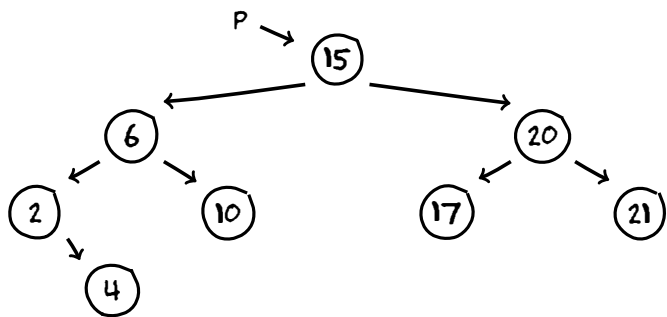


Remoção do menor elemento

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

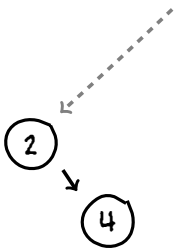


A tarefa é

→ *Remover o menor elemento da árvore*

Bom, a gente chega no menor elemento da árvore descendo pela esquerda até não poder mais

E daí, a gente vê que o menor elemento da árvore não precisa ser uma folha



Mas, a sua remoção não é difícil.

Porque o filho direito pode ser colocado no seu lugar — *(se houver um filho direito)*

Quer dizer,

1. primeiro a gente alcança o menor elemento — *(descendo sempre pela esquerda)*
2. e daí, a gente retorna
 - um ponteiro para o seu filho direito — *(para a reconexão)*
 - um ponteiro para o menor elemento — *(que vai chegar até a `main()`)*

A coisa fica assim

```
func RMA-Rec (q,k)                                     // Remove o Menor da Árvore

    Se (q == Nulo)   Retorna (Nulo,Nulo)               // árvore vazia

    Sse (q->esq == Nulo)                                // eu sou o menor elemento?
        fd = q->dir
        q->dir = Nulo
        Retorna (fd,q)

    Senão
        fi,menor = RMA-Rec (q->esq)                    // descida na árvore
        q->esq = fi                                     // reconexão
        Retorna (q,menor)
```