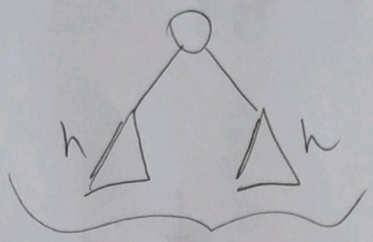
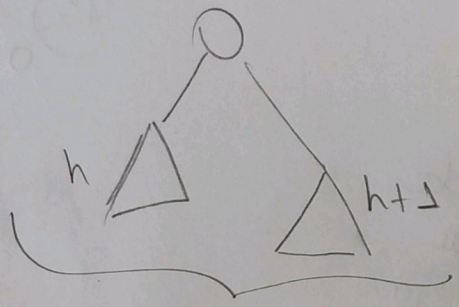
- **direita - esquerda = h**

- se h = 0



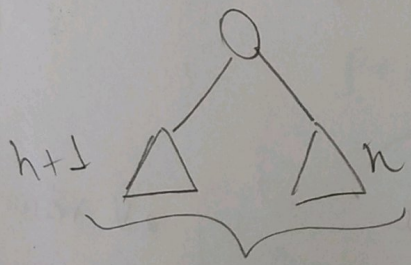
mesmo tamanho

- se h = +1



+1, subárvore da direita é maior

- se h = -1



-1, subárvore da esquerda é maior

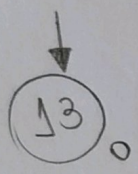
AVL Tree

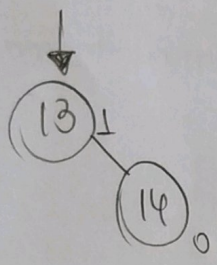
AVL = Adelson-Velsky and Landis } 2 caras

