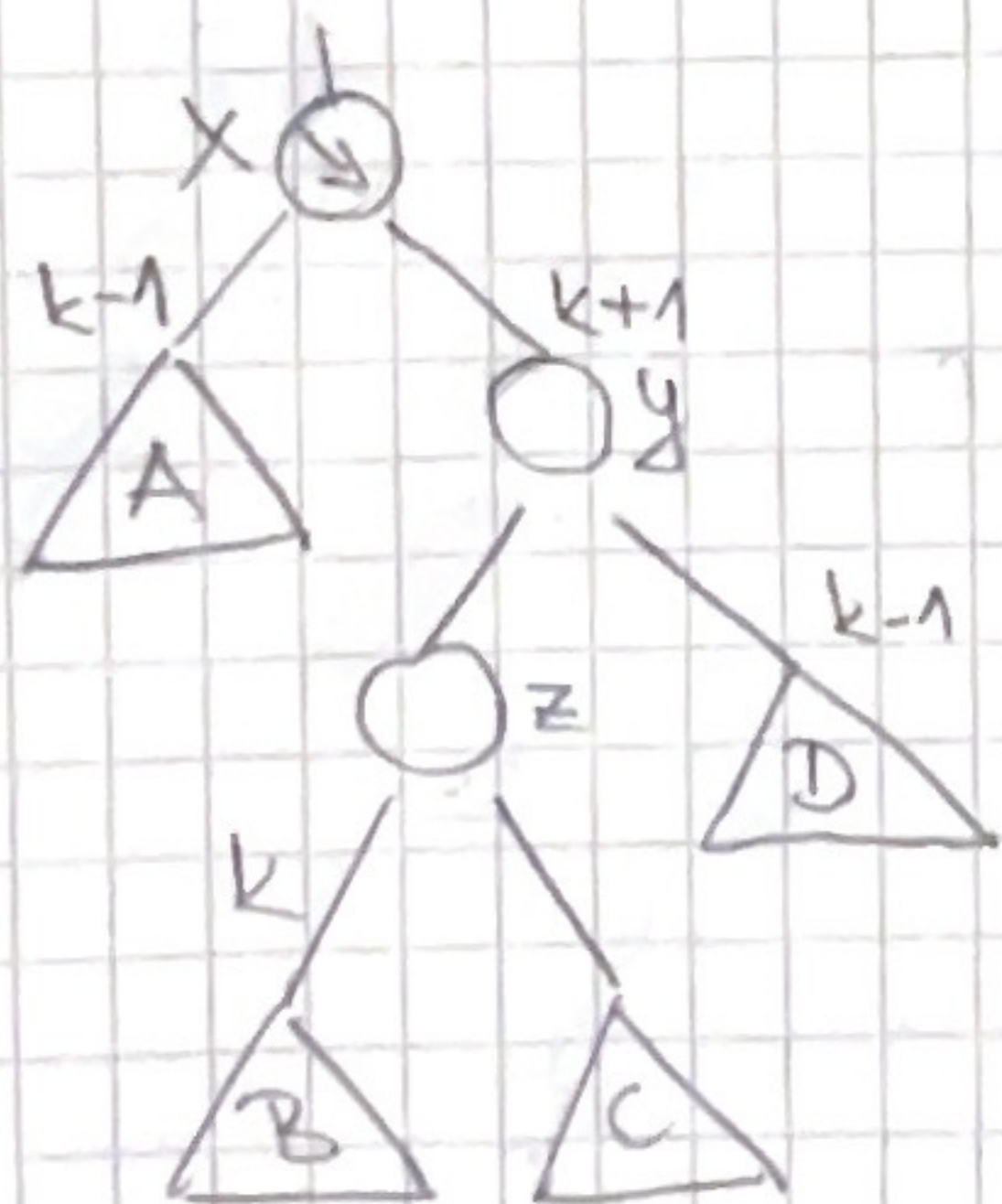


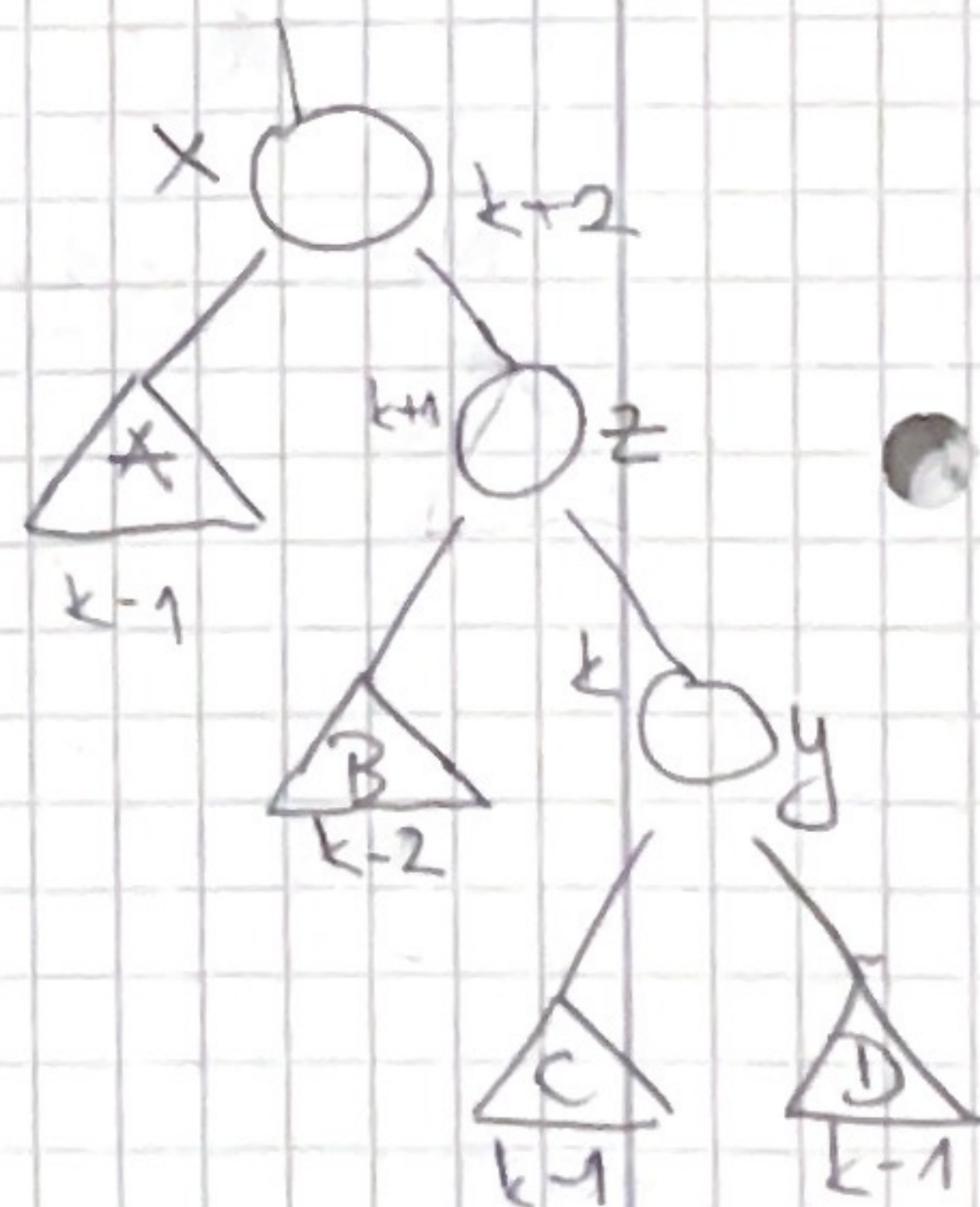
# ZADACA 5 - SPH 2

Zad 1. RIGHT-ROTATE( $T, x$ ):

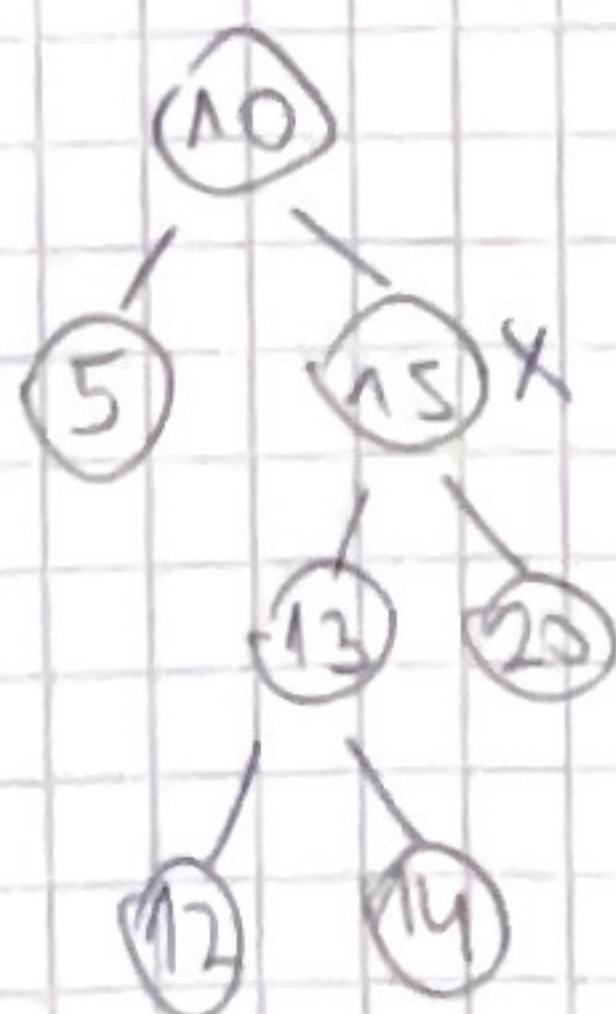
1.  $y = x.left$  //  $y$  lijevo dijete od  $x$ .
2.  $x.left = y.right$  // lijevo dijete od  $x$  postavlja se kao desno od  $y$
3.  $\text{if } y.right \neq T.nil$  // ako je desno dijete od  $y$  neprazno
4.  $y.right.p = x$  // postavljamo  $x$  kao roditelja desnog djeteta od  $y$
5.  $y.p = x.p$  // roditelj od  $x$  postaje roditelj od  $y$
6.  $\text{if } x.p == T.nil$  // ako je  $x$  root stabla
7.  $T.root = y$  // postavljamo  $y$  kao root stabla
8.  $\text{else if } x == x.p.right$  // ako je  $x$  desno dijete roditelja
9.  $x.p.right = y$  //  $y$  postaje desno dijete roditelja
10.  $\text{else } x.p.left = y$  // inače je  $y$  lijevo dijete roditelja
11.  $y.right = x$  //  $x$  postavljamo kao desno dijete od  $y$
12.  $x.p = y$  //  $y$  postaje roditelj od  $x$



