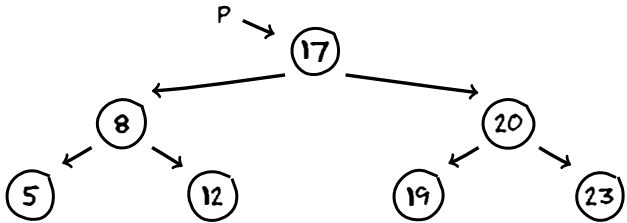


Remoção de uma folha

A remoção de um elemento na árvore binária de busca é uma operação um pouquinho complicada — *(porque?)*

Por isso, nós vamos fazer alguns exemplos simples antes de apresentar o caso geral.

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



A tarefa é

- *remover a folha `k` da árvore*
 - *(se `k` não é uma folha, a função não faz nada)*

Daí, o raciocínio fica assim

- primeiro a gente faz a busca pelo elemento `k` — *(descendo recursivamente pela árvore)*
- daí, se o `k` é uma folha, a gente retorna
 - `Nulo`, para indicar que não há mais nada ali
 - e um ponteiro para o elemento que foi removido — *(que vai até a `main()`)*
- e se `k` não é uma folha, a gente retorna
 - um ponteiro para esse elemento, para indicar que ele continua ali
 - e `Nulo`, para indicar que ele não foi removido

A coisa fica assim

```
func RFA-Rec (q, k)                                     // Remove Folha da Árvore

    Se (q == Nulo)   Retorna (Nulo, Nulo)                // não encontrei ...

    Sse (q->num == k)
        Se (q->esq == Nulo E q->dir == Nulo)             // eu sou uma folha?
            Retorna (Nulo, q)
        Senão
            Retorna (q, Nulo)

    Sse (q->num > k)
        fi = RFA-Rec (q->esq, k)                         // descida na árvore
        q->esq = fi                                       // reconexão

    Senão
        fi, folha = RFA-Rec (q->dir, k)                  // descida na árvore
        q->dir = fi                                       // reconexão

    Retorna (q, folha)
```

E a função `main()` fica assim

```
main ()

    p, folha = RFA-Rec (p, 12)

    Se (folha != Nulo)   free (folha)
```