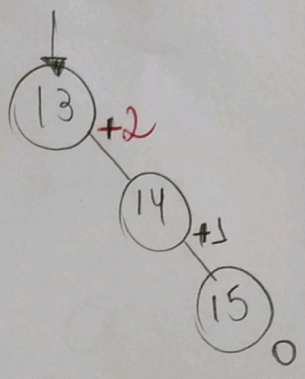
→ Sequência de inserções

M N O L K Q P H I A

→ 13, 14, 15, 12 11, 17, 16 8 9 1

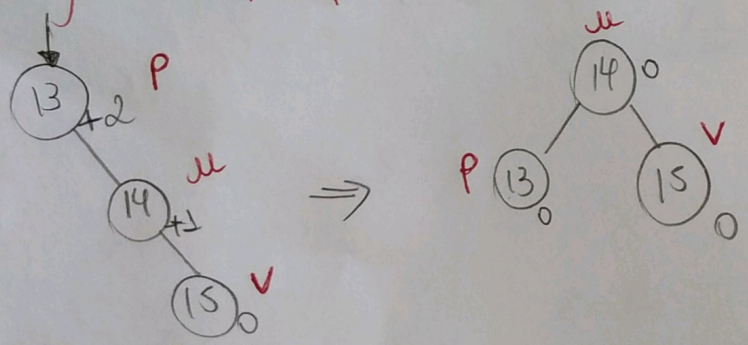


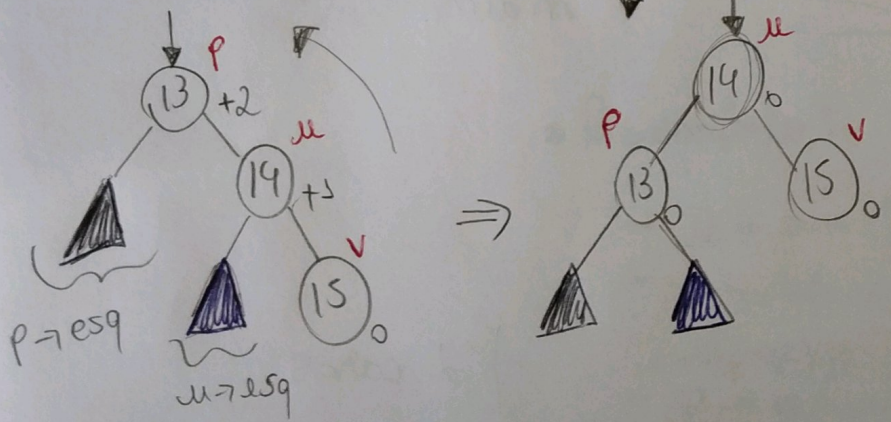




rotação simples para esquerda

→ rotação simples para esquerda





- V não é alterado

u = p → direita

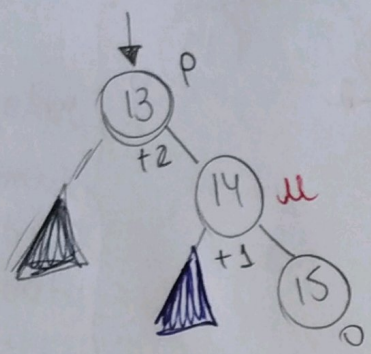
p → dir = u → esq

u → esq = p

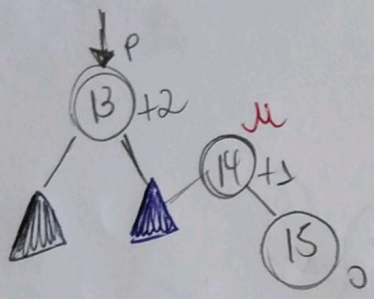
p → balance = 0

p = u

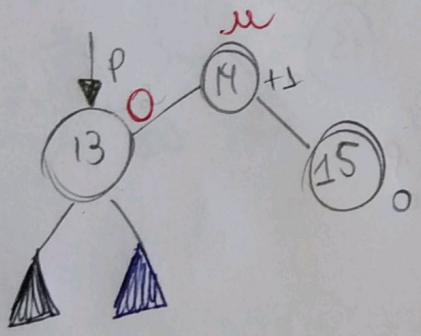
