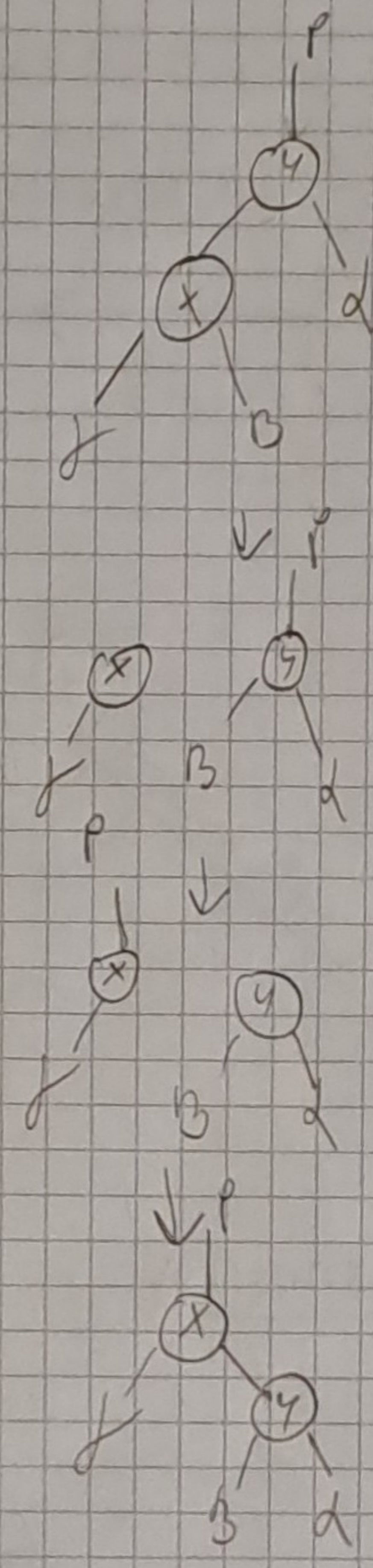


# RIGHT-ROTATE (T, x)

1.  $y = x.left$
2.  $x.left = y.right$
3. if  $y.right \neq NIL$
4.  $y.right.p = x$
5. if  $x.p == NIL$
6.  $T.root = y$
7. else if  $x == x.p.right$
8.  $x.p.right = y$
9. else  $x.p.left = y$
10.  $y.right = x$
11.  $y.p = x.p$
12.  $x.p = y$

