AVL tree

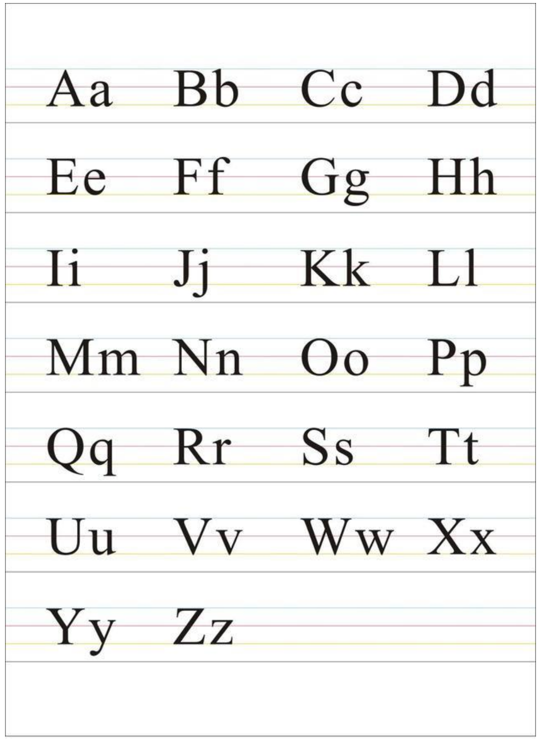
Ch. 10

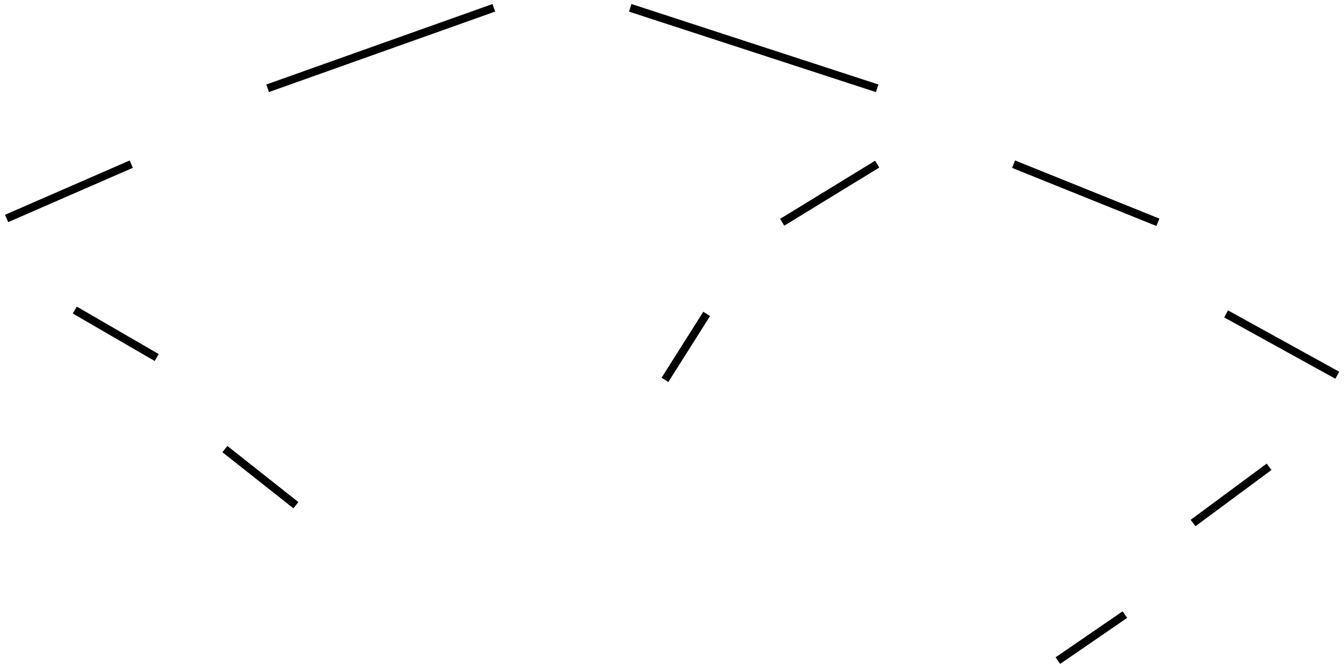
Binary search tree

- Height of generated tree depends on the input sequence.

Sequence 1:

JAN FEB MAR APR MAY JUNE JULY AUG SEP OCT NOV DEC

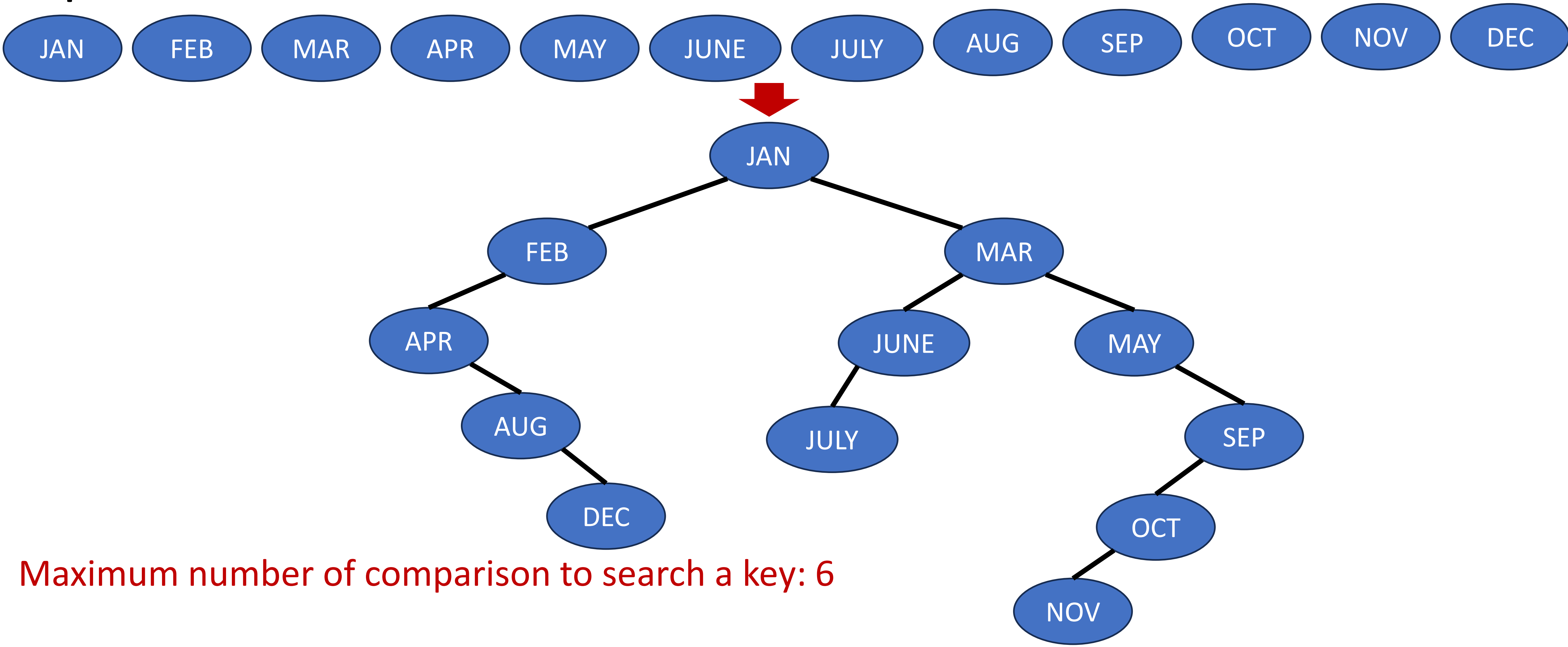




Binary search tree

- Height of generated tree depends on the input sequence.

Sequence 1:

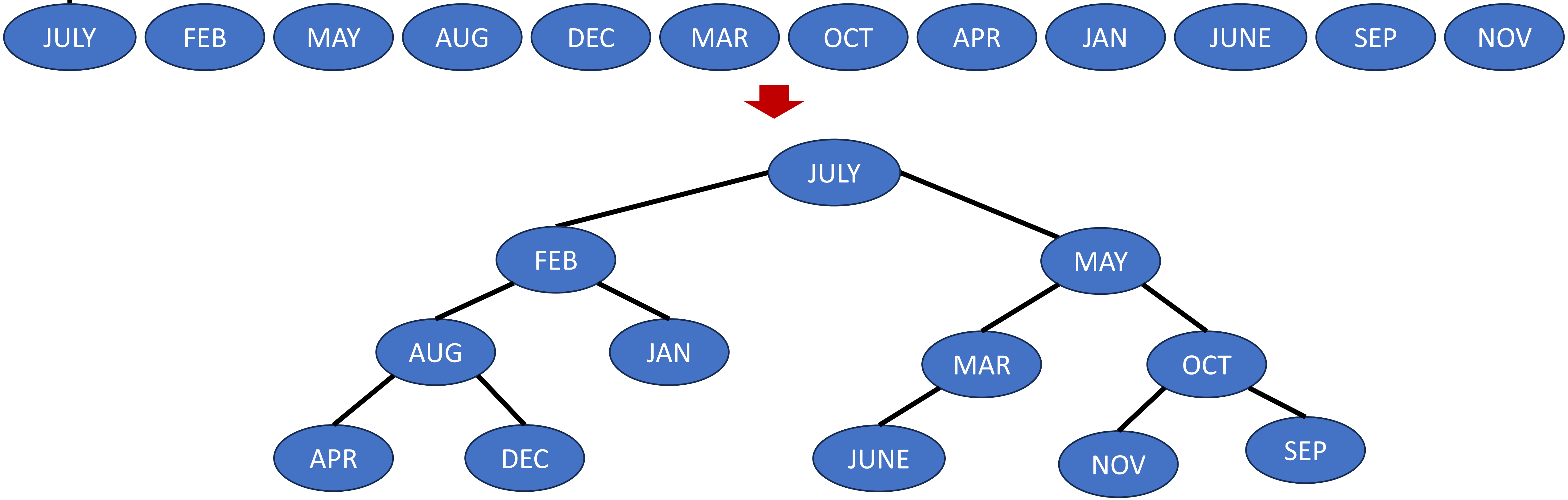


Maximum number of comparison to search a key: 6

Binary search tree

- Height of generated tree depends on the input sequence.

