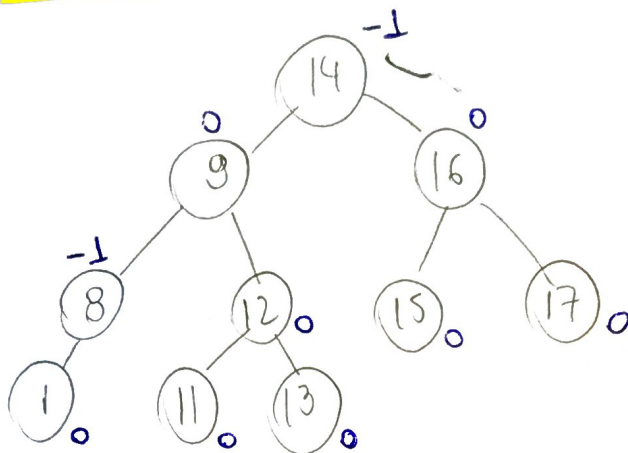


→ Remoção em AVLs



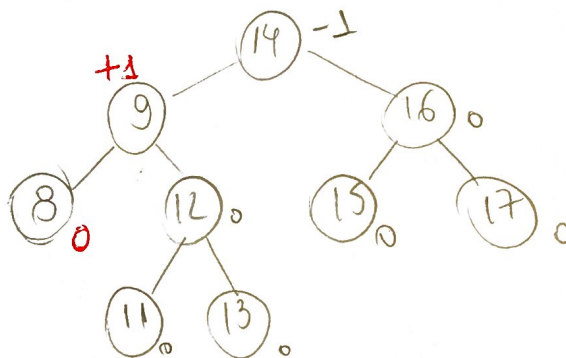
• hipótese:

→ remover um nó não altera o balanceamento de seus subárvores

Método: substituir o nó pelo maior filho da sub-árvore direita

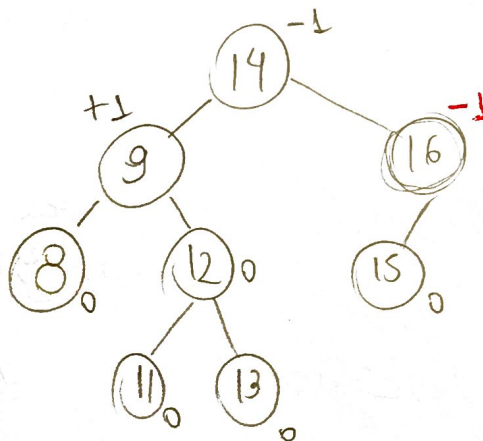
→ remover ①:

• não altera o balanceamento da árvore



→ remover ①⑦:

• não altera o balanceamento da árvore

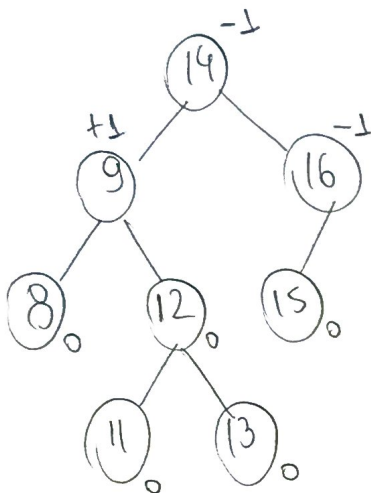


Condições que:

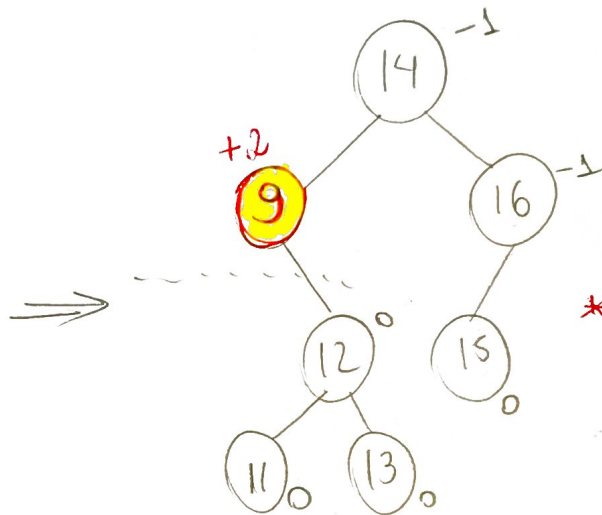
* remoção de folhas segue a mesma lógica dos árvores binárias

