ساختمان داده ها

درخت AVL
(AVL Trees)

مدرس: غیاثی‌شیرازی
دانشگاه فردوسی مشهد

AVL Trees

AVL اولاً یک درخت جستجوی دودویی است

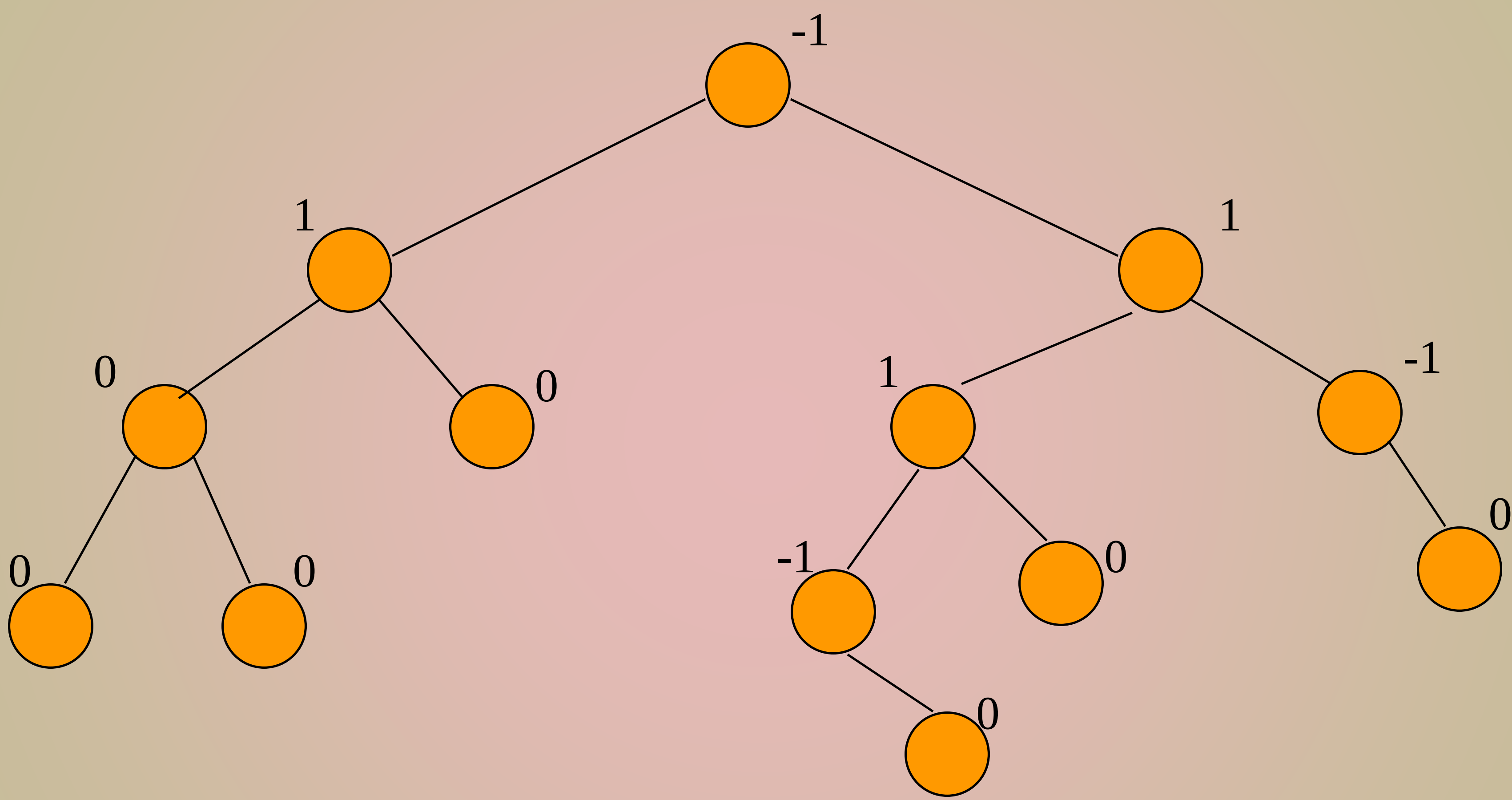
- named after **A**delson-**V**elsky & **L**andis
- binary tree   Search
- for every node x, define its balance factor

  balance factor of x = height of left subtree of x

  – height of right subtree of x

- balance factor of every node x is – 1, 0, or 1
- $\log_2 (n+1) <= \text{height} <= 1.44 \log_2 (n+2)$

Balance Factors



This is an AVL tree.

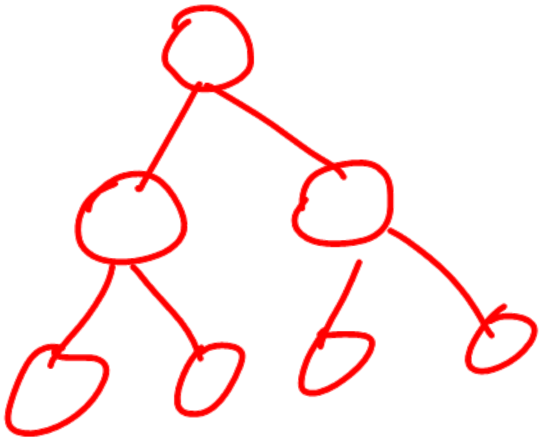
Height Of An AVL Tree

The height of an AVL tree that has n nodes is at most $1.44 \log_2 (n+2)$.