→ rotação passo a passo



$u = p \rightarrow direita$

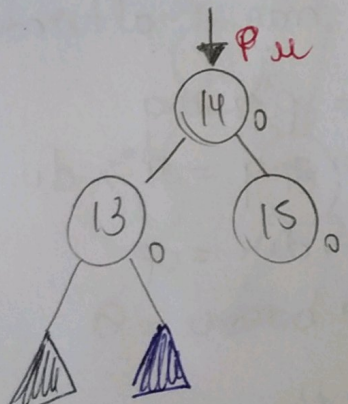


$p \rightarrow dir = u \rightarrow esq$



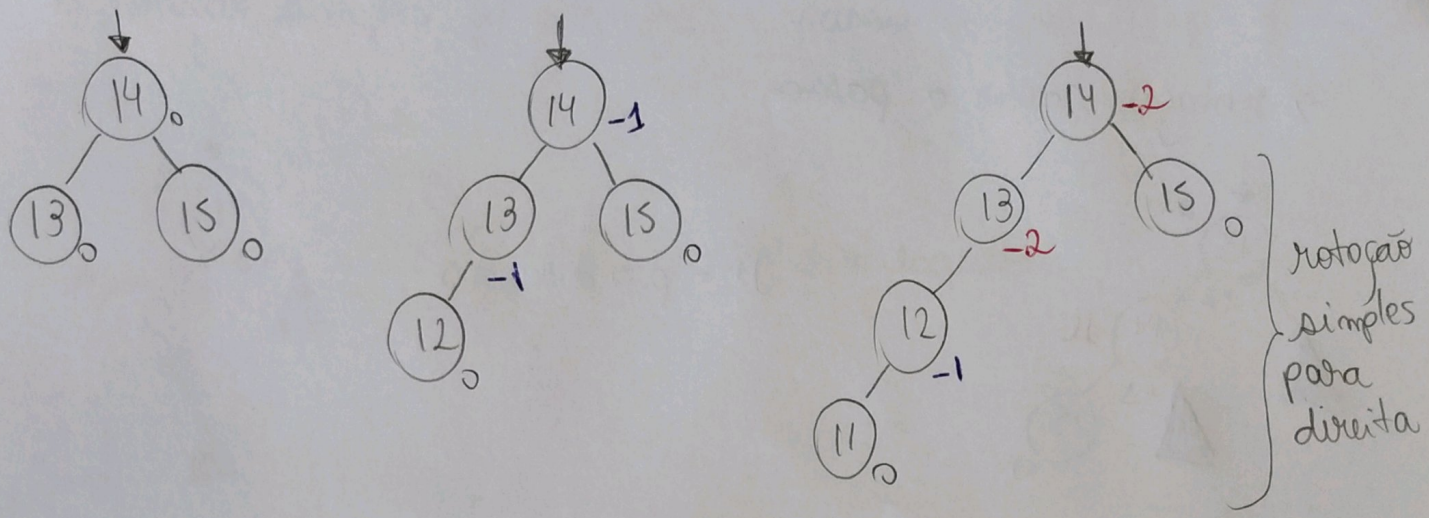
$u \rightarrow esq = p$

$p \rightarrow balance = 0$

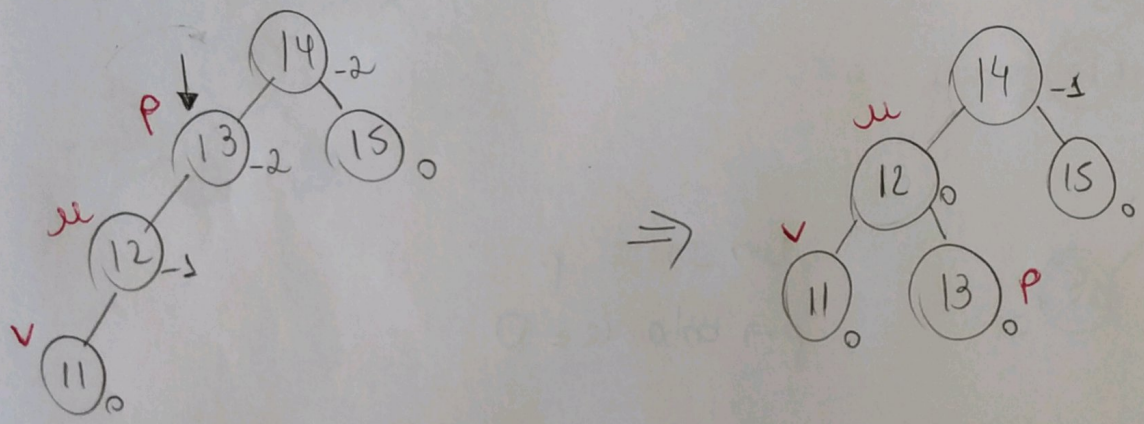


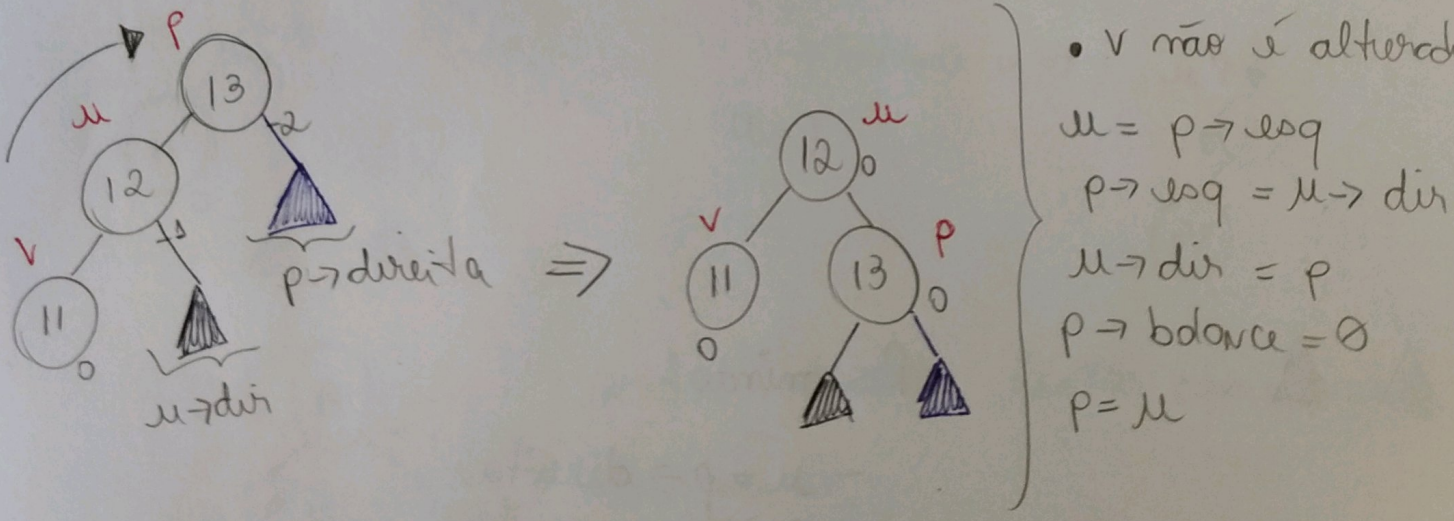
$p = u$

- Resumindo:
  - $u = p \rightarrow direita$
  - $p \rightarrow dir = u \rightarrow esq$
  - $u \rightarrow esq = p$
  - $p \rightarrow balance = 0$
  - $p = u$