Sequence 2:

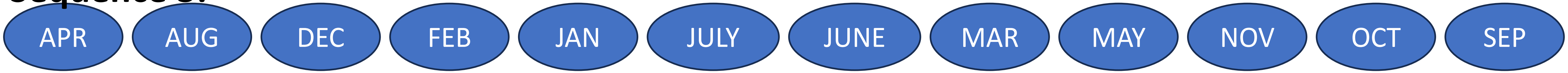


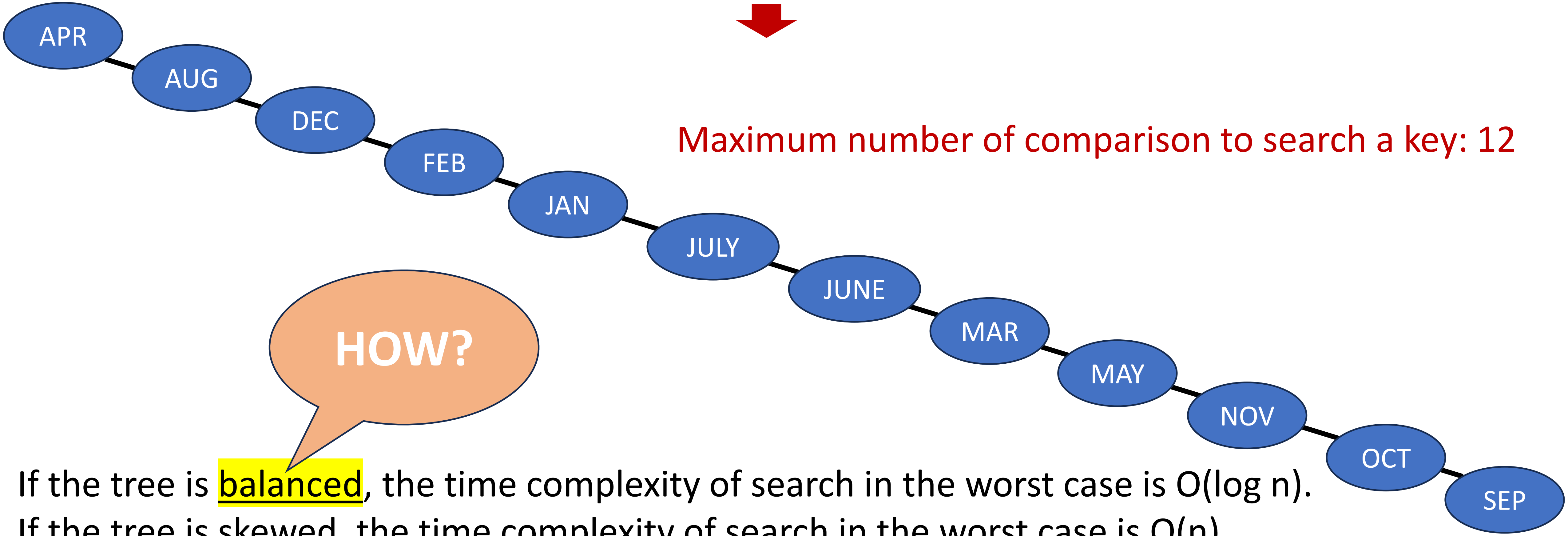
Maximum number of comparison to search a key: 4

Binary search tree

- Height of generated tree depends on the input sequence.

Sequence 3:



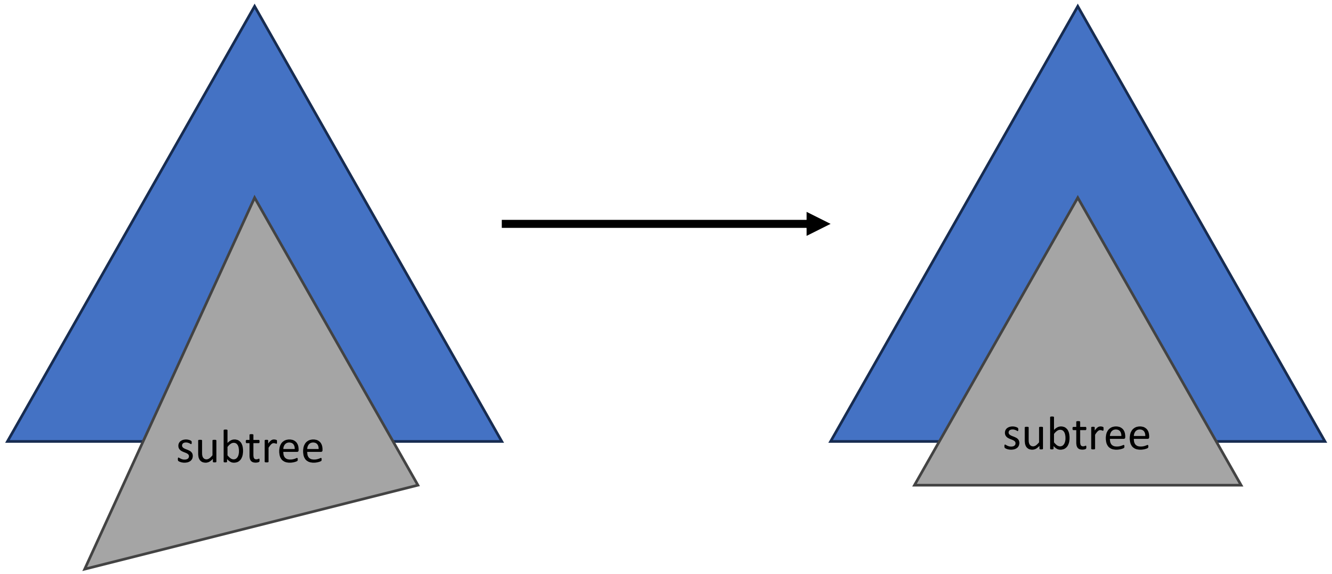


Maximum number of comparison to search a key: 12

If the tree is balanced, the time complexity of search in the worst case is O(log n).
If the tree is skewed, the time complexity of search in the worst case is O(n).

AVL-tree: an efficient binary search tree

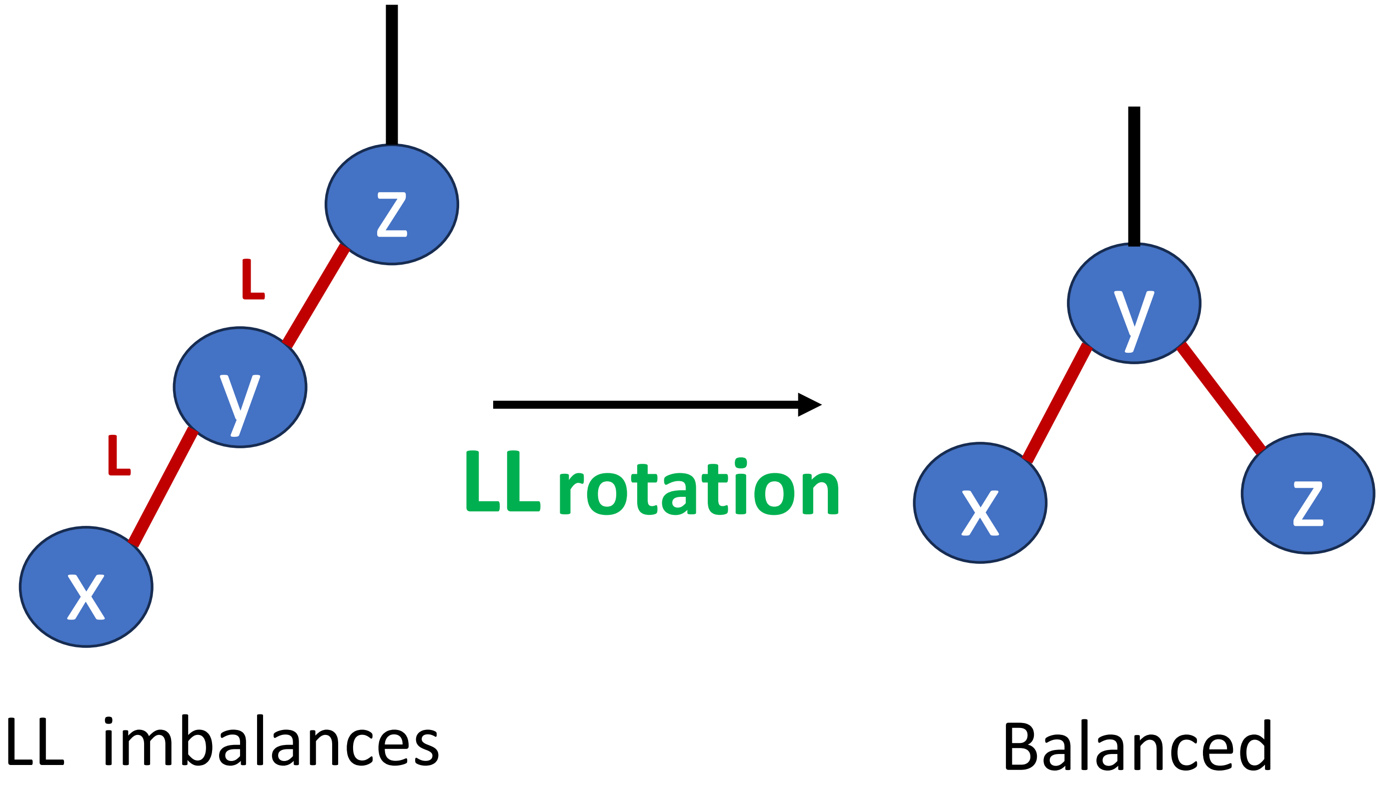
- Dynamically rebalance subtrees to maintain <u>height-balanced</u>.



Reference: Dinesh P. Mehta and Sartaj Sahni. Handbook of data structures and applications. Chapman & Hall/CRC.