• remover 8

antes



depois



* Requer rotação p/ esquerda

→ Perceber que:

* o desbalanceamento ocorre sempre p/ cima na árvore

* os nós dos sub-árvores balanceadas continuarão balanceados

→ Critério da rotação na inserção

Se $h_d - h_e \geq 2$

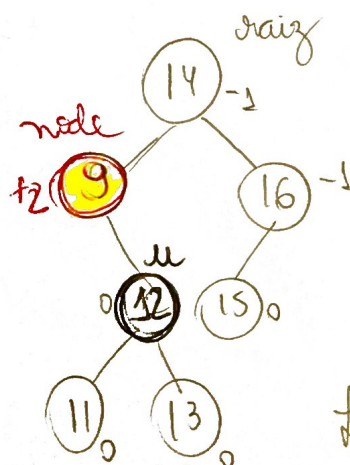
→ rotação para esquerda

Se $h(u)_d > h(u)_e$

→ Rotação Simples Esquerda

Senão

→ Rotação Dupla Esquerda

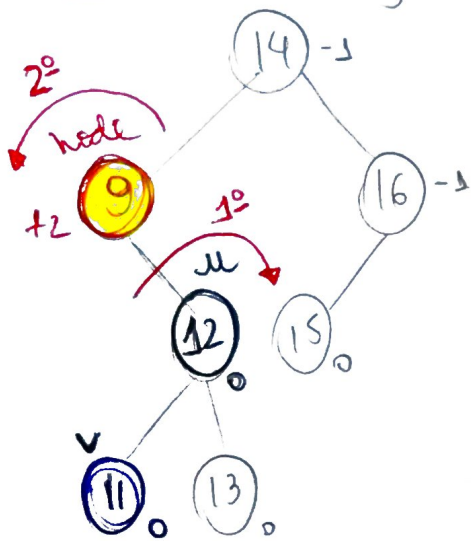


$h(u)_d = 1$

$h(u)_e = 1$

Logo: Rotação Dupla

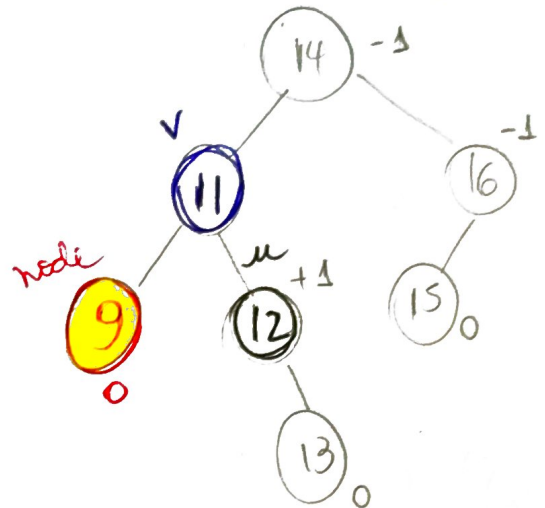
antes da rotação



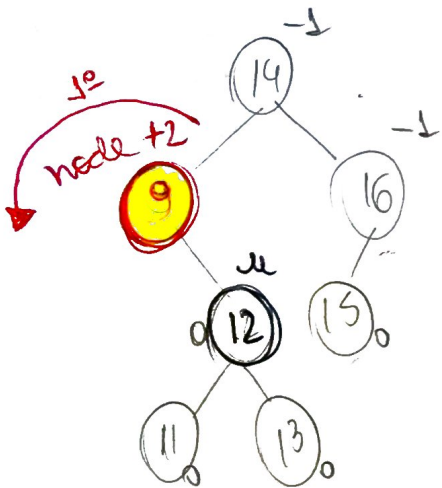
rotação dupla



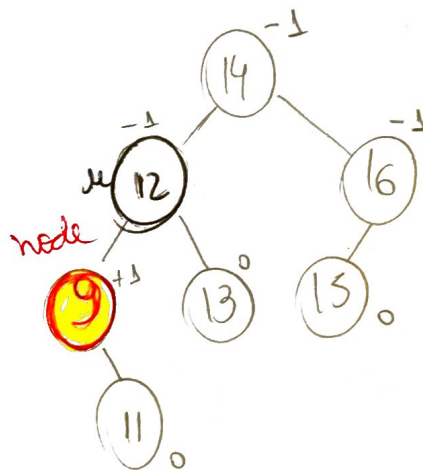
depois da rotação



E se: tivéssemos feito uma rotação simples?