→ rotação simples para direita





- v não é alterado

$u = p \rightarrow esq$

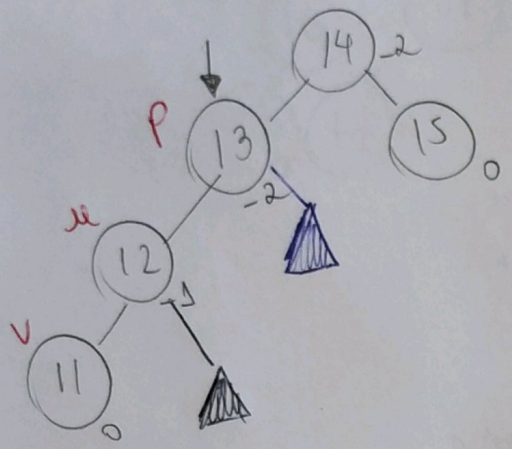
$p \rightarrow esq = u \rightarrow dir$

$u \rightarrow dir = p$

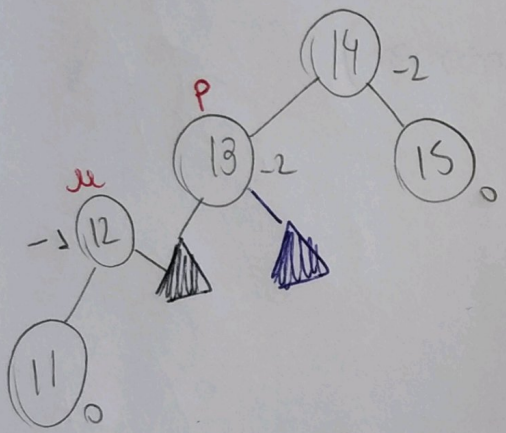
$p \rightarrow bdance = 0$

$p = u$

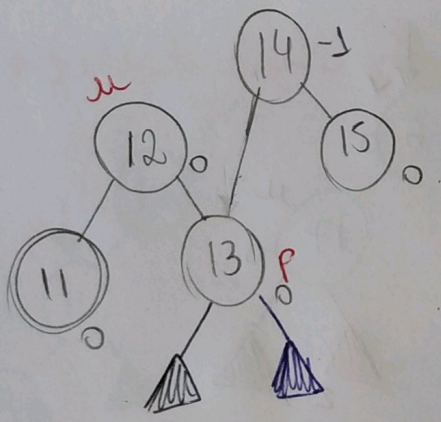
→ rotação simples p/ direita, passo a passo:



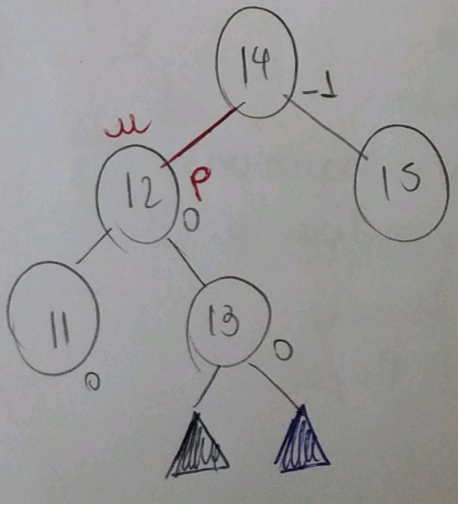
u = p → esquerda;



p → esq = u → direita;



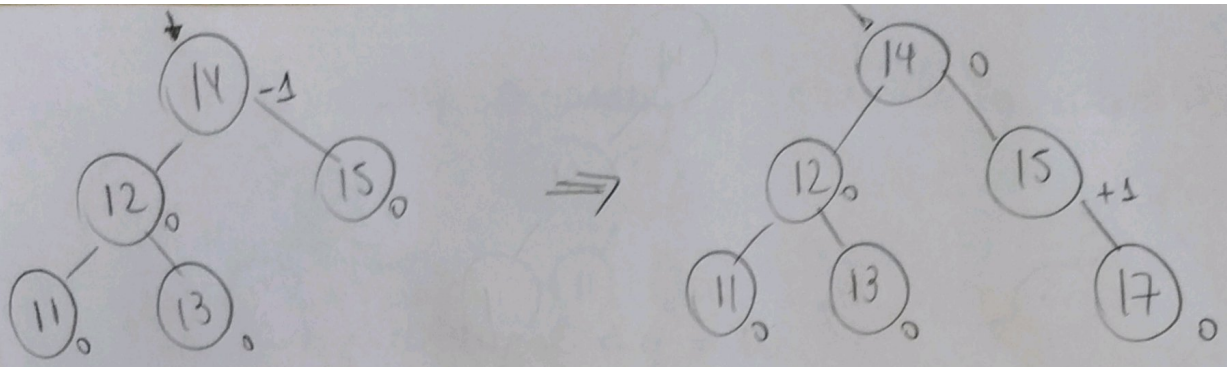
u → direita = p;
p → balance = 0;