Rotations

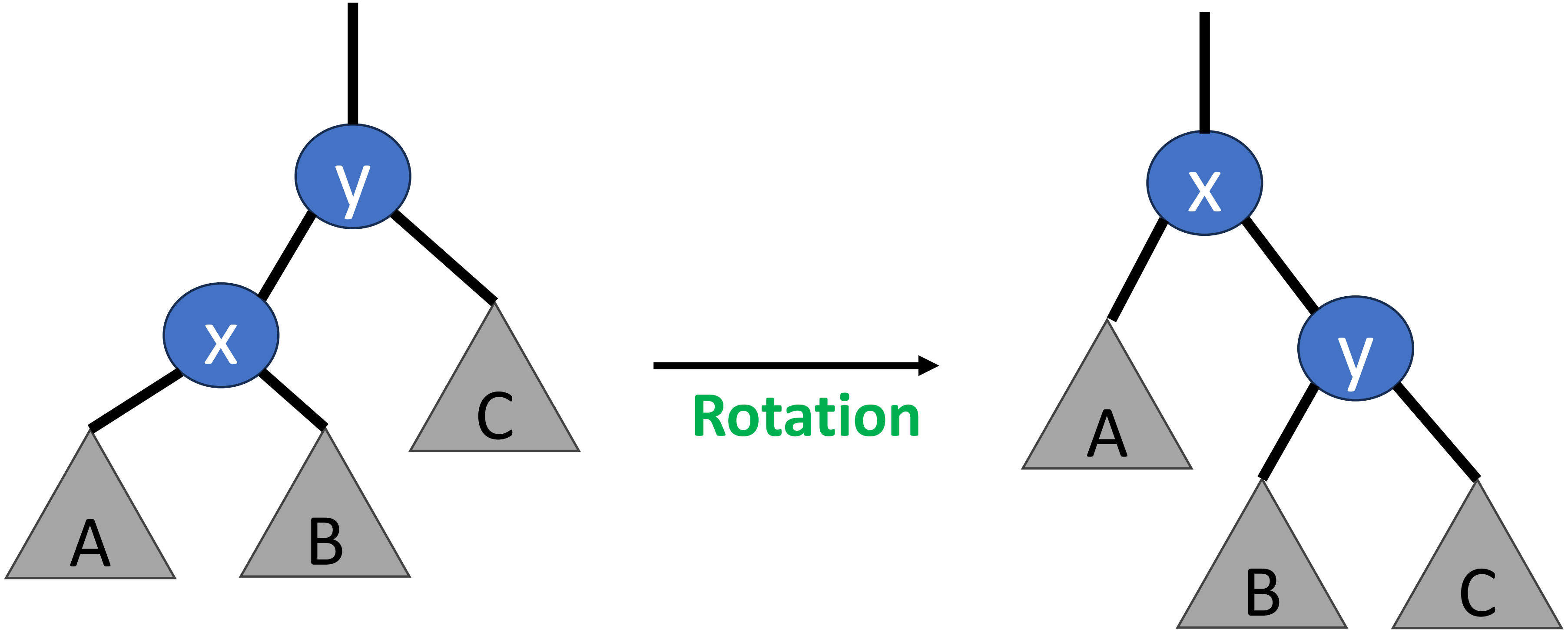
- Preserving the in-order invariant of the search tree while moving one subtree upward.



Reference: Dinesh P. Mehta and Sartaj Sahni. Handbook of data structures and applications. Chapman & Hall/CRC.

Rotations (with subtrees)

- Preserving the <u>in-order</u> invariant of the search tree while moving one subtree upward.



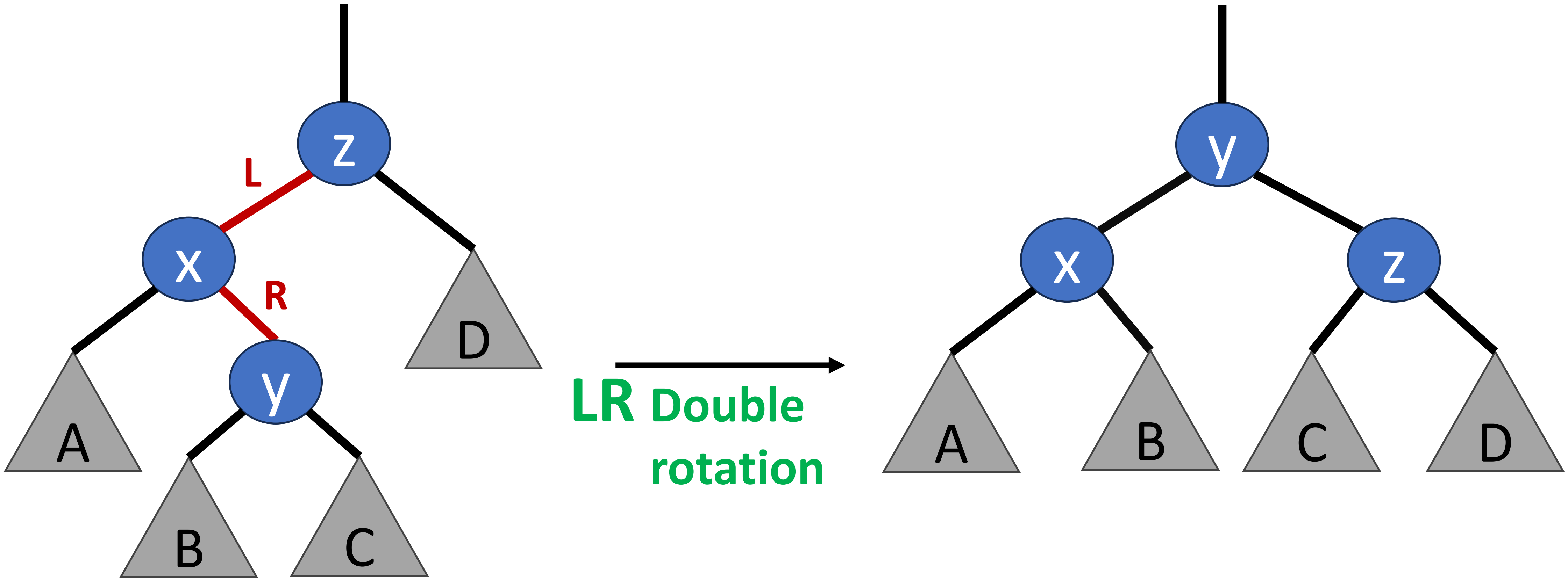
Inorder: AXBYC

Inorder: AXBYC

Reference: Dinesh P. Mehta and Sartaj Sahni. Handbook of data structures and applications. Chapman & Hall/CRC.

Double rotations

- Equivalent to two consecutive rotations

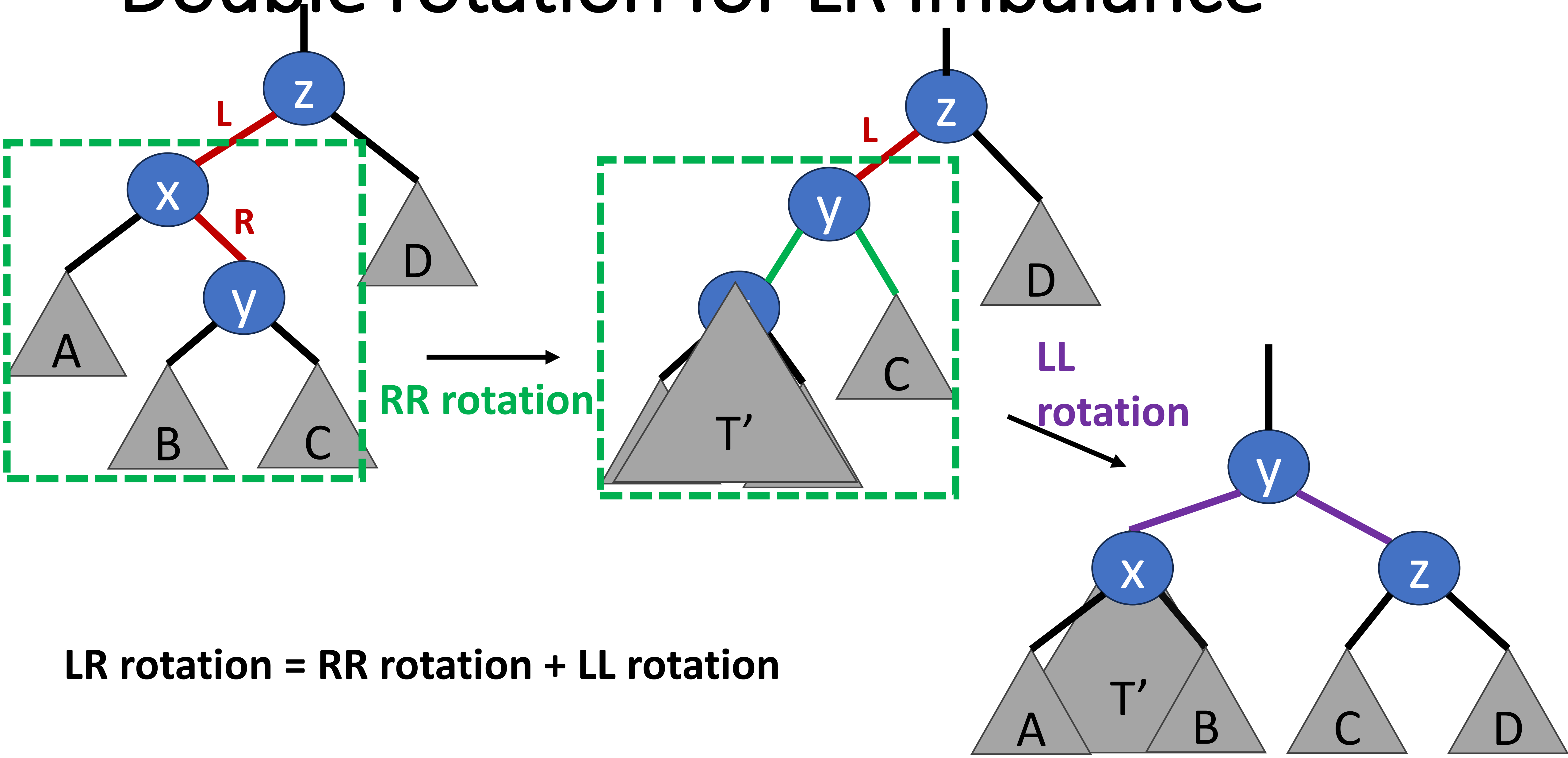


Inorder: AXBYCZD

Inorder: AXBYCZD

Reference: Dinesh P. Mehta and Sartaj Sahni. Handbook of data structures and applications. Chapman & Hall/CRC.

