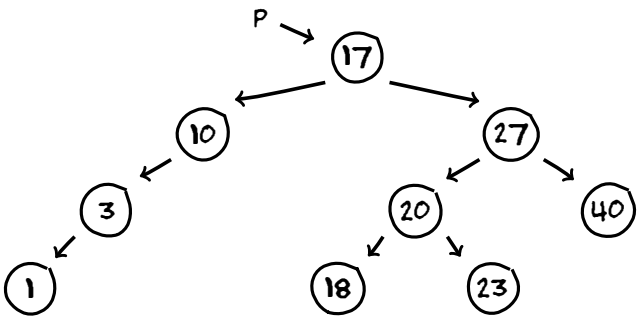


Contando os elementos e calculando a altura

Imagine que o ponteiro p aponta para uma árvore binária de busca



A tarefa é

→ Contar a quantidade de elementos da árvore

Aqui, o raciocínio recursivo também é bem fácil

- 1. primeiro a gente conta os elementos na subárvore esquerda
— (fazendo uma chamada recursiva)
- 2. depois a gente conta os elementos na subárvore direita
— (fazendo uma chamada recursiva)
- 3. e daí, a gente soma 1 para contar a raiz

A coisa fica assim

```
func contaElArv-Rec (q)

    Se ( q == Nulo )    Retorna (0)

    c1 = contaElArv-Rec (q->esq)
    c2 = contaElArv-Rec (q->dir)

    cont = c1 + c2 + 1

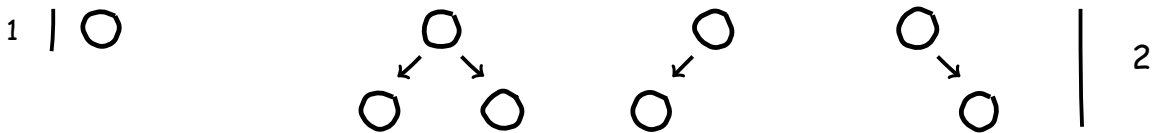
    Retorna (cont)
```

Agora, imagine que a tarefa é

→ Calcular a altura da árvore

Bom, a gente imagina que a árvore que só tem a raiz possui altura 1.

E que a árvore que tem dois níveis tem altura 2



— (e que o ponteiro para Nulo tem altura 0)

Daí, cada nível a mais aumenta a altura de 1.

Com essa definição, o raciocínio recursivo fica assim

- 1. primeiro a gente calcula a altura da subárvore esquerda
— (fazendo uma chamada recursiva)
- 2. depois, a gente calcula a altura da subárvore direita
— (fazendo uma chamada recursiva)
- 3. daí, a gente pega o máximo entre os dois e soma 1
— (para contar o nível da raiz da árvore)

A coisa fica assim

```
func altArv-Rec (q)

    Se ( q == Nulo )    Retorna (0)

    a1 = altArv-Rec (q->esq)
    a2 = altArv-Rec (q->dir)

    alt = Max {a1, a2} + 1

    Retorna (alt)
```