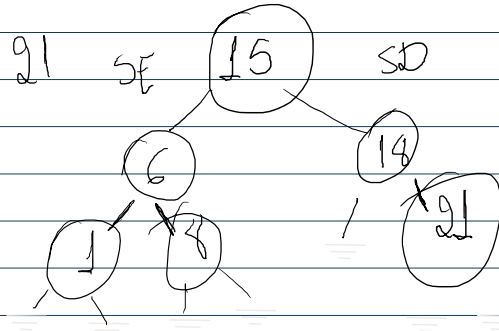
$O(\log n)$  pessoas

## \* Métodos

### ↳ Inserção:

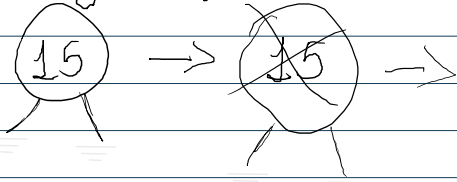
15, 6, 18, 1, 8, 21 SE SD

Altura = 3

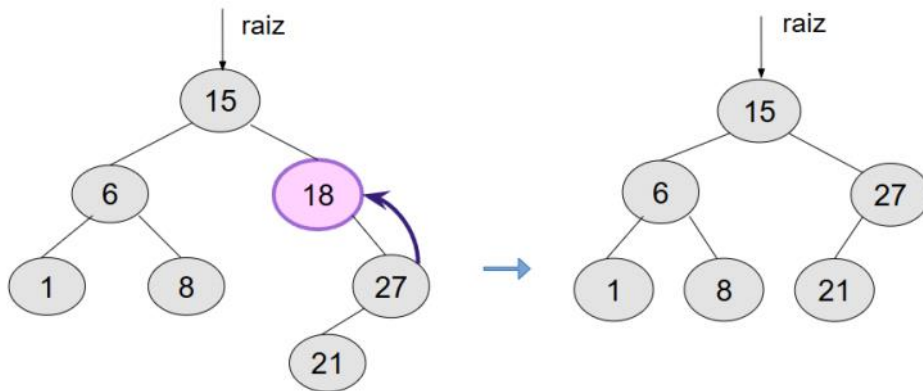


### ↳ Remoção

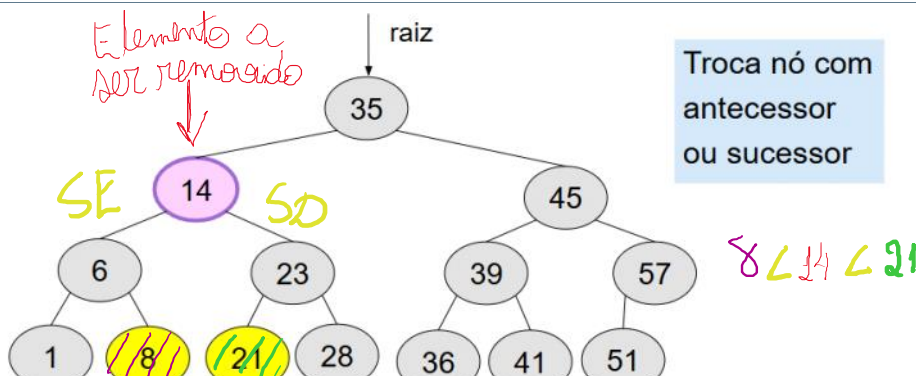
- Caso seja o único item:



- Caso seja uma subárvore c/ apenas um filho

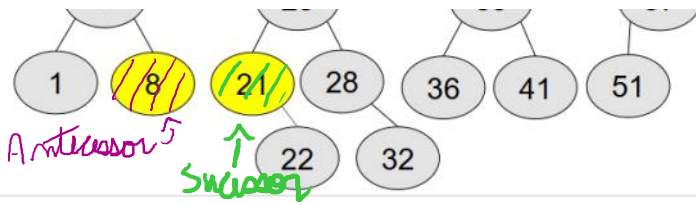


- Caso seja uma subárvore c/ 2 filhos

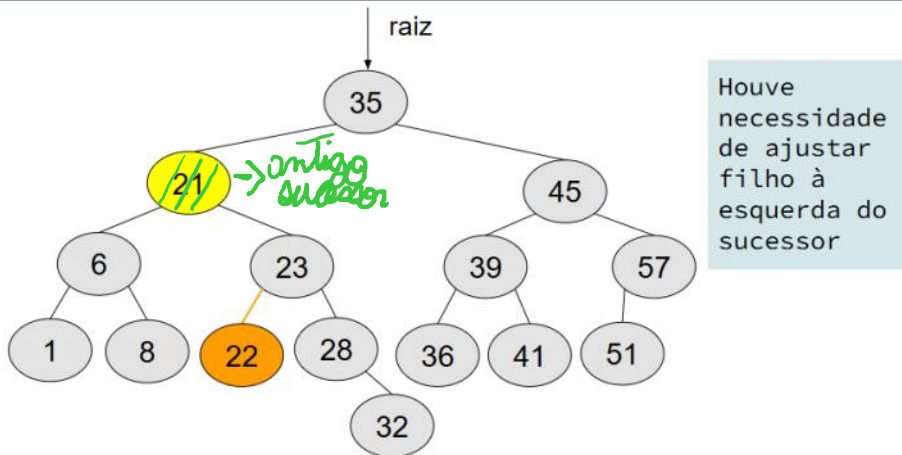


+ à direita

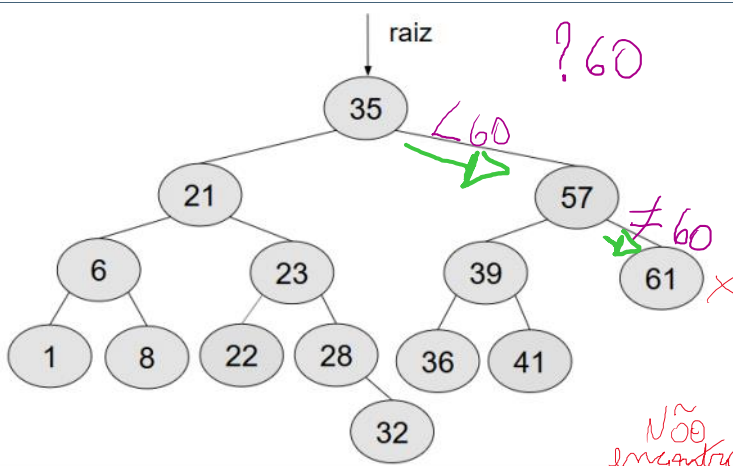
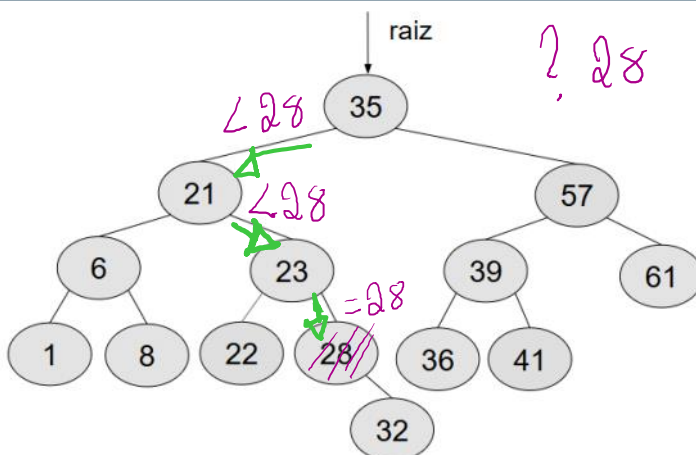
Antecessor = maior elemento da SE  
+ à esquerda  
Sucessor = menor elemento da SD



- Exemplo escolhendo o **Sucessor**:



↳ Busca



u> Mínimo e máximo