Double rotation for LR imbalance



LR rotation = RR rotation + LL rotation

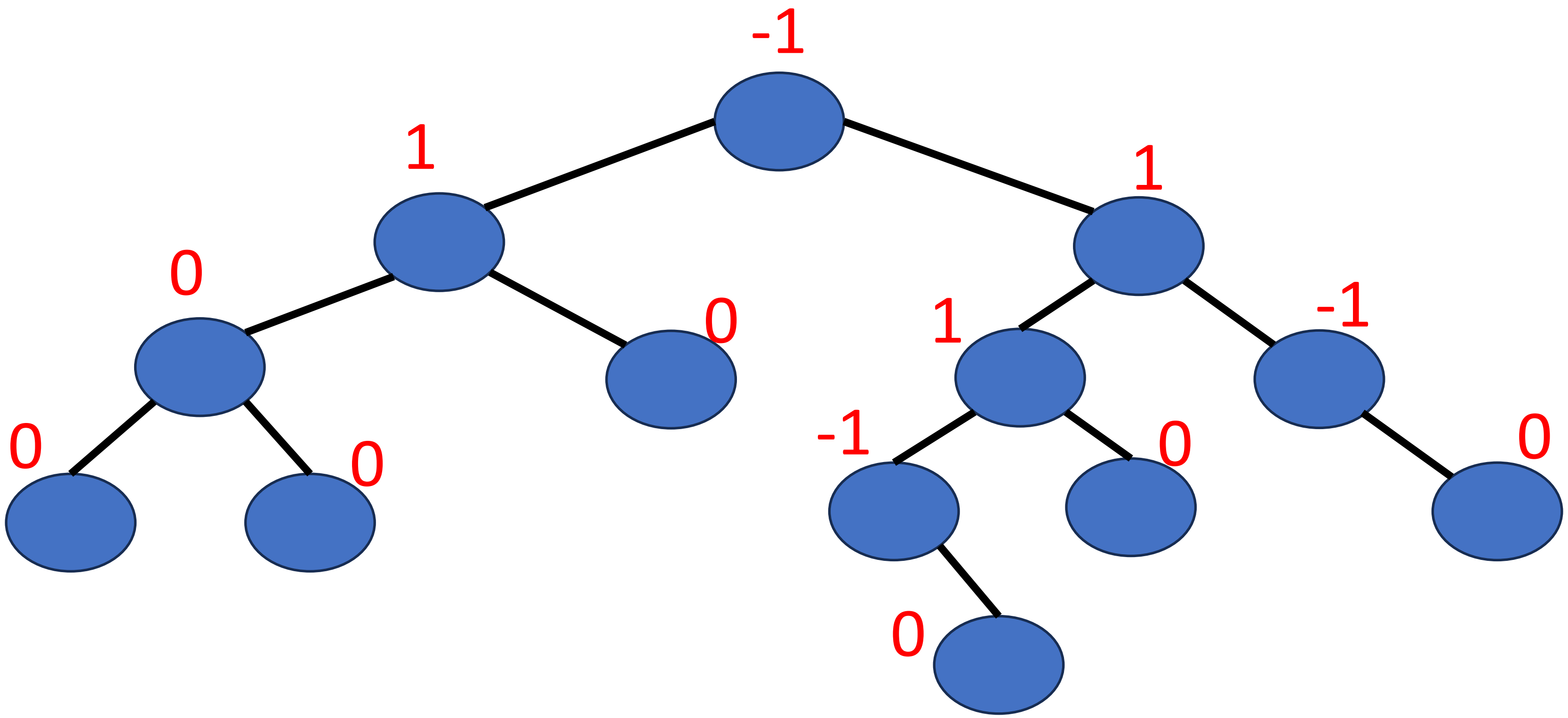
Reference: Dinesh P. Mehta and Sartaj Sahni. Handbook of data structures and applications. Chapman & Hall/CRC.

Balance factor in AVL tree

- For **every node** *x*, define its balance factor *BF*:

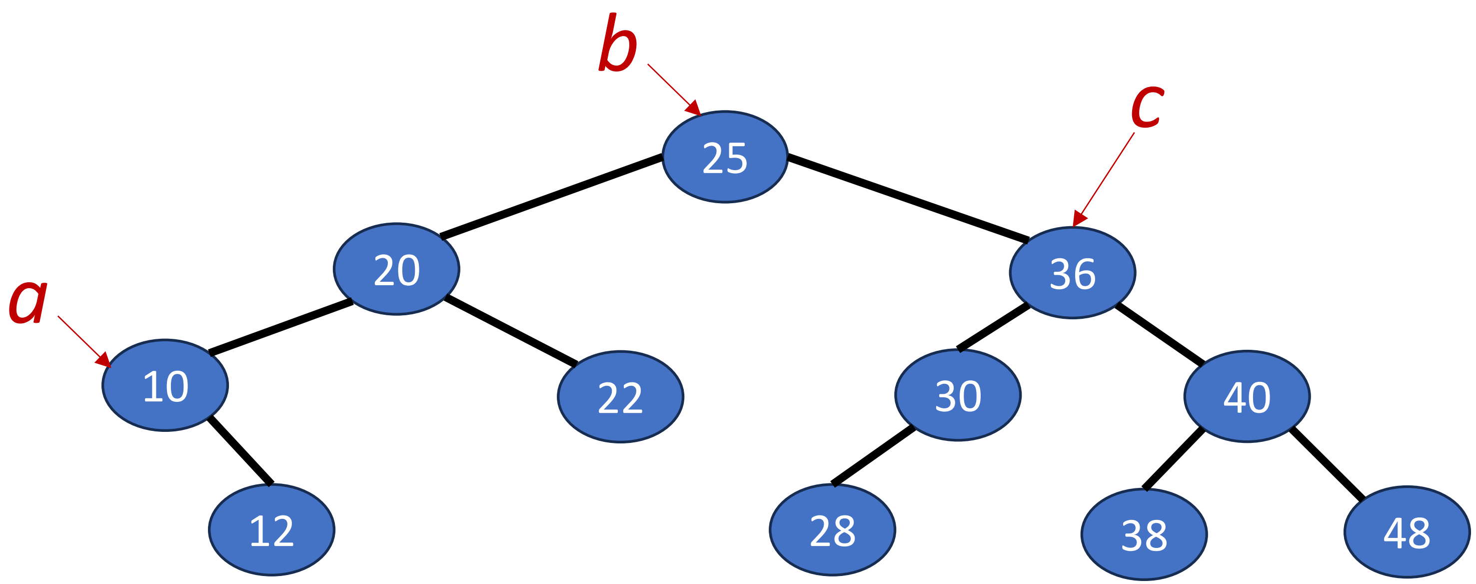
  $$BF(x) = \text{height}(x\text{'s left subtree}) - \text{height}(x\text{'s right subtree})$$

- Balance factor of every node x is -1, 0, or 1



Exercise

- Given the following AVL tree.
  Q9: What is the balance factor of Node *a*?
  Q10: What is the balance factor of Node *b*?
  Q11: What is the balance factor of Node *c*?



Please reply your answers of Q9-11 via the following link:

https://forms.gle/aWUwuR7JjTtyMCMv7

Group members: 2~4 people

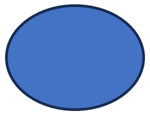
Height of AVL tree is O(log *n*)

n: total number of nodes in the AVL tree

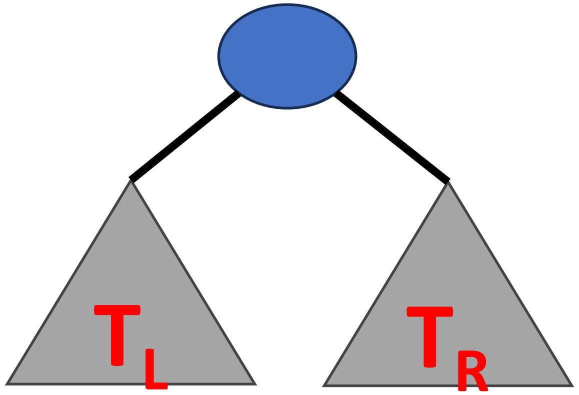
Proof:

Let N_h = number of nodes in an AVL tree whose height is h.

- $N_0 = 0$.
- $N_1 = 1$.



- N_h for $h > 1$.

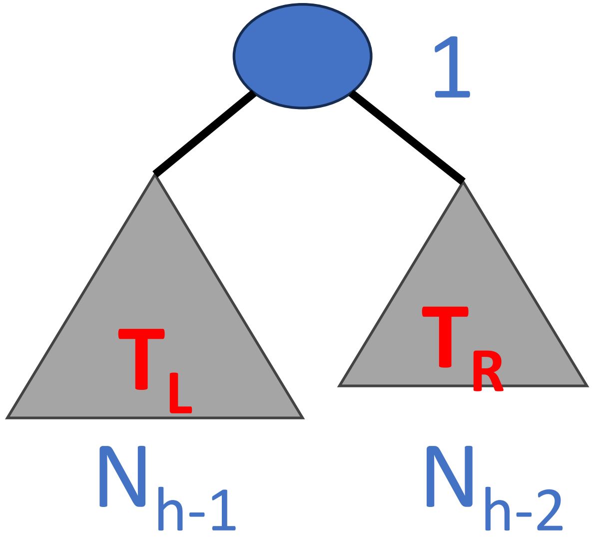


- Both T_L and T_R are AVL trees.
- Assume that height of T_L is h-1.
  Then height of T_R can be h-2 in the worst case.
- T_L has N_{h-1} nodes.
  T_R has N_{h-2} nodes.
- So, $N_h = N_{h-1} + N_{h-2} + 1$.