The height of every n node binary tree is at least $\log_2 (n+1)$.

$\log_2 (n+1) <= \text{height} <= 1.44 \log_2 (n+2)$

ارتفاع h



h	n
1	1
2	3
3	7

$$n \leqslant 2^h - 1 < 2^h$$

$$\log_2 n < h$$

$$n = 2^h - 1$$

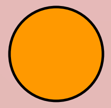
حداکثر تعداد گره‌ها

$$n+1 \leqslant 2^h$$

$$\log_2(n+1) \leqslant h \checkmark$$

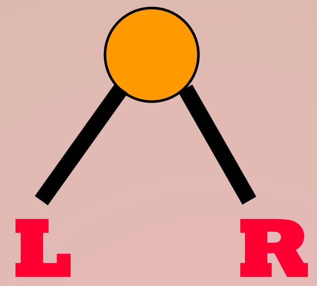
Proof Of Upper Bound On Height

- Let N_h = min # of nodes in an AVL tree whose height is h.
- $N_0 = 0$.
- $N_1 = 1$.



N_h, $h > 1$

N_{h-2} $h-2$



$h-1$ N_{h-1}

- Both **L** and **R** are AVL trees.
- The height of one is $h-1$.
- The height of the other is $h-2$.
- The subtree whose height is $h-1$ has N_{h-1} nodes.
- The subtree whose height is $h-2$ has N_{h-2} nodes.
- So, $N_h = N_{h-1} + N_{h-2} + 1$. $h \geqslant 2$

Relation to Fibonacci Numbers

0, 1, $\boxed{1, 2}$, 3

- $F_0 = 0$, $F_1 = 1$. $F_i = F_{i-1} + F_{i-2}$, $i > 1$.
- $N_0 = 0$, $N_1 = 1$ $\underbrace{N_h^{+1}}_{M_h} = \underbrace{N_{h-1}^{+1}}_{M_{h-1}} + \underbrace{N_{h-2} + 1}_{M_{h-2}}$, $i > 1$.
- $M_h := N_h + 1$
- $\boxed{M_0 = 1, M_1 = 2}$ $M_h = M_{h-1} + M_{h-2}$, $i > 1$.
- $M_h = F_{h+2}$. $N_h = F_{h+2} - 1$.

حل روابط
بازگشتی از
ریاضیات گسسته

$$F_i = \frac{\phi^i - (-\phi)^{-i}}{\sqrt{5}} \geq \frac{\phi^i - \phi^{-i}}{\sqrt{5}}$$

$$\phi = \frac{1 + \sqrt{5}}{2}$$

حل کامل

$$\phi = \frac{1+\sqrt{5}}{2} \approx 1.618$$

$$F_i = \frac{\phi^i - (-\phi)^{-i}}{\sqrt{5}} \geq \frac{\phi^i - \phi^{-i}}{\sqrt{5}}$$

$$\overbrace{N_h + 1}^{M_h} = F_{h+2} \geq \frac{\phi^{h+2} - (\phi^{-1})^{h+2}}{\sqrt{5}}$$

$$\geq \frac{\phi^{h+2} - (\phi^{-1})^2}{\sqrt{5}} \geq \frac{\phi^{h+2}}{\sqrt{5}} - 0.18$$

حل کامل

$\log_b a = \log_c a \, \log_b c$

$$N_h + 2 \geq \frac{\phi^{h+2}}{\sqrt{5}}$$

$$\log_{\phi}(\overbrace{N_h+2}^{a}) = \left(\log_{\phi} 2\right) \log_2(N_h+2)$$