rotação simples

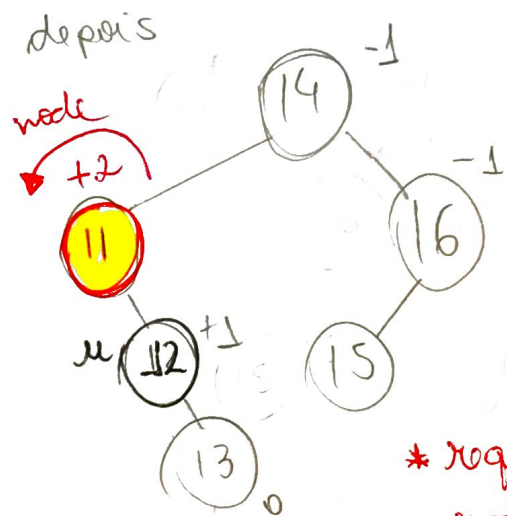
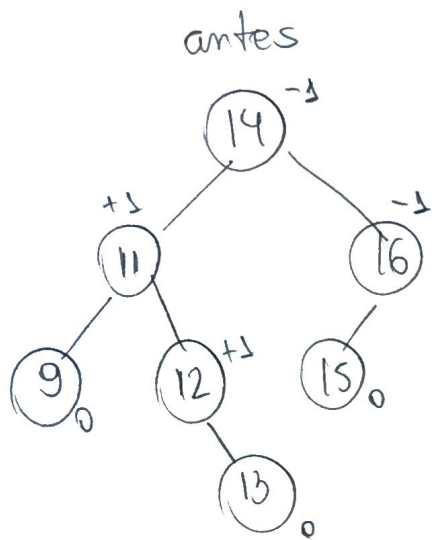


O que houve?

O que podemos inferir?

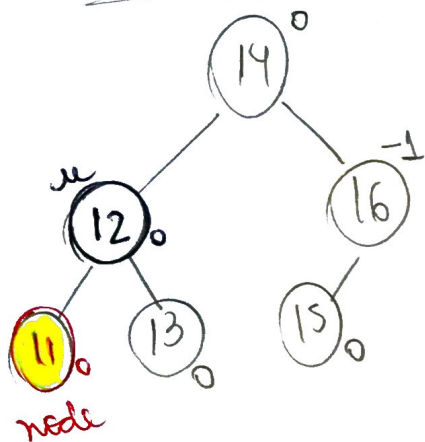
quando $h(u)_d == h(u)_e$, tanto faz.
Só precisamos fixar um critério

• Remover (9):

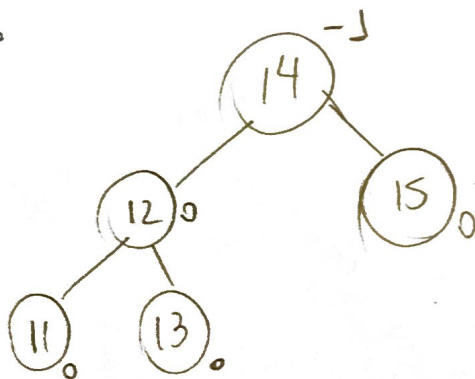


* requer
rotação
simples p/
esquerda

depois da rotação



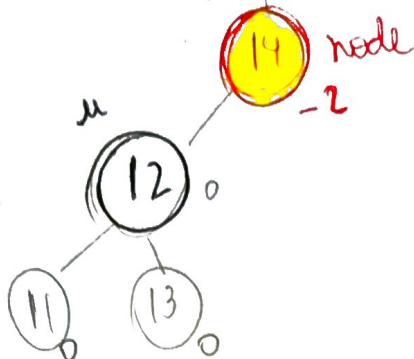
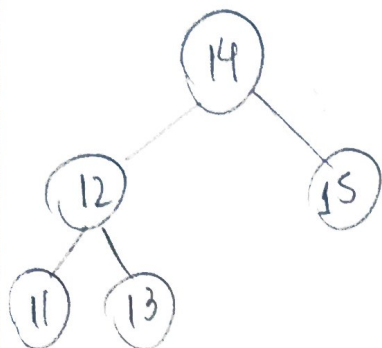
• Remover (16):