N_h is similar with Fibonacci Number

Let N_h = number of nodes in an AVL tree whose height is h.

- $F_0 = 0, F_1 = 1.$
- $F_i = F_{i-1} + F_{i-2}, i > 1.$
- $N_0 = 0, N_1 = 1.$
- $N_h = N_{h-1} + N_{h-2} \underline{+ 1}, i > 1.$

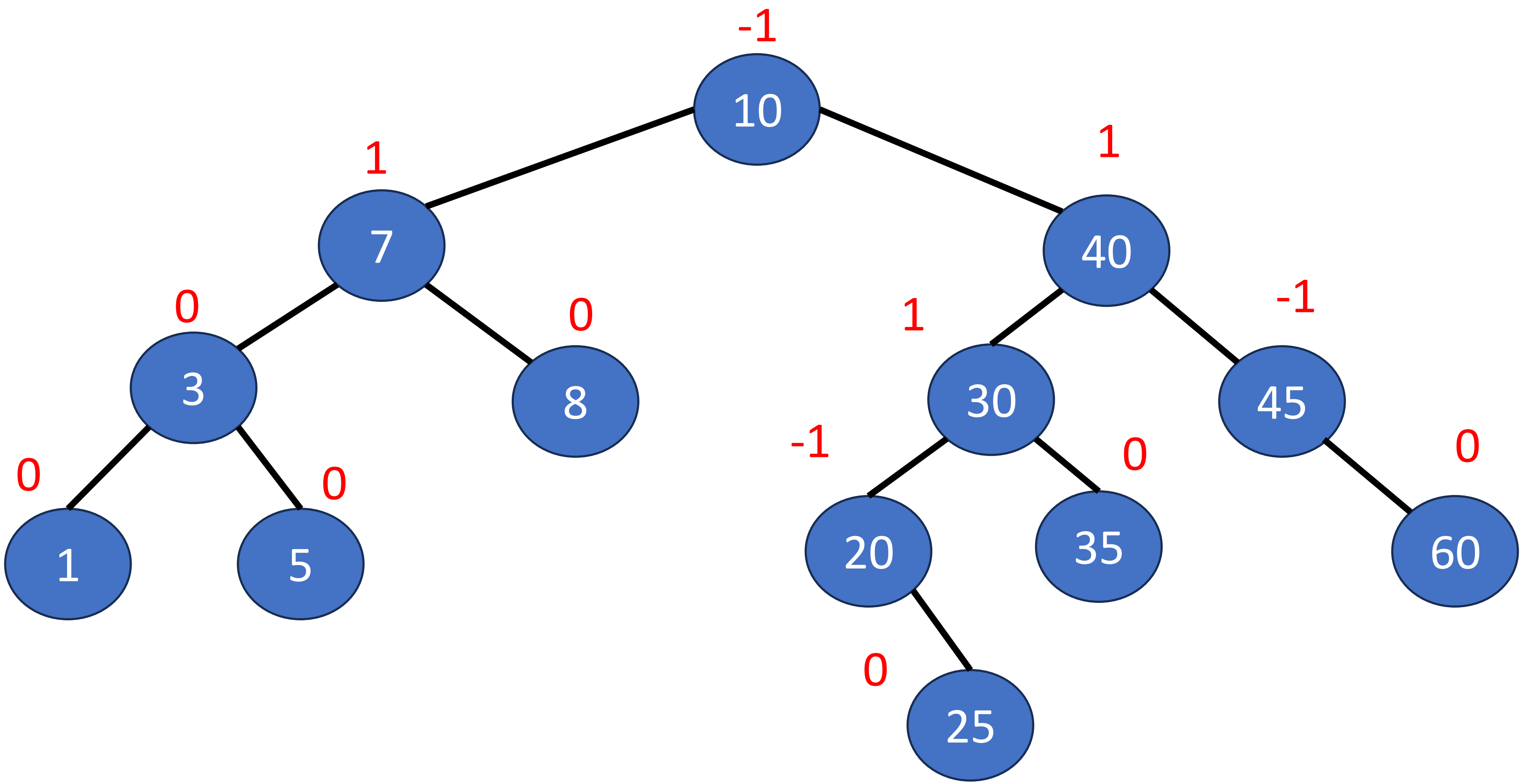


If ignore, N_h becomes Fibonacci number

- $F_i \sim \phi^i/\mathrm{sqrt}(5).$
- $\phi = (1 + \mathrm{sqrt}(5))/2 \approx 1.618$