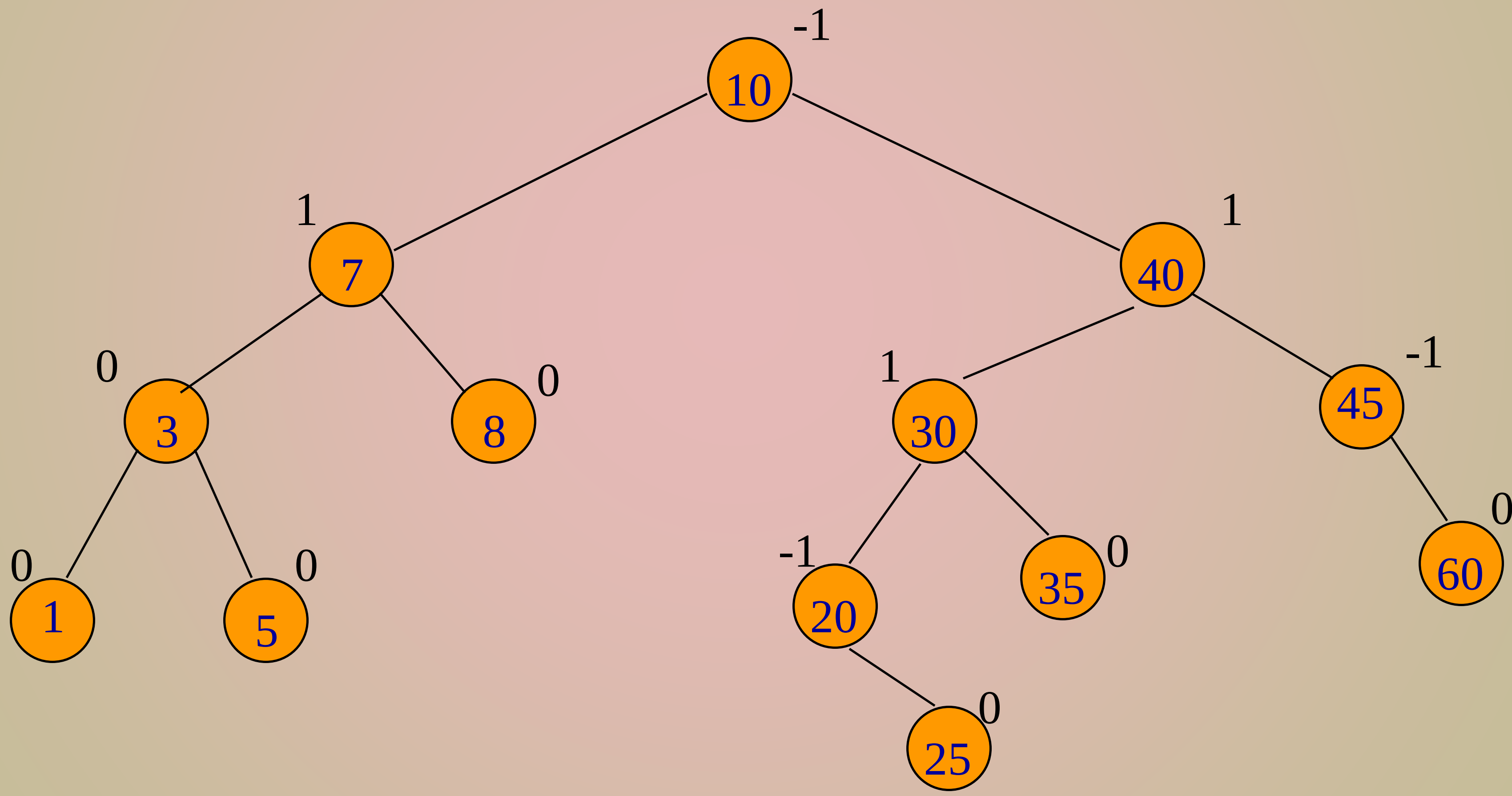
$$\geq h + 2 - \log_{\phi}\sqrt{5} \geq h + 2 - 1.7 \geq h$$

$$\log_{\phi} 2 = 1.44042009 \ldots$$

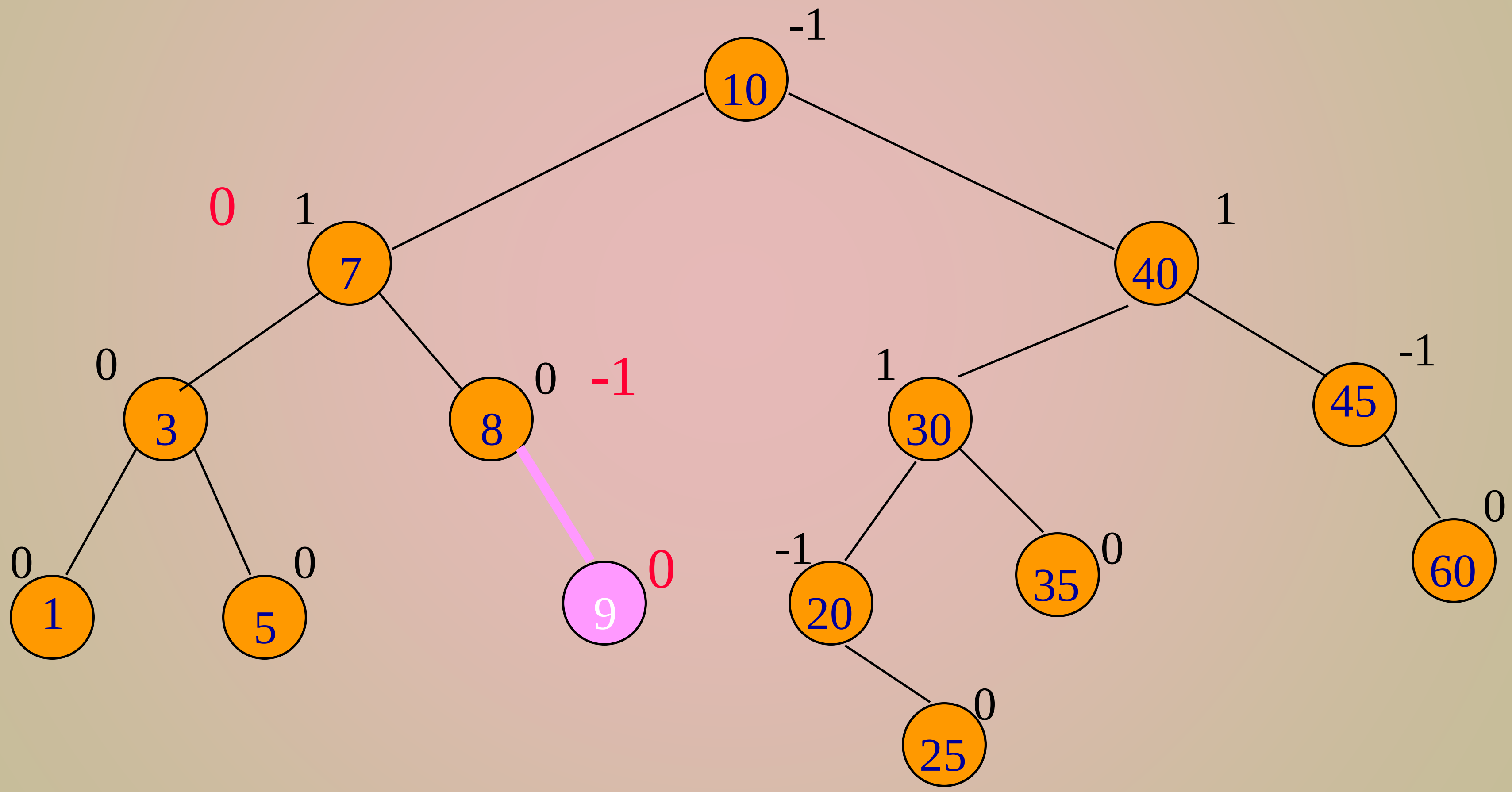
بنابراین:

$$h \leq 1.441 * \log_2(n+2)$$

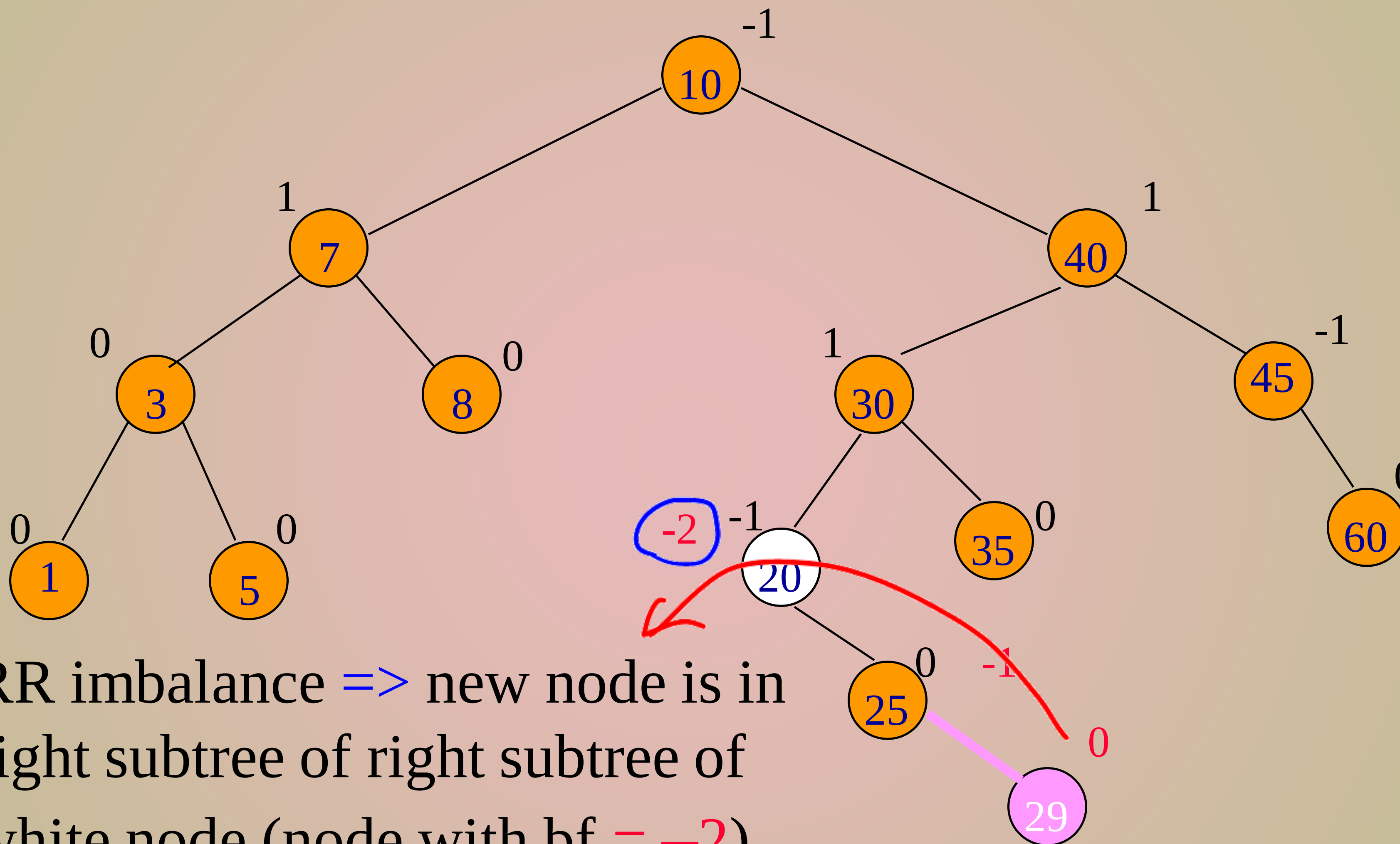
Example AVL Tree



put(9)

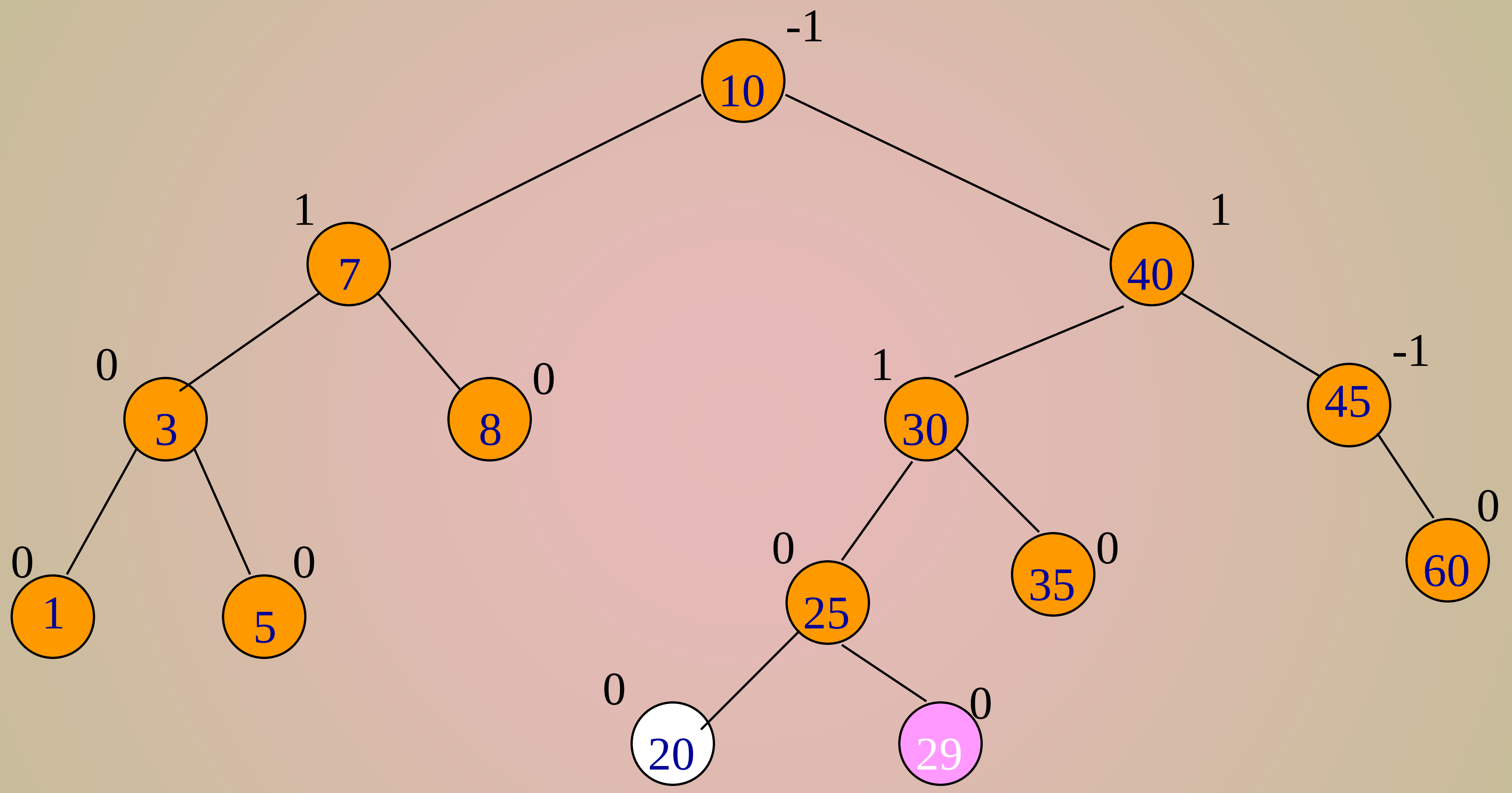


put(29)



RR imbalance => new node is in right subtree of right subtree of white node (node with bf = –2)

put(29)



RR rotation.

Insert/Put

- Following insert/put, 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.

A-Node

فرض خلف، ابتدا bf برداشت => چون bf تغییر کرده است =>
bf جدید +2 یا 0 است. ... به تناقض می‌رسیم.

