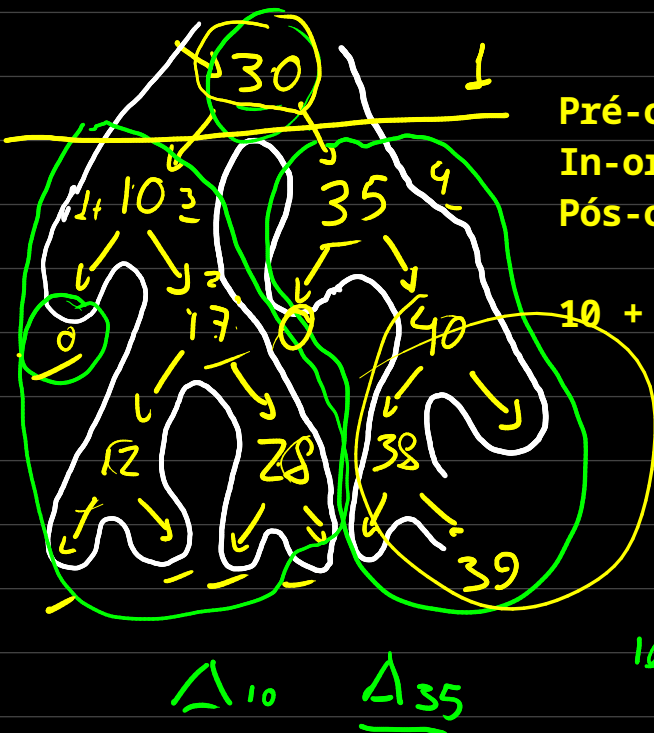


30 - 10 - 17 - 35 - 28 - 40 - 38 - 12

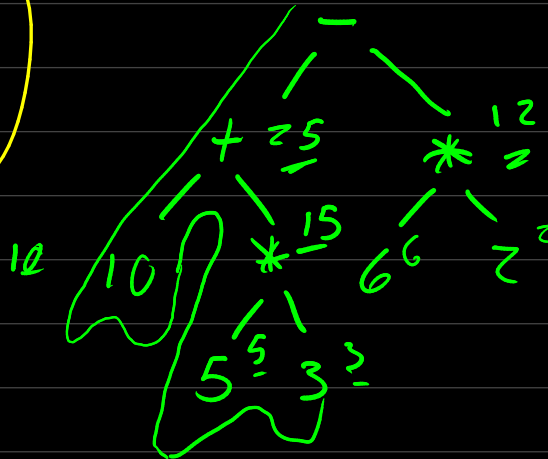


Pré-order: 30 - 10 - 17 - 12 - 28 - 35 - 40 - 38

In-order: 10 - 12 - 17 - 28 - 30 - 35 - 38 - 40

Pós-order: 12 - 28 - 17 - 10 - 38 - 40 - 35 - 30

~~A(38)~~  
~~A(A<sub>40</sub>)~~  
~~A(N)~~  
~~A(A<sub>35</sub>)~~  
~~A(A<sub>10</sub>)~~  
~~A(A<sub>30</sub>)~~



Seja T uma árvore BST:

1. T é uma árvore vazia, representada por  $\emptyset$  ou NULL
  2. T é um nó denominado raiz, representado por (k, t\_esq, t\_dir)
- Onde,

k é uma chave (dado, item ou valor) armazenado

t\_esq é uma BST, na qual todos os elementos são menores que k

t\_dir é uma BST, na qual todos os elementos são maiores que k

Função 1 - Quantidade de Elementos

1.  $\leftarrow 0$
2.  $1 \text{ (raiz)} + \text{quantidade}(t\_esq) + \text{quantidade}(t\_dir)$

Função 2 - Somatório do valor dos nós

1. 0
2.  $k + \text{somatorio}(t\_esq) + \text{somatorio}(t\_dir)$

Função 3 - Altura da árvore

1. 0
2.  $1 + \underline{\text{maximo}(\text{altura}(t\_esq), \text{altura}(t\_dir))}$

## Função Inserir (v)

1. criar a raiz

2. Comparar (v e k) - Caso  $v > k$

$t\_dir = \text{inserir}(t\_dir, v)$

Caso contrário

$t\_esq = \text{inserir}(t\_esq, v)$