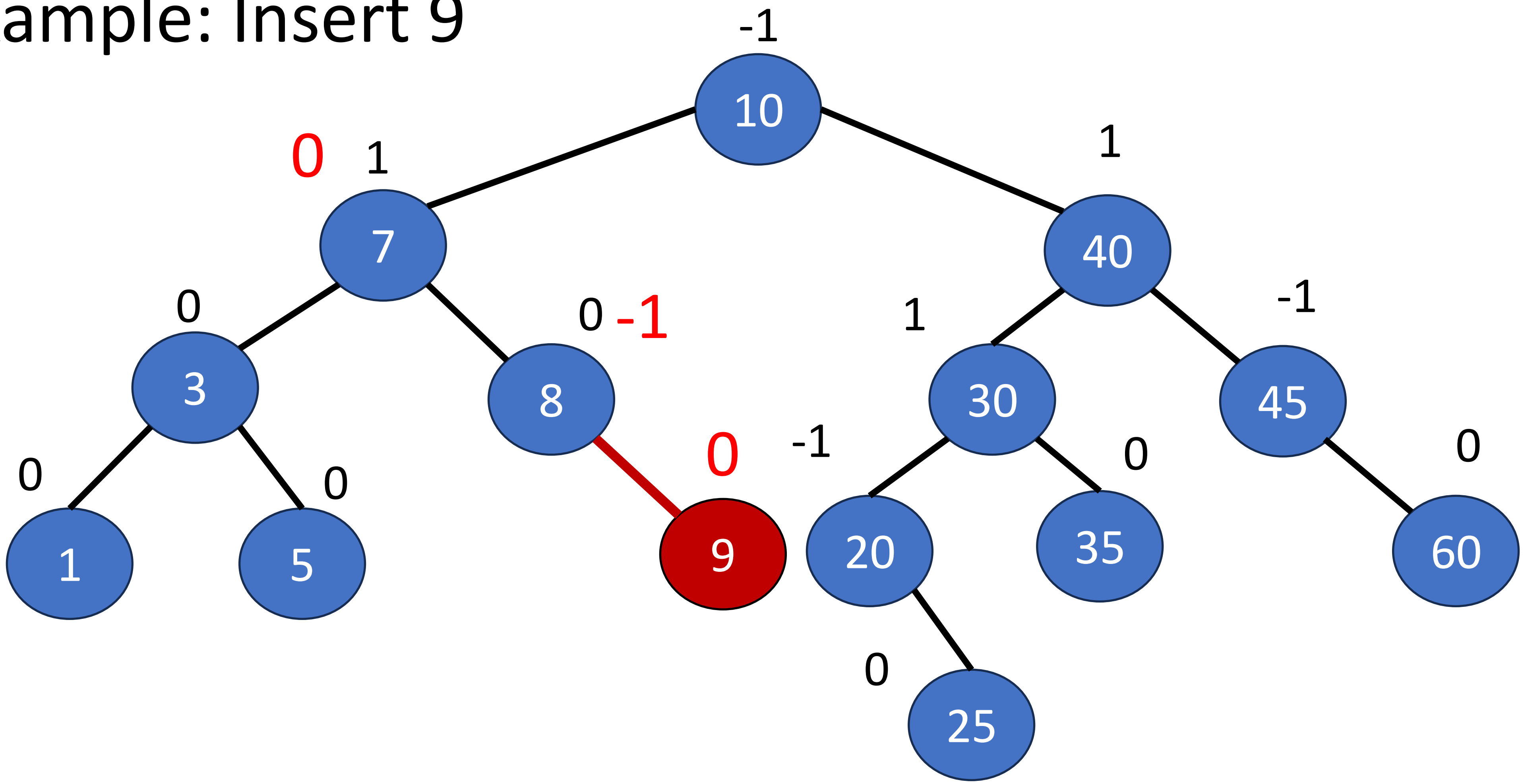
Height of AVL tree is $O(\log_\phi(n))$

AVL tree example



Operation: Insertion

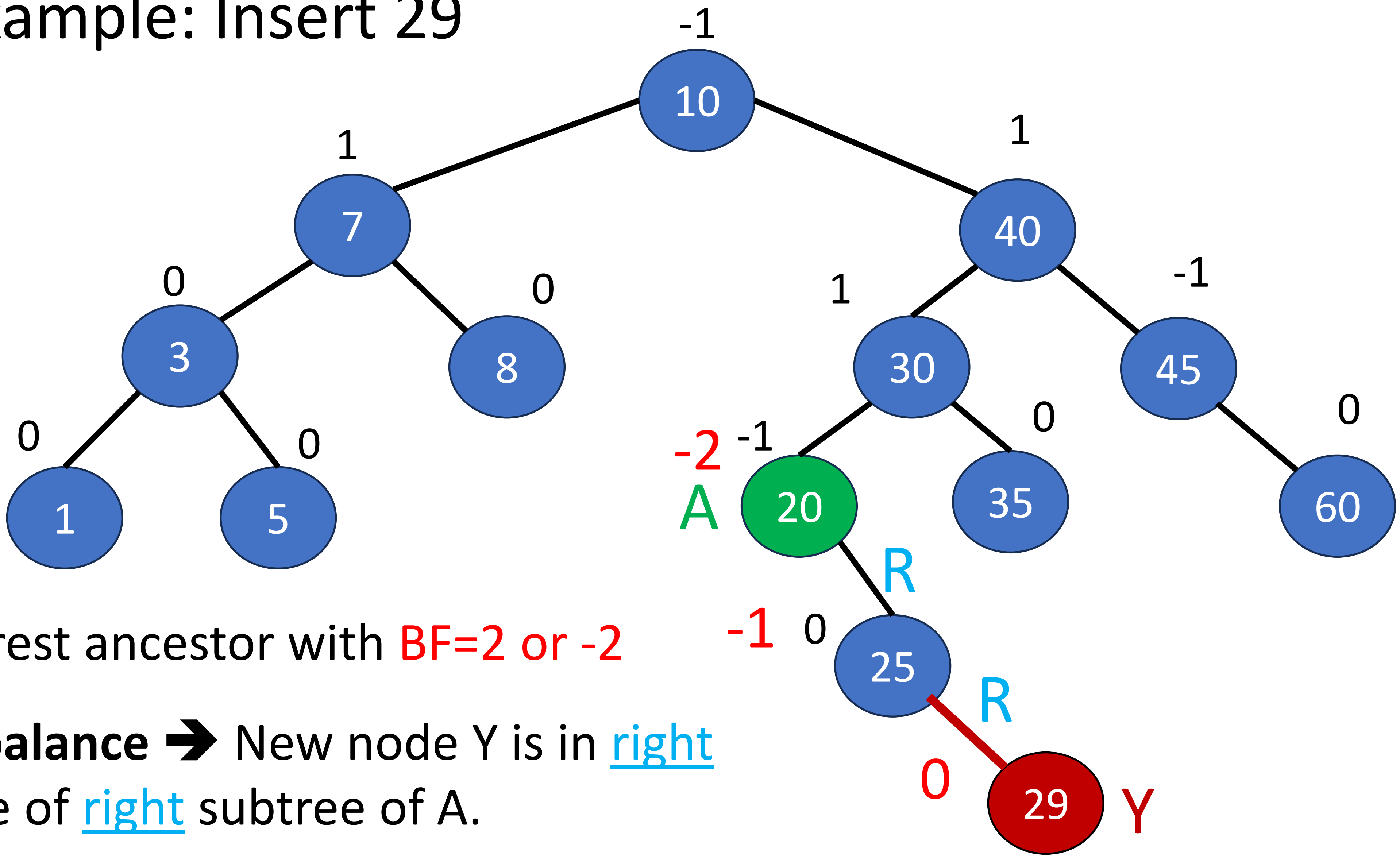
- Example: Insert 9



- Insert the node using the insertion algorithm of binary search tree.
- If BF=0 becomes -1 or 1, subtree height changes and rebalancing may be needed.

Operation: Insertion

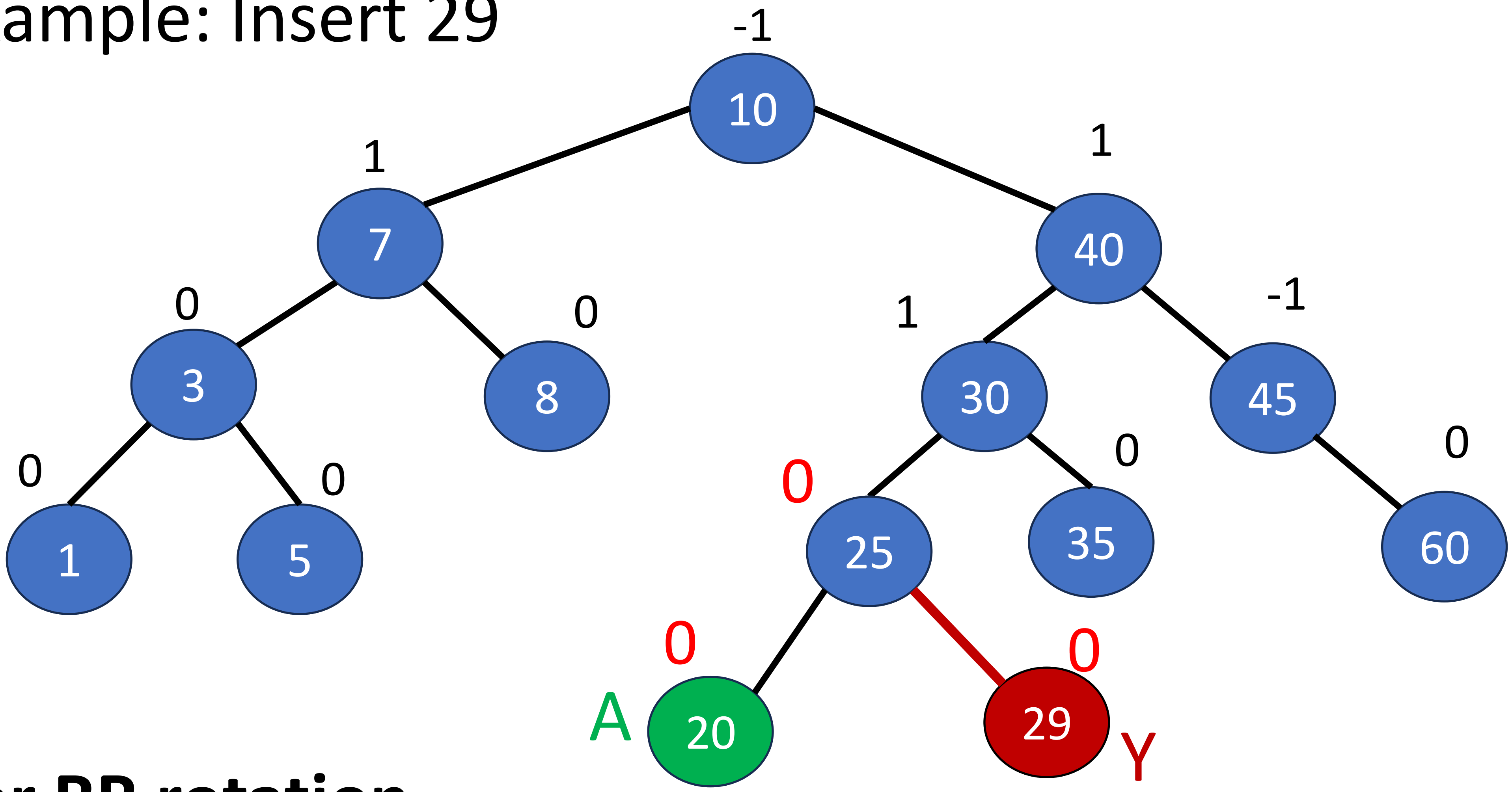
- Example: Insert 29



- A: nearest ancestor with BF=2 or -2
- **RR imbalance** ➔ New node Y is in right subtree of right subtree of A.

Operation: Insertion

- Example: Insert 29



- **After RR rotation**

Insertion may cause imbalance

- Following insert, retrace path towards root and adjust balance factors as needed.
- Stop when you reach a node whose balance factor becomes 0, 2, or –2, or when you reach the root.
- The new tree is **not** an AVL tree only if you reach a node whose balance factor is either 2 or –2.
- In this case, we say the tree has become unbalanced.

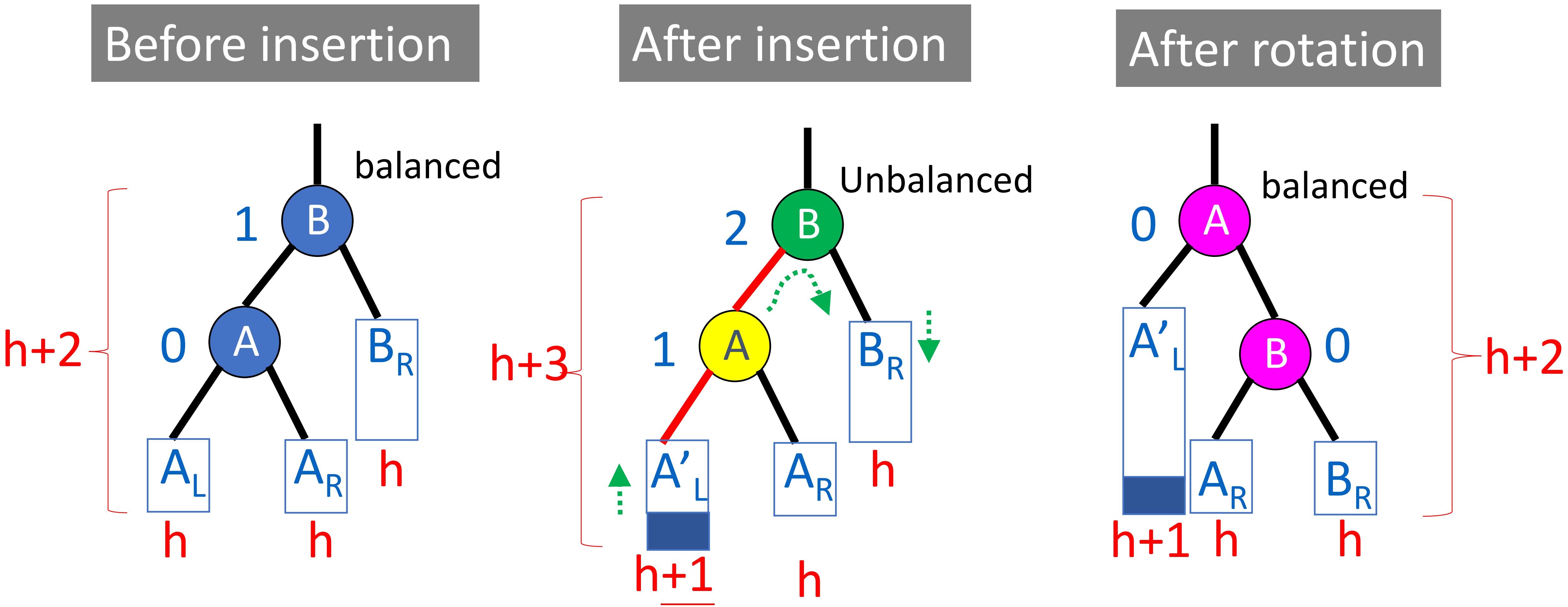
Imbalance type

A: The nearest ancestor of the newly inserted node whose balance factor becomes +2 or –2 following the insert.