- Let A be the nearest ancestor of the newly inserted node whose balance factor becomes +2 or –2 following the insert.
- Balance factor of nodes between new node and A is 0 before insertion. A 0

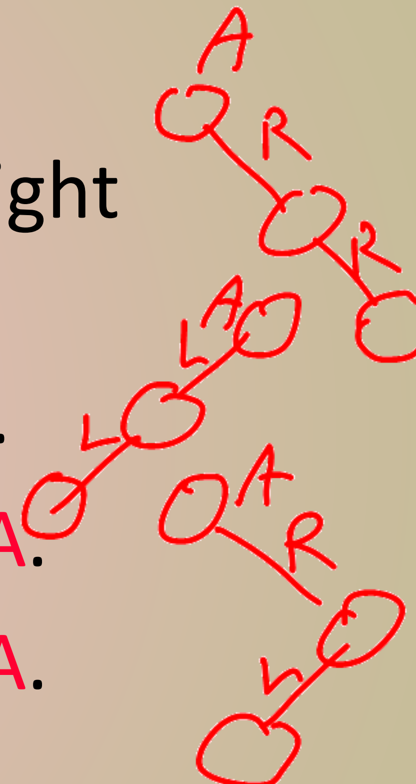
ارتفاع زیردرخت مربوط به تمام گره‌های بین گره درج شده
و گره A تغییر کرده است => bf تمام این گره‌ها
تغییر کرده است.

0 درج شده

–1 +1

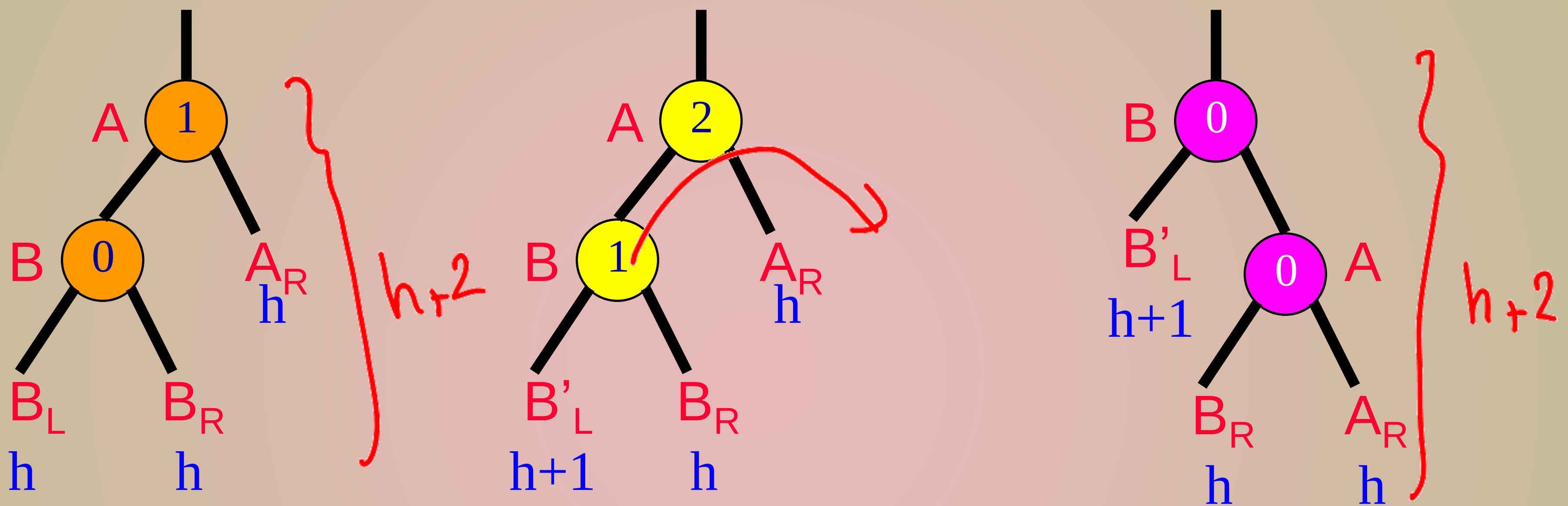
Imbalance Types

- 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.