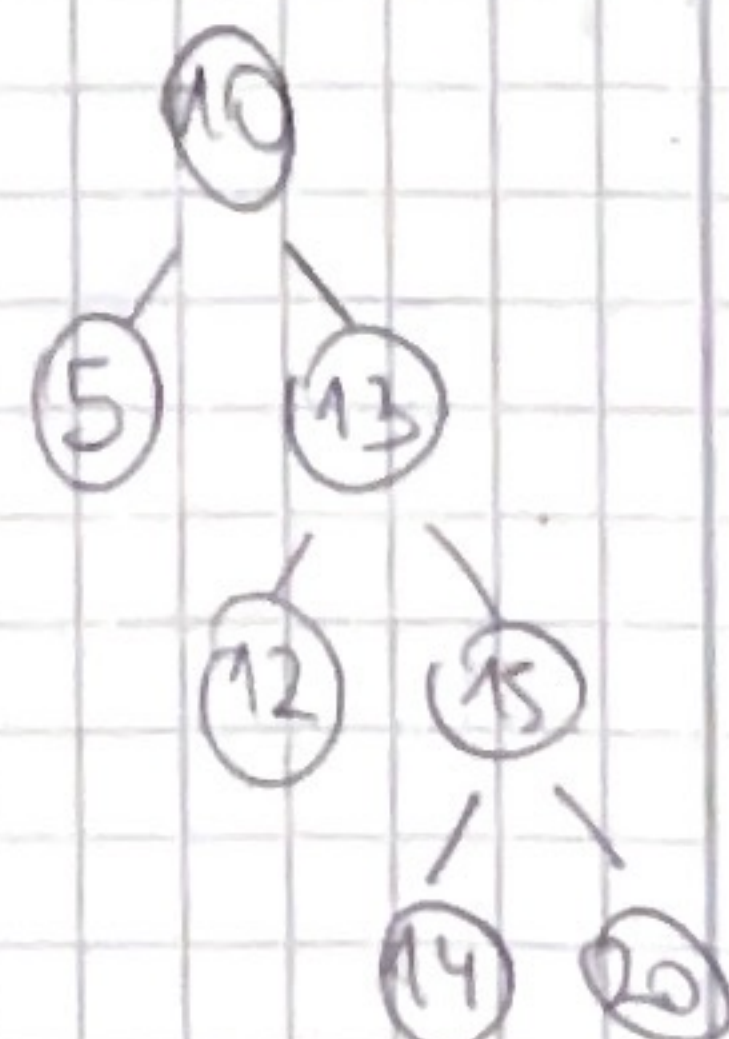
RIGHT-ROTATE( $T, x$ )



mrph.



RIGHT-ROTATE( $T, x$ )





## Zad 2. AVL-INSERT (T, z)

1. if T.root == NIL

2. T.root = z

3. else

4. INSERT (T.root, z)

5. BALANCE-FACTOR (z) = HEIGHT (z.left) - HEIGHT (z.right)

6. if BALANCE-FACTOR (z) > 1 or BALANCE-FACTOR (z) < -1

7. REBALANCE (z)

8. INSERT (x, z)

9. if z.key < x.key

10. if x.left == NIL

11. x.left = z

12. z.p = x

13. else

14. INSERT (x.left, z)

15. else

16. if x.right == NIL

17. x.right = z

18. z.p = x

19. else

20. INSERT (x.right, z)

### REBALANCE (z)

1. if BALANCE-FACTOR (z) < -1

2. if BALANCE-FACTOR (z.right) > 0

3. RIGHT-ROTATE (z.right)

4. LEFT-ROTATE (x)

5. else

6. if BALANCE-FACTOR (z.left) < 0

7. LEFT-ROTATE (z.left)

8. RIGHT-ROTATE (z)

VSA: ovi o VSA insert i rebalance alg.

U najgorem slucaju, kada se stablo mora pregrupirati nakon ubacivanja novog, insert ce proci kroz minimum stabla do mjesta ubacivanja  $O(\log n)$ . Kada se radi REBALANCE programom i BALANCE-FACTOR na svakom čvoru do unetrnutog  $O(\log n)$  VSA od AVL-INSERT je  $O(\log n)$