Four types of imbalance:

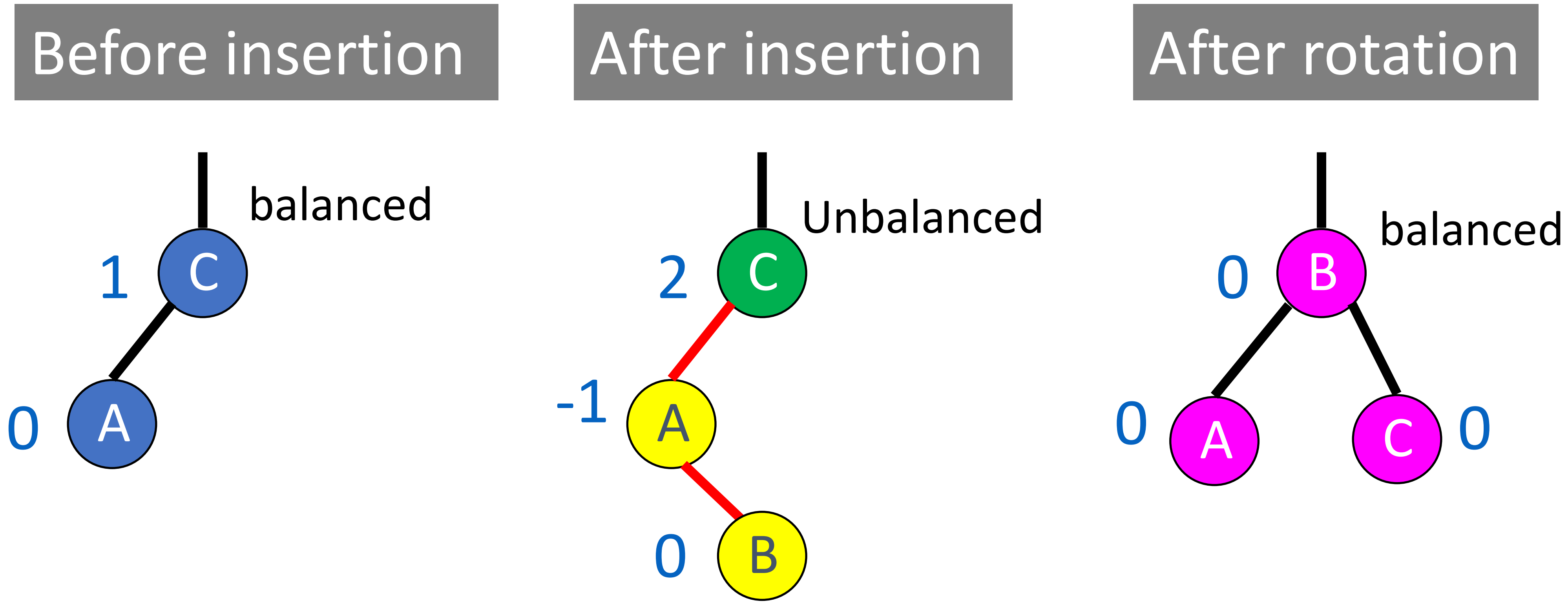
- **RR** … newly inserted node is in the **right** subtree of the **right** subtree of A.
- **LL** … **left** subtree of **left** subtree of A.
- **RL** … **left** subtree of **right** subtree of A.
- **LR** … **right** subtree of **left** subtree of A.

LL rotation



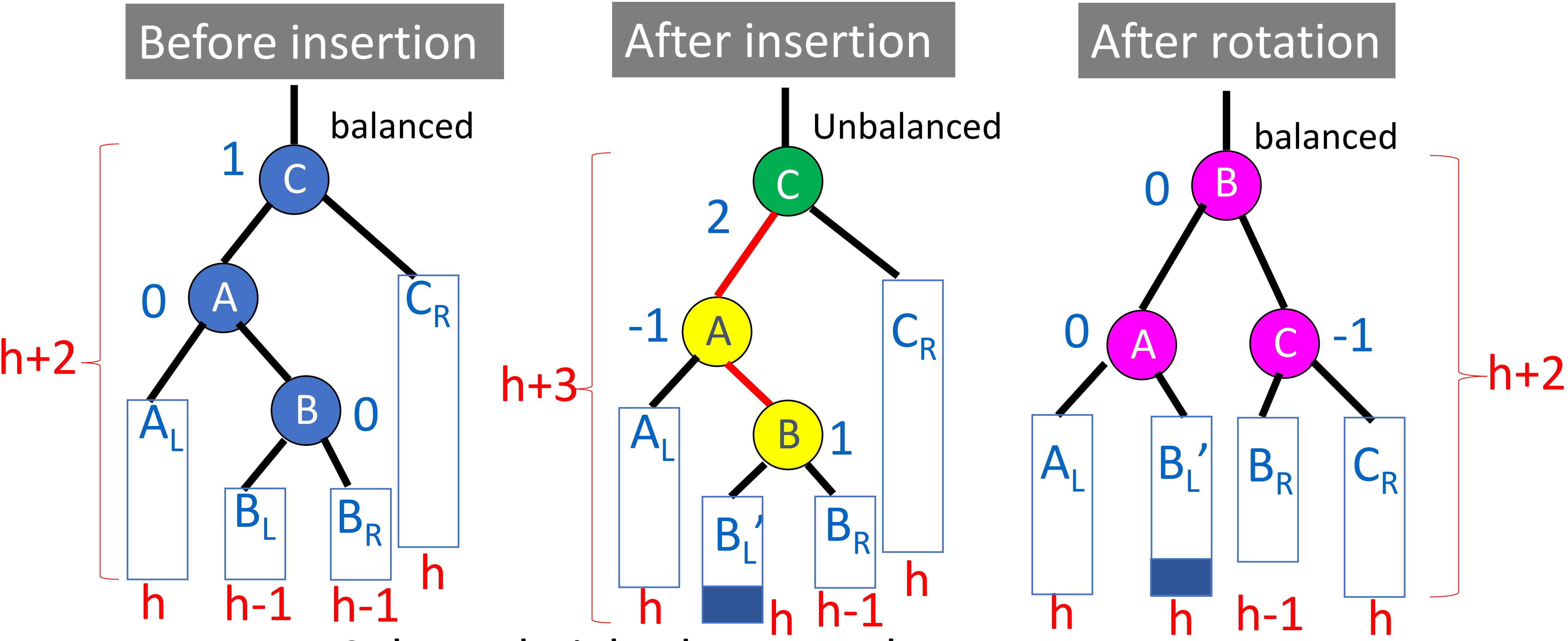
- Subtree height does not change. (h+2 → h+2)
- No adjustment to be done for its ancestors.

LR rotation (Case 1)



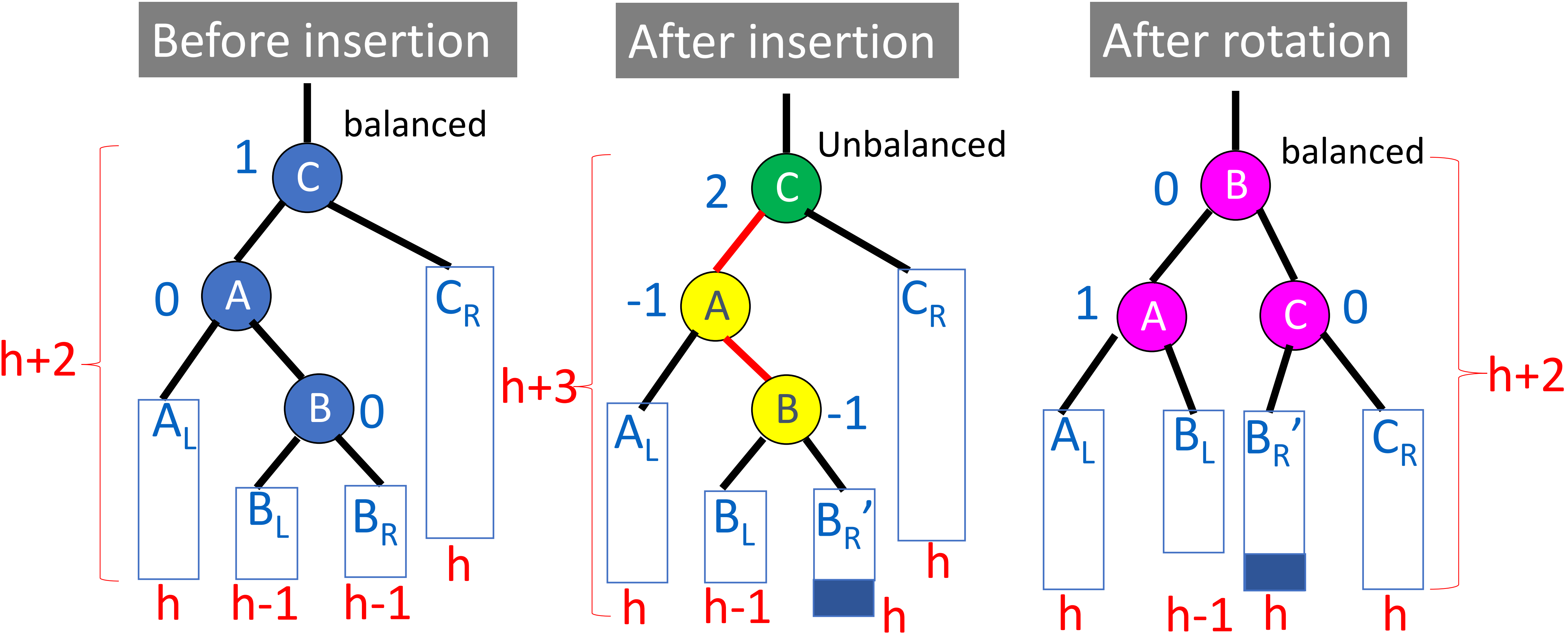
- Subtree height does not change.
- No adjustment to be done for its ancestors.

LR rotation (Case 2)



- Subtree height does not change.
- No adjustment to be done for its ancestors.