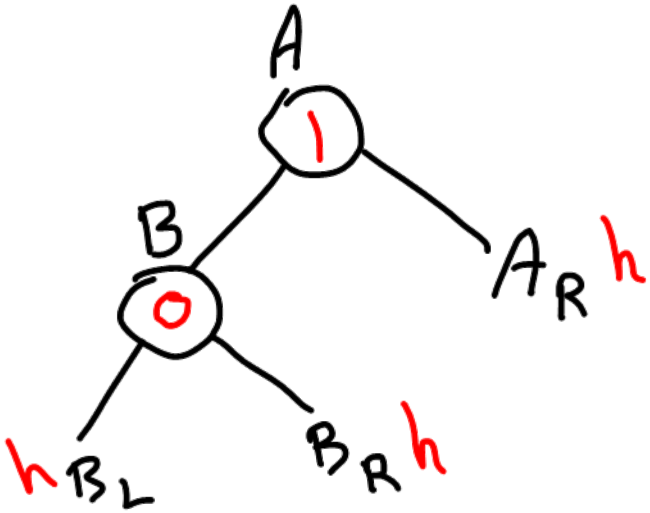
LR

LL Rotation



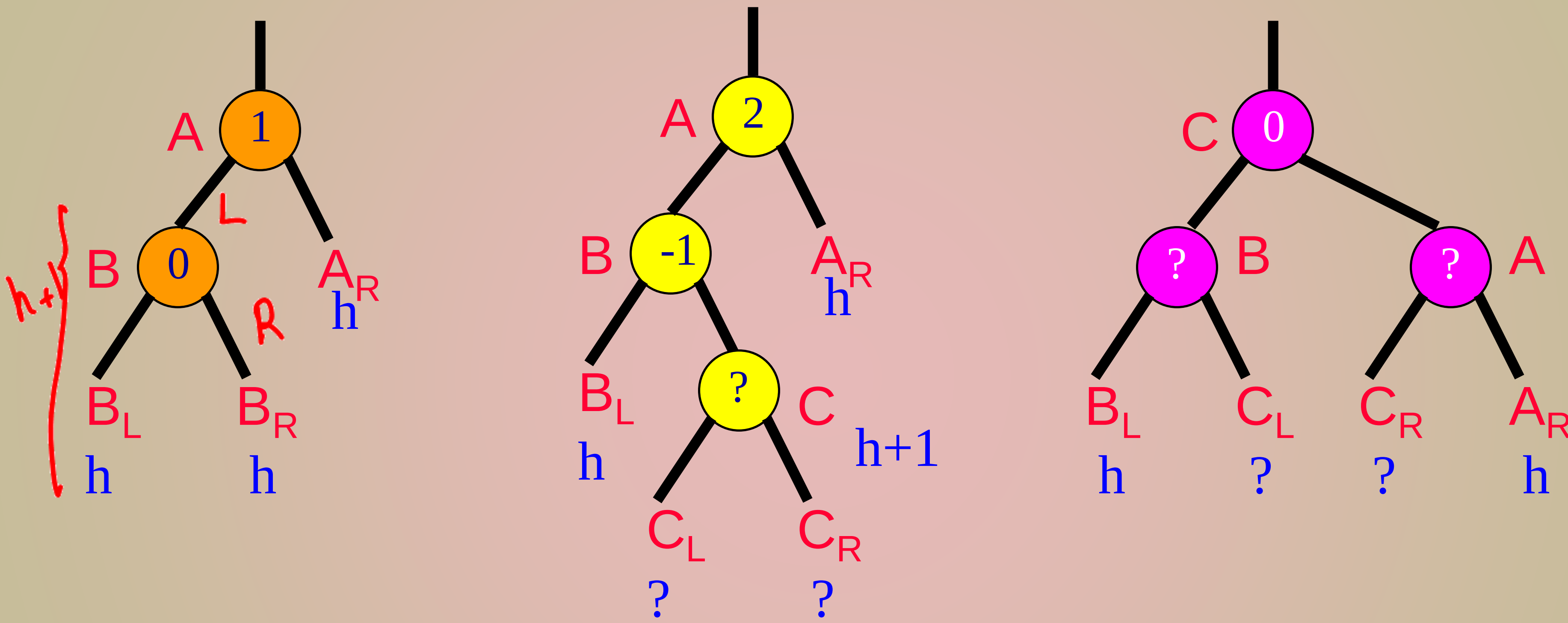
Before insertion. After insertion. After rotation.

- Subtree height is unchanged.
- No further adjustments to be done.