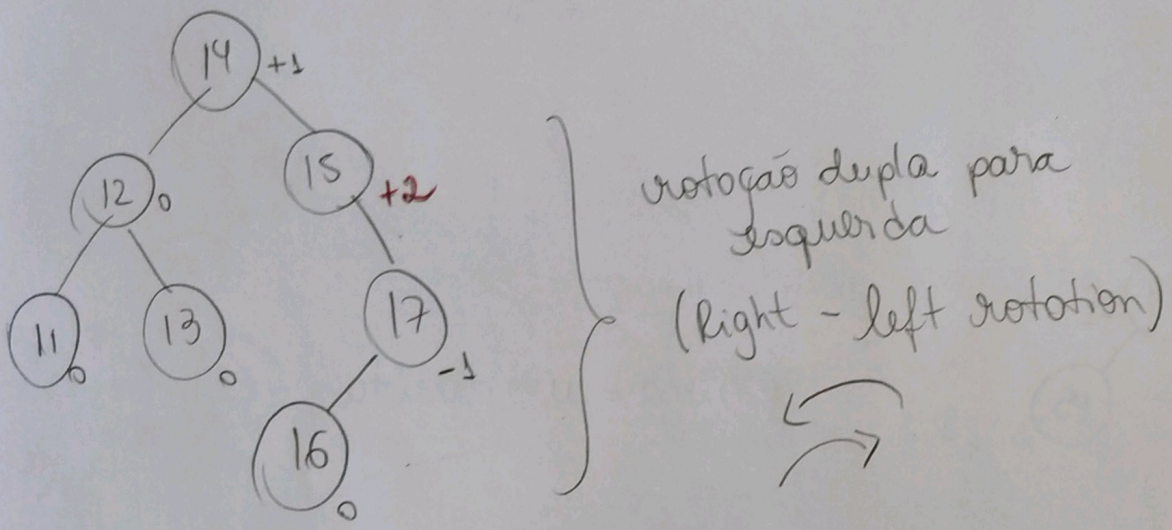


p = u } permite linkar com o pai

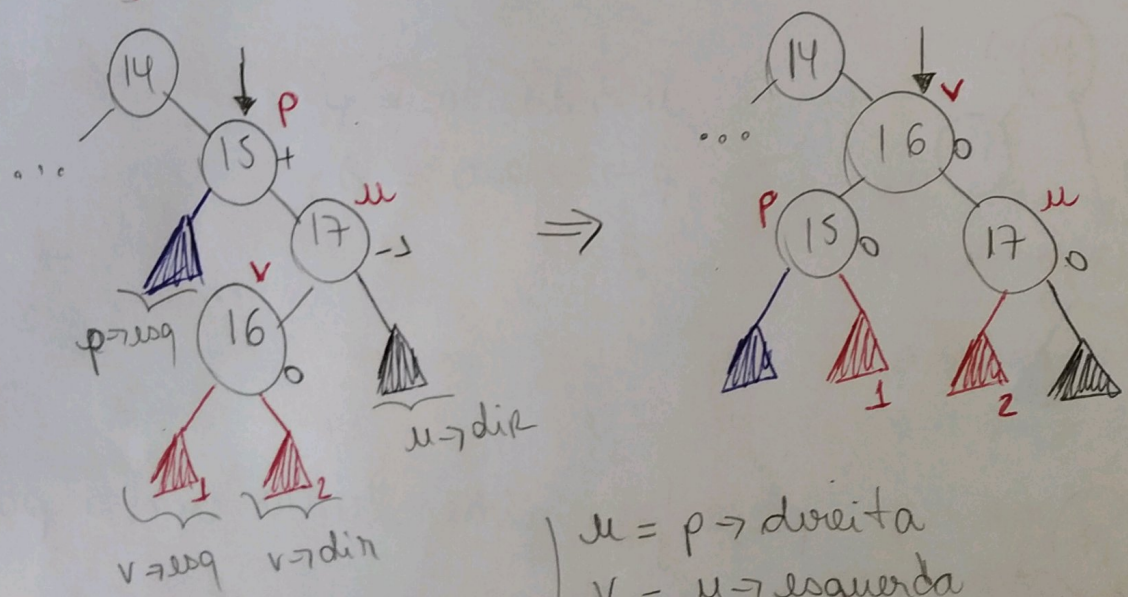
:Resumindo

- u = p → esquerda
- p → esquerda = u → direita
- u → direita = p;
- p → balance = 0;
- p = u





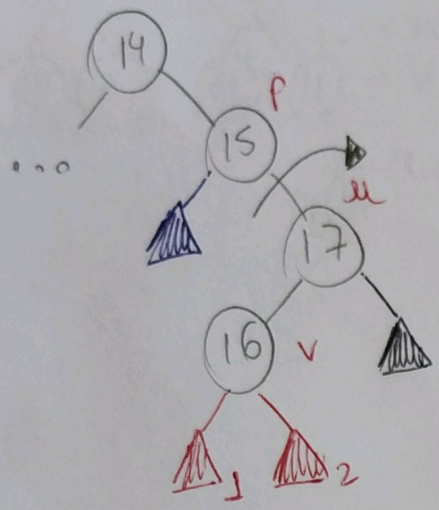
* rotação dupla para esquerda



atualizar:
- p→bolonce
- v→bolonce
- u→bolonce

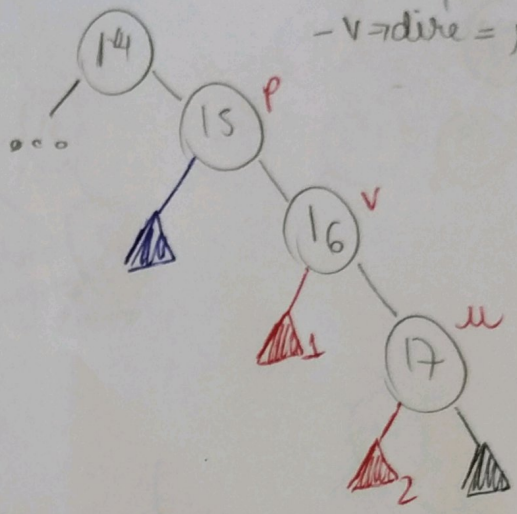
u = p→direita
v = u→esquerda
p→direita = v→esquerda
u→esquerda = v→direita
v→direita = u
v→esquerda = p
p = v

Passo a passo:

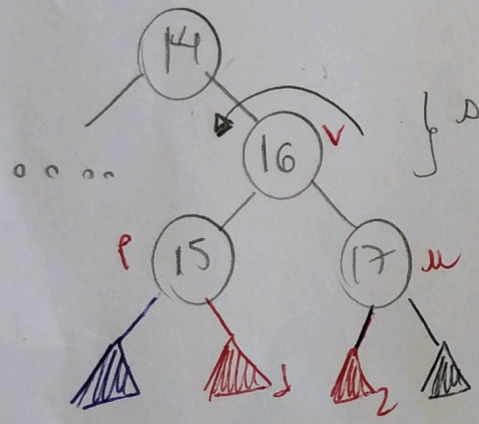


} primeira rotação para direita

- u = p->dire
- v = u->esq



- u->esq = v->direit
- v->dire = u



} segunda rotação para esquerda

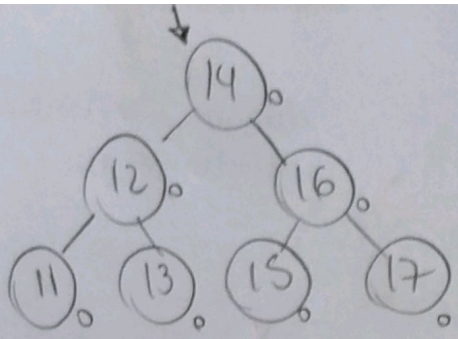
- p->dir = v->esq
- v->esq = p

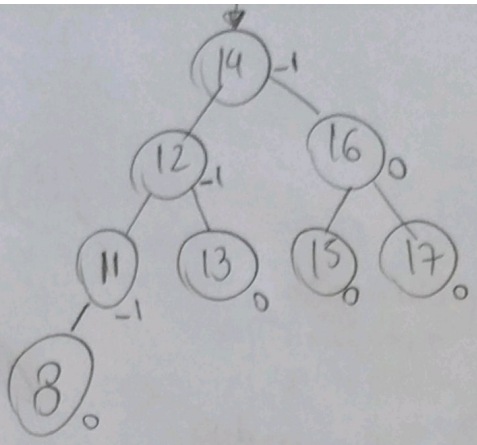
atualiza desbalanceamento:

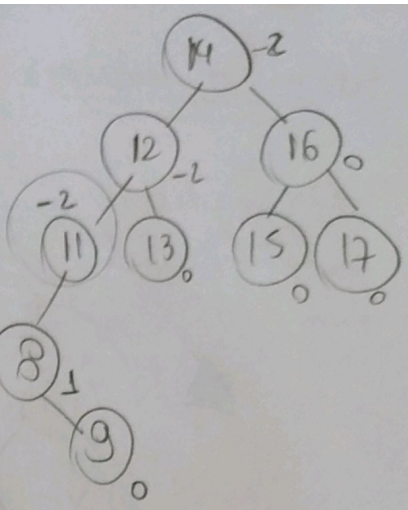
- p
- v
- u

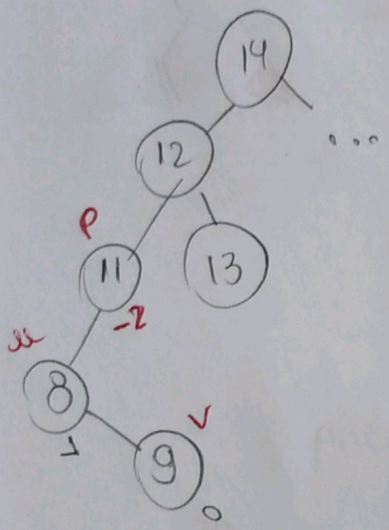
final:

p = v;



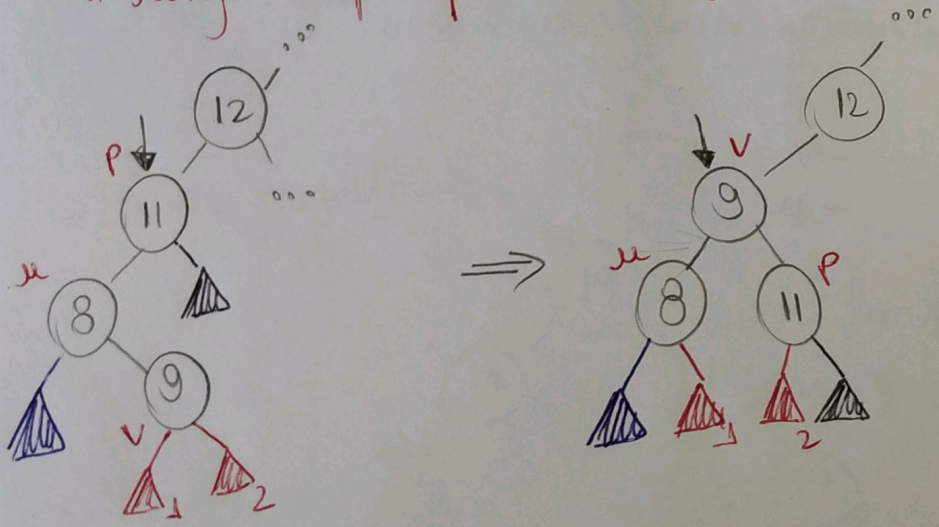






rotação dupla à direita
→ esquerda (8)
→ direita (9)

