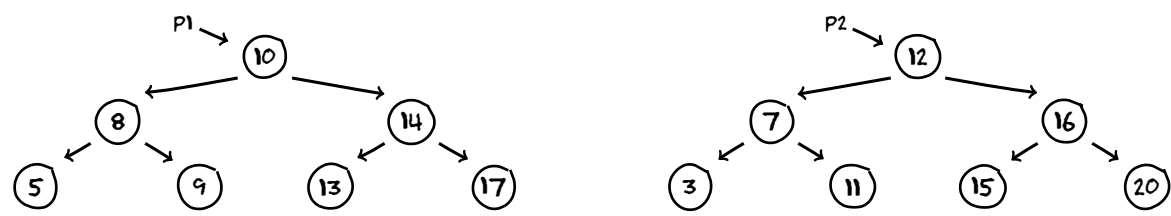


Intercalação de árvores

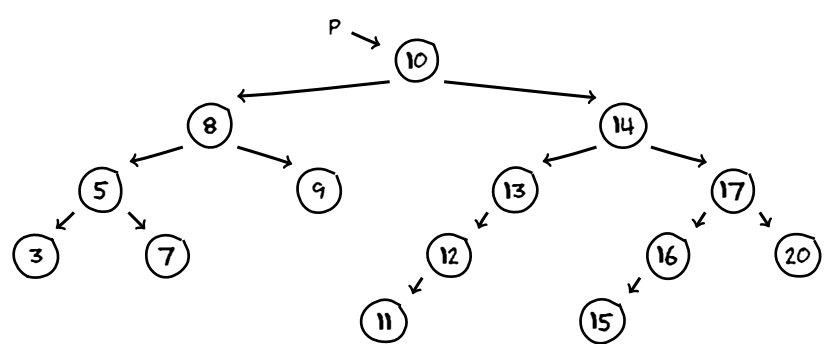
Imagine que os ponteiros p1, p2 apontam para duas árvores binárias de busca



A tarefa é

→ produzir uma única árvore binária de busca com os elementos de p1 e p2

No exemplo acima, isso nos daria



Legal.

Mas, como é que a gente faz isso?

Bom, faz sentido escolher a raiz de uma das duas árvores para ser a raiz da árvore final.

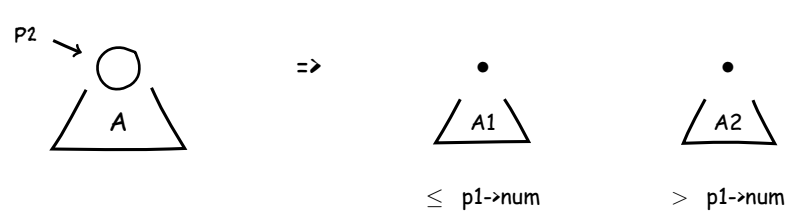
Nesse raciocínio, nós vamos escolher a raiz de p1.

Daí, é bem fácil ver que

- os elementos da subárvore esquerda de p1 vão ficar do lado esquerdo
- os elementos da subárvore direita de p1 vão ficar do lado direito

Mas, o que é que a gente faz com p2?

Bom, faz sentido particionar p2 utilizando a raiz de p1



Daí,

- a primeira parte vai ficar do lado esquerdo
- a segunda parte vai ficar do lado direito

E agora basta

- intercalar as duas subárvores que vão ficar do lado esquerdo
- intercalar as duas subárvores que vão ficar do lado direito

A coisa fica assim

```
func intercalação-Rec (p1, p2)

  Se (p1 == Nulo)  Retorna ( p2 )

  Se (p2 == Nulo)  Retorna ( p1 )

  Senão
    raiz = p1
    a1 = p1->esq;  a2 = p1->dir
    b1,b2 = partição-Rec (p2, p1->num)

    raiz->esq = intercalação-Rec (a1, b1)
    raiz->dir = intercalação-Rec (a2, b2)

  Retorna (raiz)
```