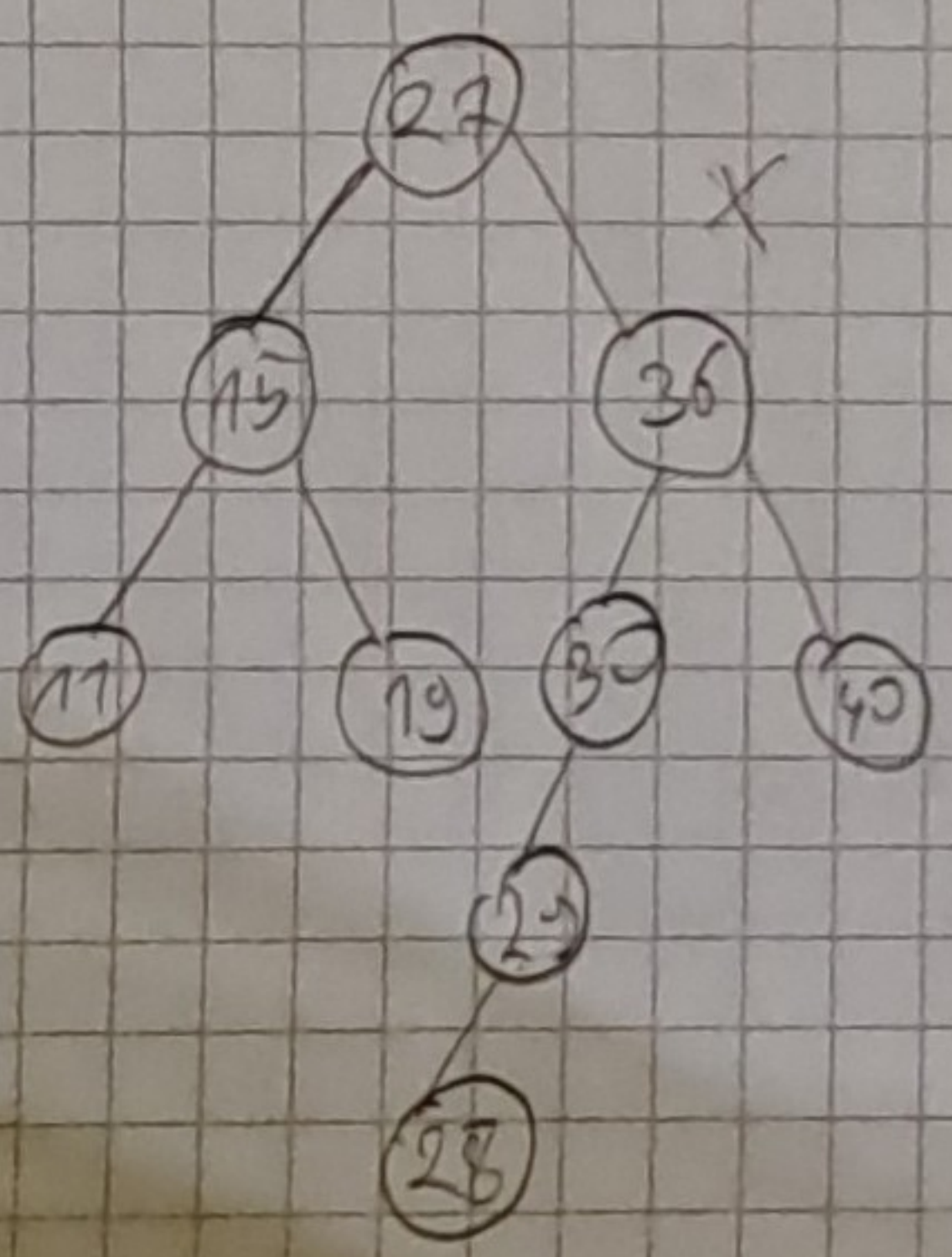


$x.right.p = y$

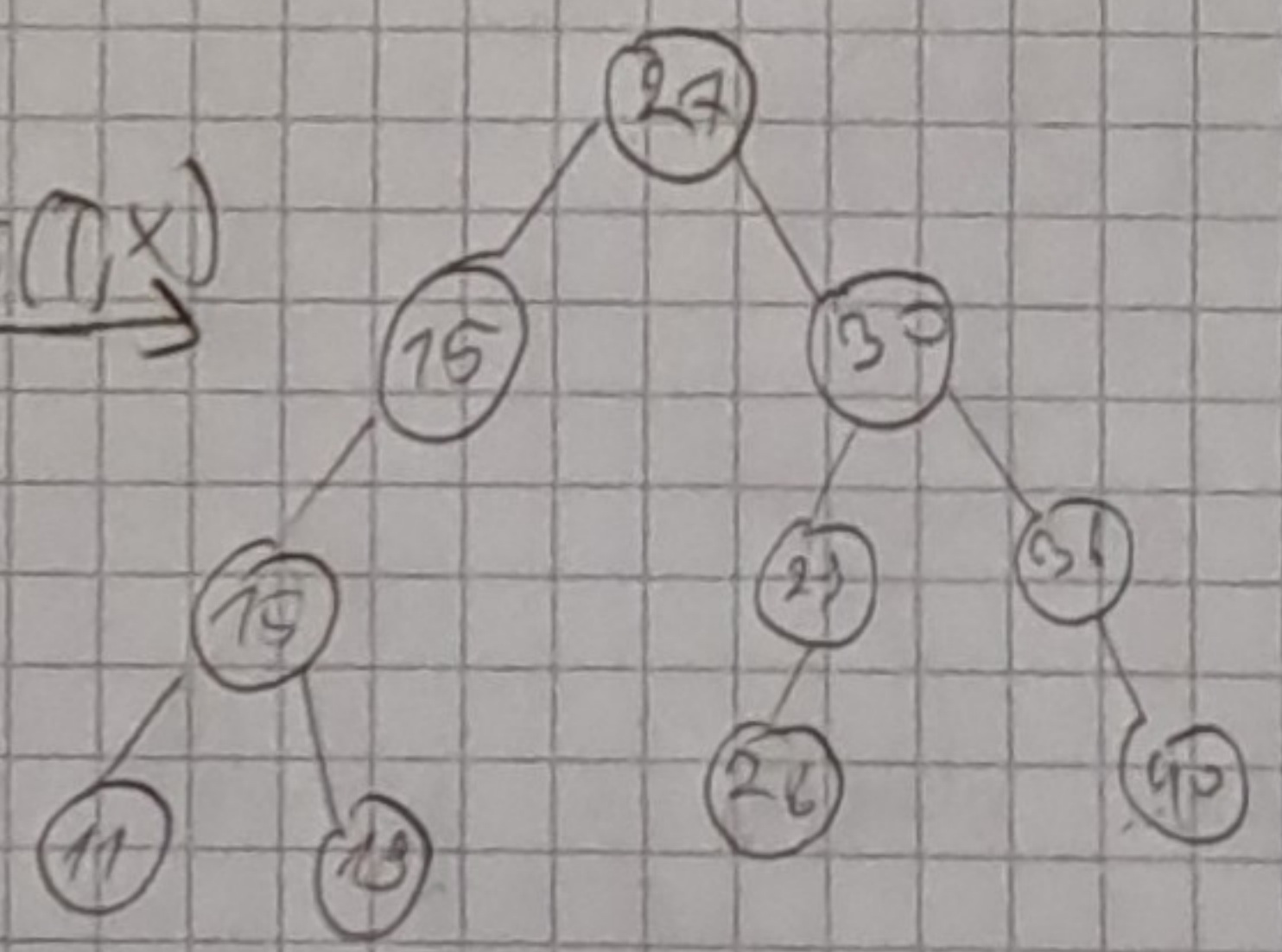
$root = x$

$x.right = y$

## PRIMER:



RIGHT-ROTATE(T, x)





2. INSERT(T, X, key)

1. if  $X == NIL$

2.  $X.key = key$

3.  $X.h = 1$

4. else if  $key < X.key$

5.  $X.left = INSERT(T, X.left, key)$

6.  $X.left.p = X$

7. else if  $key > X.key$

8.  $X.right = INSERT(T, X.right, key)$

9.  $X.right.p = X$

10. else return NIL

11.  $X.h = 1 + \max(X.left.h, X.right.h)$

12.  $balance = BALANCE\_FACTOR(X)$

13. if  $balance > 1$  and  $key < X.left.key$

14. return ROTATE\_RIGHT(T, X)

15. if  $balance < -1$  and  $key > X.right.key$

16. return ROTATE\_LEFT(T, X)

17. if  $balance > 1$  and  $key > X.left.key$

18.  $X.left = ROTATE\_LEFT(T, X.left)$

19. return ROTATE\_RIGHT(T, X)

20. if  $balance < -1$  and  $key < X.right.key$

21.  $X.right = ROTATE\_RIGHT(T, X.right)$

22. return ROTATE\_LEFT(T, X)

23. return X

1. BALANCE\_FACTOR(X)

2. if  $X == NIL$

3. return 0

4. else

5. return  $X.left.h - X.right.h$



Na ratu je postavimo do gornje polovine  
konstantnu vrijednost  $O(1)$  jer se samo nekoliko printera  
dijeli. Najgora vrijednost je 1 jer kod prvog čita  
tako konstantnu vrijednost. Tako da vremenska složenost izlazi  
AVL stabla je jednaka kao i kod normalnog BST  
tj. je  $O(h)$  gdje je  $h$  visina stabla. Tako se radi  
o AVL stablu koje automatski balansira stablo  
vrijednost je  $O(\log N)$  VSA iznosi  $O(\log N)$ .