LR rotation (Case 3)

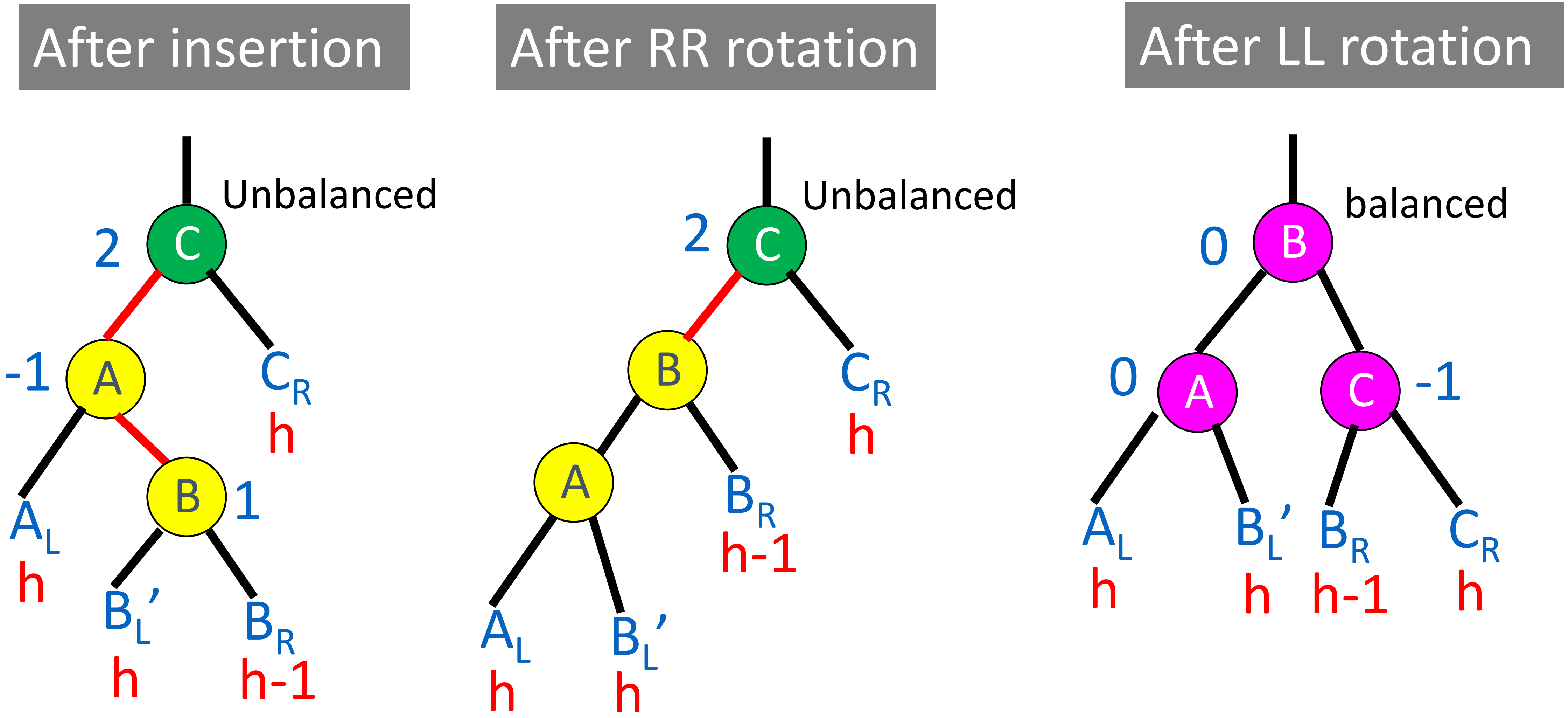


- Subtree height does not change.
- No adjustment to be done for its ancestors.

Single and double rotations

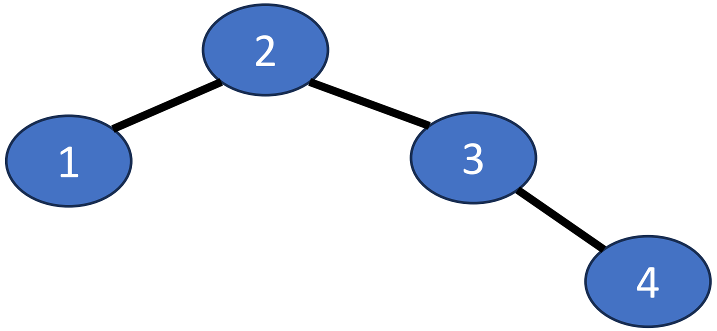
- Single
  - LL and RR
- Double
  - LR and RL
  - LR is RR (first) followed by LL (second)
  - RL is LL (first) followed by RR (second)

LR rotation is RR + LL rotations

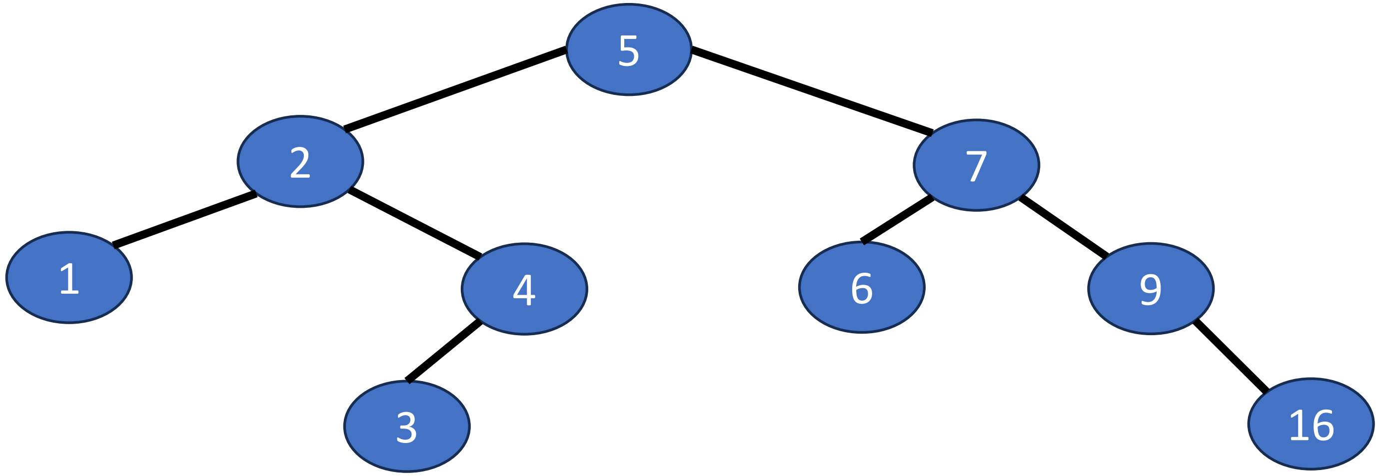


Exercise

- Q12: Please write out the result after inserting 5 into to the following AVL tree.



- Q13: Please write out the result after inserting 15 into to the following AVL tree.



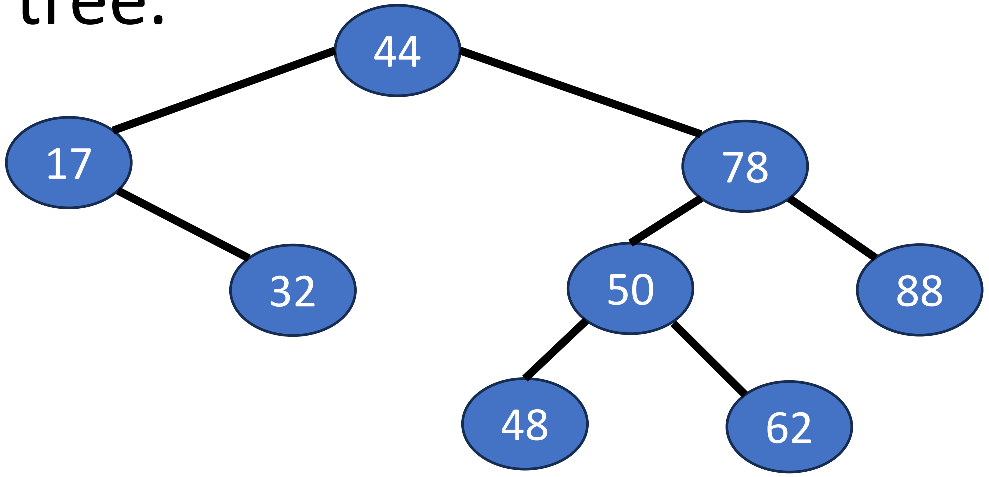
Please reply your answers of Q12-13 via the following link:

https://forms.gle/aWUwuR7JjTtyMCMv7

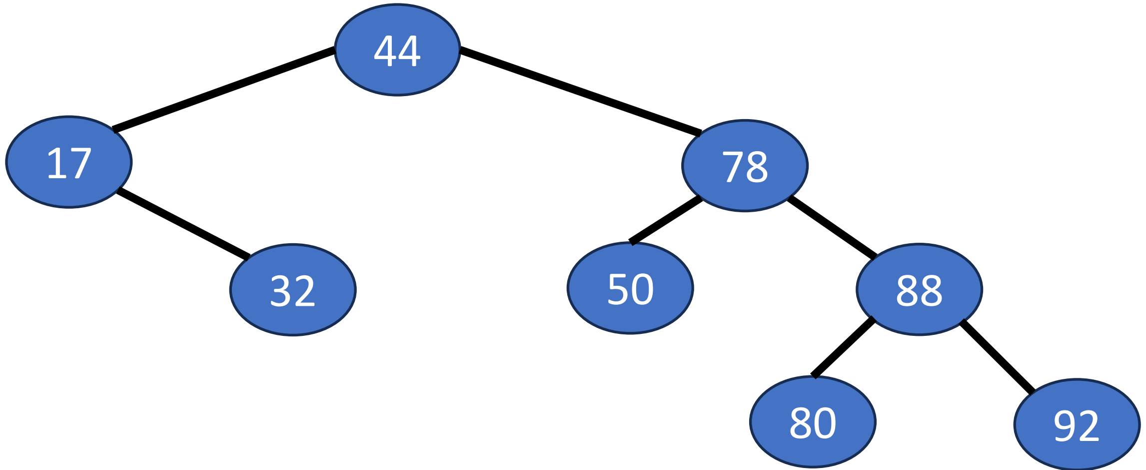
Group members: 2~4 people

Exercise

- Q14: Please write out the result after inserting 58 into to the following AVL tree.



- Q15: Please write out the result after inserting 79 into to the following AVL tree.



Please reply your answers of Q14-15 via the following link:

https://forms.gle/aWUwuR7JjTtyMCMv7

Group members: 2~4 people