* rotação dupla para direita



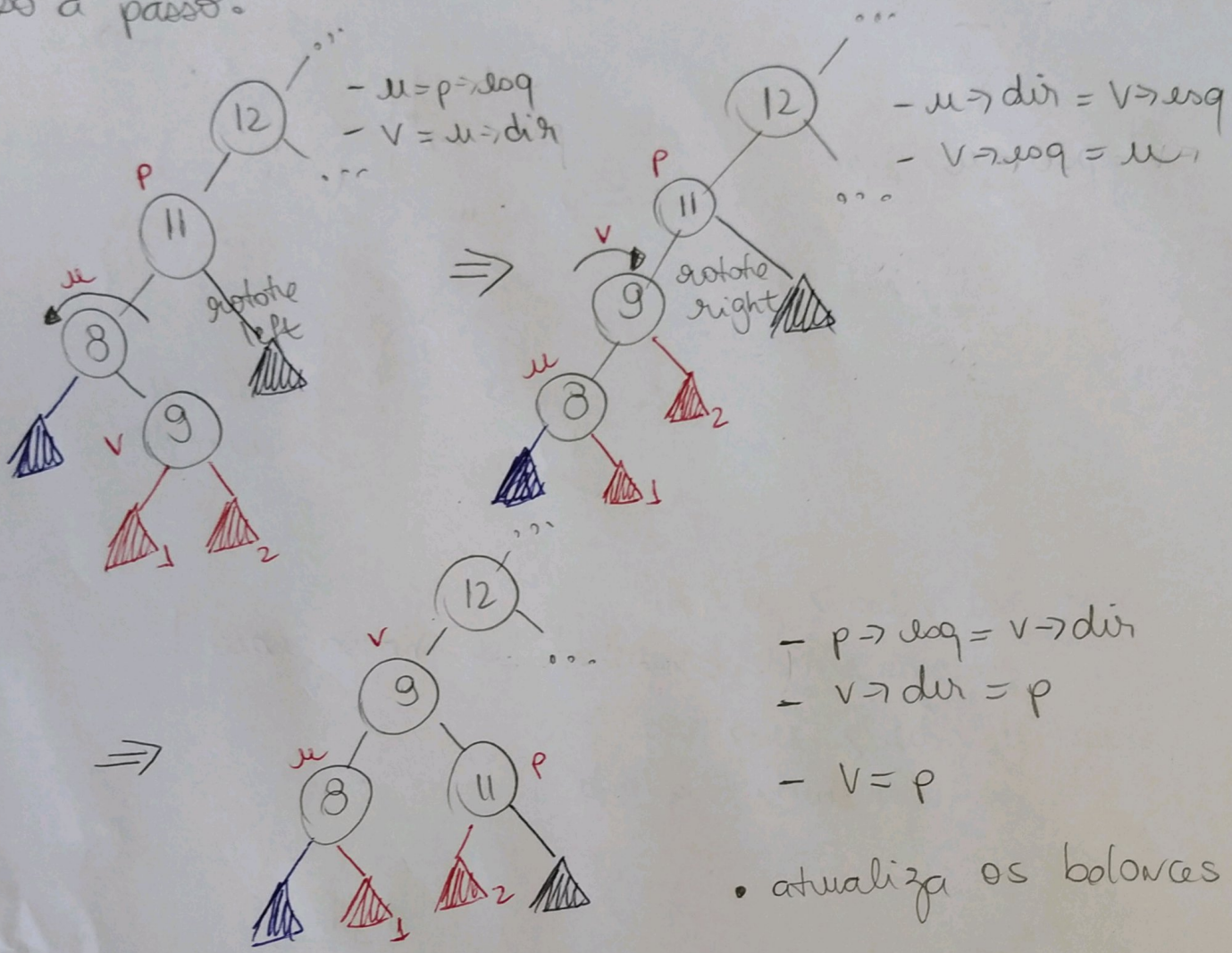
u = p → esq
v = u → dir

u → dir = v → esq
v → esq = u

p → esq = v → dir
v → dir = p
p = v } retornar a ref correta na recursão

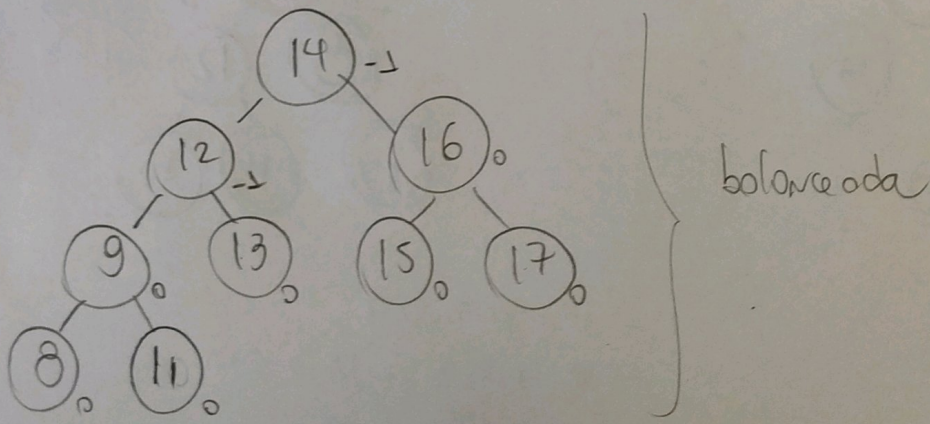
* atualizar os balances de (p, u, v) depois das rotações

