

A operação de intercalação

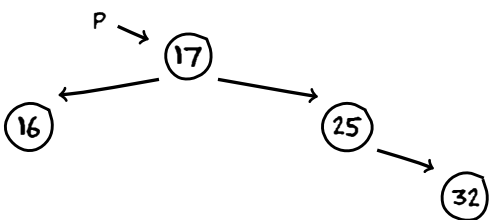
Imagine que os ponteiros p1 p2 apontam para duas árvores binárias de busca



A tarefa é

- construir uma árvore que contém uma cópia dos elementos que estão em p1 e p2
- (sem modificar as árvores p1 e p2)

No exemplo acima, isso nos daria



Certo.

Mas, como é que a gente faz isso?

Bom, o caso mais fácil é aquele onde as raízes de p1 e p2 são iguais.

Porque daí, nós criamos uma cópia da raiz.

E fazemos as interseções dos dois lados para completar a solução.

Agora, imagine que a raiz de p1 é maior do que a raiz de p2.

Nesse caso, nós demontamos a árvore apontada por p2 da seguinte maneira



Daí, nós fazemos a interseção da primeira parte de p2 com o lado esquerdo de p1.

E fazemos a interseção da segunda parte de p2 com a árvore p1 inteira — (porque podem haver elementos de B2 que estão em A1)

E daí, nós juntamos as duas árvores resultantes.

A coisa fica assim

```
func interseção-Rec (q1,q2)

  Se (q1 == Nulo OU q2 == Nulo)   Retorna ( Nulo )

  Sse (q1->num == q2->num)
    r = Novo (RegInt)
    r->num = q1->num
    r->esq = interseção-Rec (q1->esq,q2->esq)
    r->dir = interseção-Rec (q1->dir,q2->dir)
    Retorna (r)

  Sse (q1->num > q2->num)
    b1 = q2;  b2 = q2->dir
    q2->dir = Nulo
    a1 = interseção-Rec (q1->esq,b1)
    a2 = interseção-Rec (q1,b2)
    r = juntaArv (a1,a2)
    Retorna (r)

  Senão
    // análogo ao caso anterior ...
```