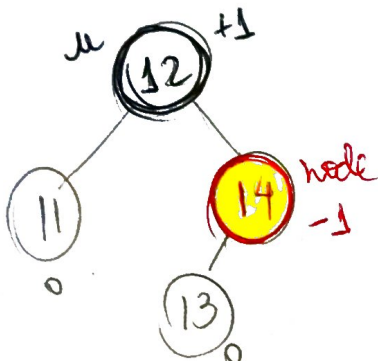
• Remover 15:

antes

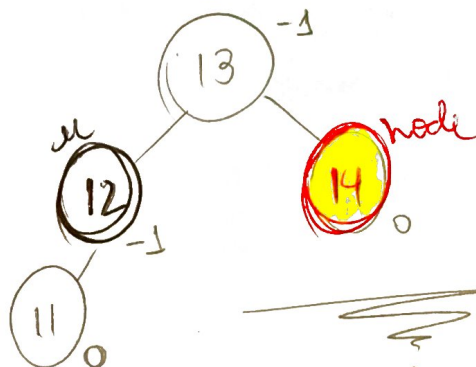
depois



rotação simples
direita



rotação dupla P/
direita

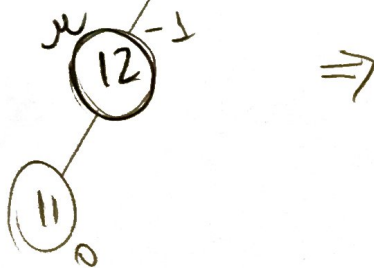
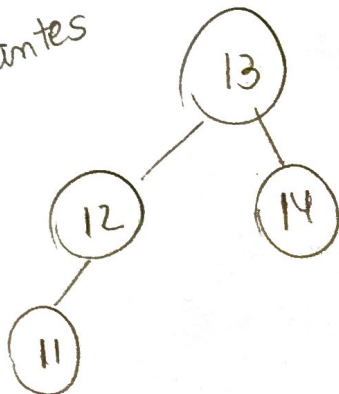


(nosso critério)

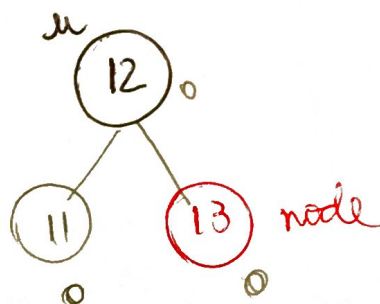
• Remover 14:

antes

depois



⇒



* Pseudocódigo de Remoção AVL (node, valor)

1. Se node é NULL
retorna falso;

2. Se $\text{node} \rightarrow \text{elemento} == \text{valor}$ // encontrei o valor

2.1 sub-árvore esquerda é nula, ou é folha
 $\text{node} = \text{node} \rightarrow \text{direita}$

2.2 sub-árvore direita é nula
 $\text{node} = \text{node} \rightarrow \text{esquerda}$

2.3 nó intermediário, nem folha nem raiz
 $\text{tmp} = \text{maior elemento da sub-árvore direita}$
// $\text{tmp} = \text{menor elemento da sub-árvore esquerda}$
 $\text{node} = \text{tmp}$
libera tmp
retorna (true);

3. Se $\text{chave} < \text{node} \rightarrow \text{elemento}$

... teste = Remoção (node → esquerda)

Senão

3. teste = Remoção (node → direita)

! fim se

4. Se teste é falso;
retorna falso;

5. $AlturaEsq = altura(node \rightarrow esq)$

6. $AlturaDir = altura(node \rightarrow dir)$

7. Se $|AlturaEsq - AlturaDir| > 1$

7.1 Aplicar Rotações (node)

8. Atualizar altura de Node

9. Retorna True.

* Aplicar Rotações (node)

1. $hd = altura(node \rightarrow direita)$

2. $he = altura(node)$

3. Se $he > hd$ // rotação p direita

3.1 $u = node \rightarrow esq$

3.2 Se $altura(u \rightarrow esq) \geq altura(u \rightarrow dir)$

3.2.1 Rotação Simples Direita (node)

Senão

3.2.2 Rotação Dupla Direita (node)

4. Serão // rotações p/ esquerda

4.1 $\mu = \text{node} \rightarrow \text{direita};$

4.2 Se $\text{altura}(\mu \rightarrow \text{dir}) > \text{altura}(\mu \rightarrow \text{esq})$

| 4.2.1 Rotação Simples Esquerda(node)

| Serão

| 4.2.2 Rotação Dupla Esquerda(node)

| fim se

| fim-se