Passo a passo:

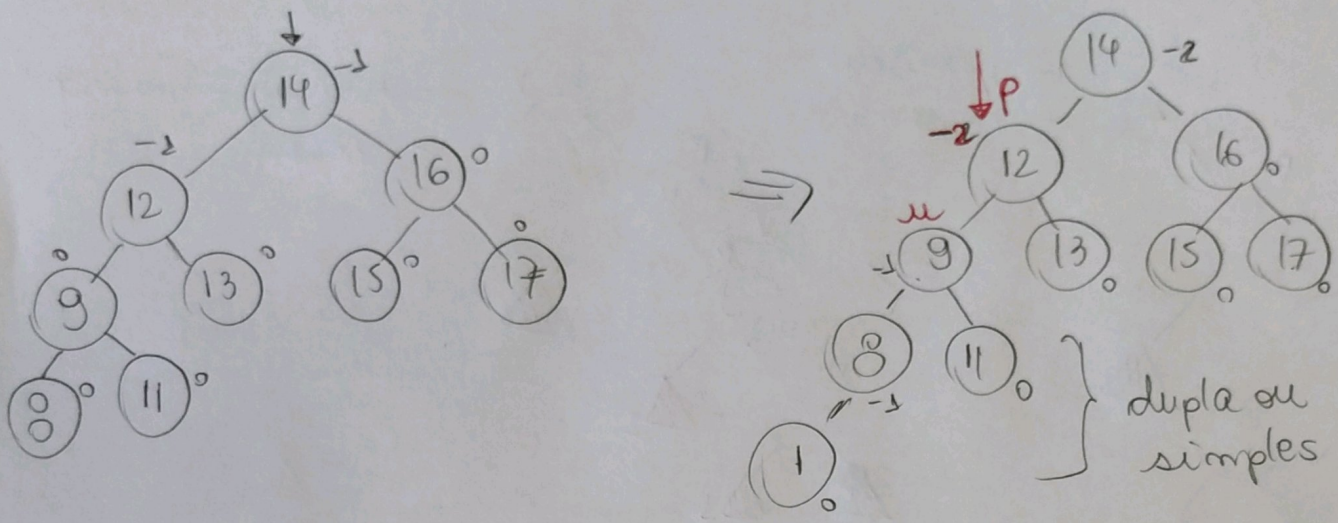


• atualiza os balances

→ árvore resultante:

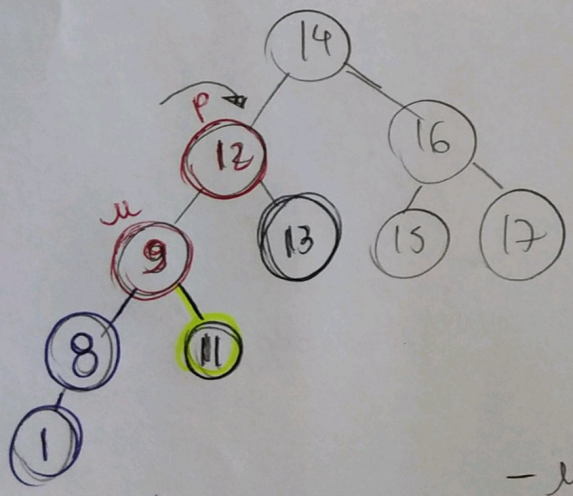


→ inserindo 1:

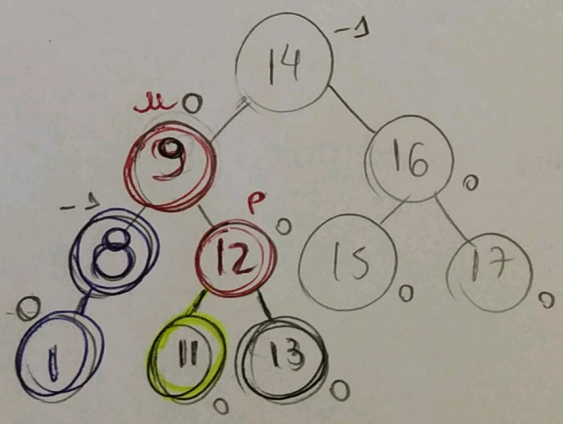


- se $u \to esq > u \to dir$
  rotação simples Esquerda * nosso caso!
- se $u \to dir > u \to esq$
  rotação dupla Esquerda

antes



depois



- $u = p \to esq$
- $p \to esq = u \to dir$
- $u \to dir = p$
- $p = u$