A
1
B
0
A_R h
h B_L
B_R h

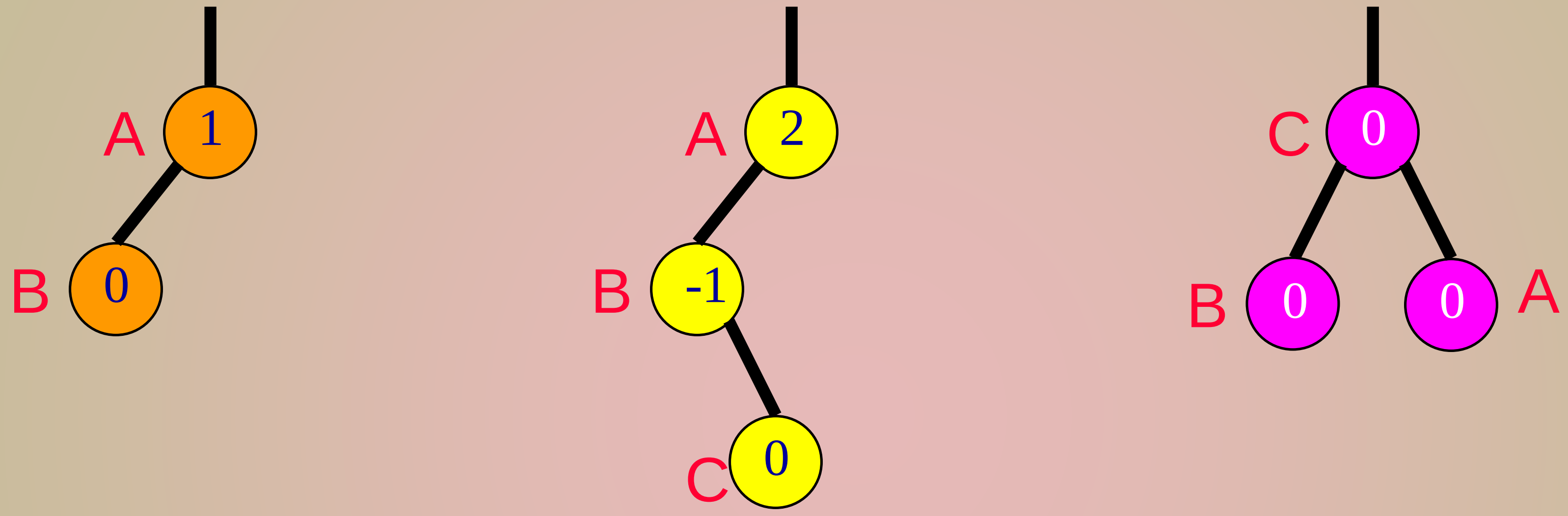

2

LR Rotation (all 3 cases)



- Subtree height is unchanged.
- No further adjustments to be done.

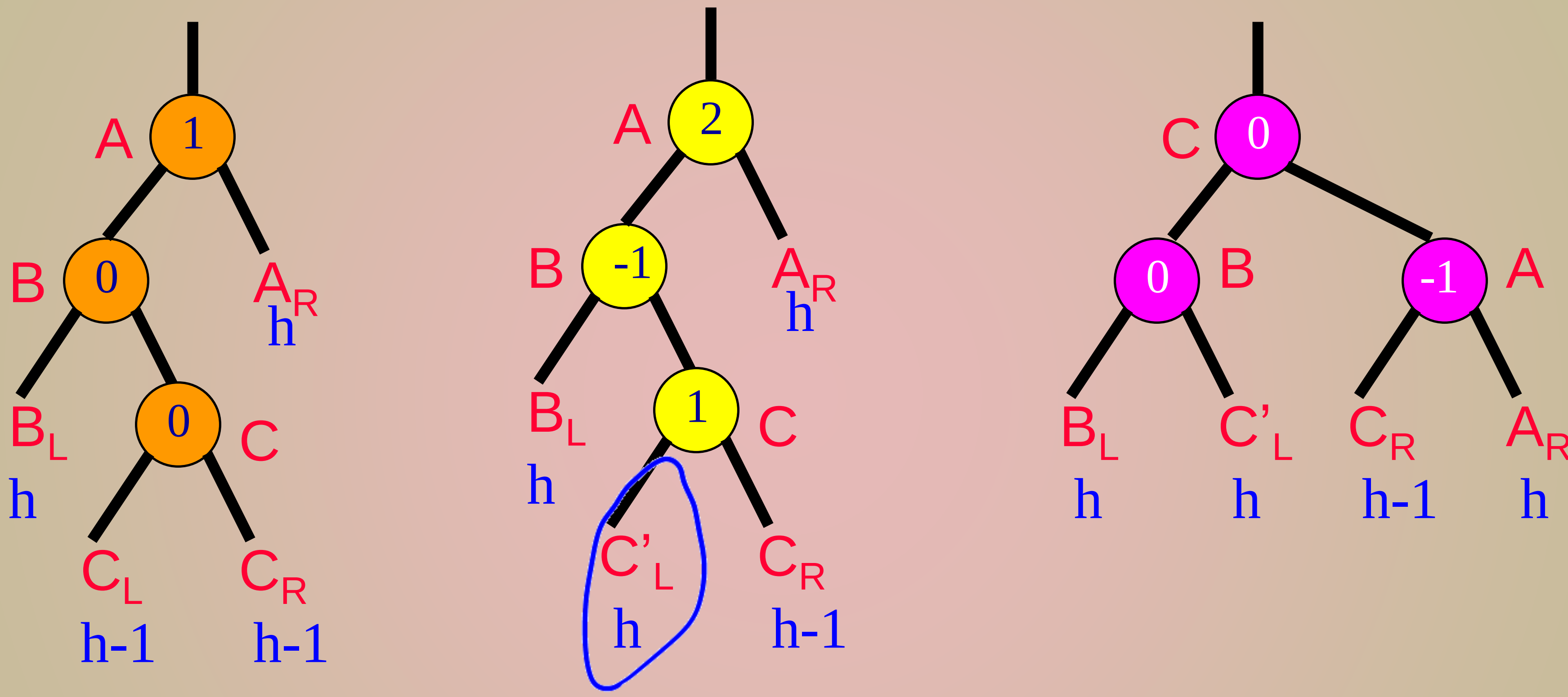
LR Rotation (case 1)



Before insertion. After insertion. After rotation.

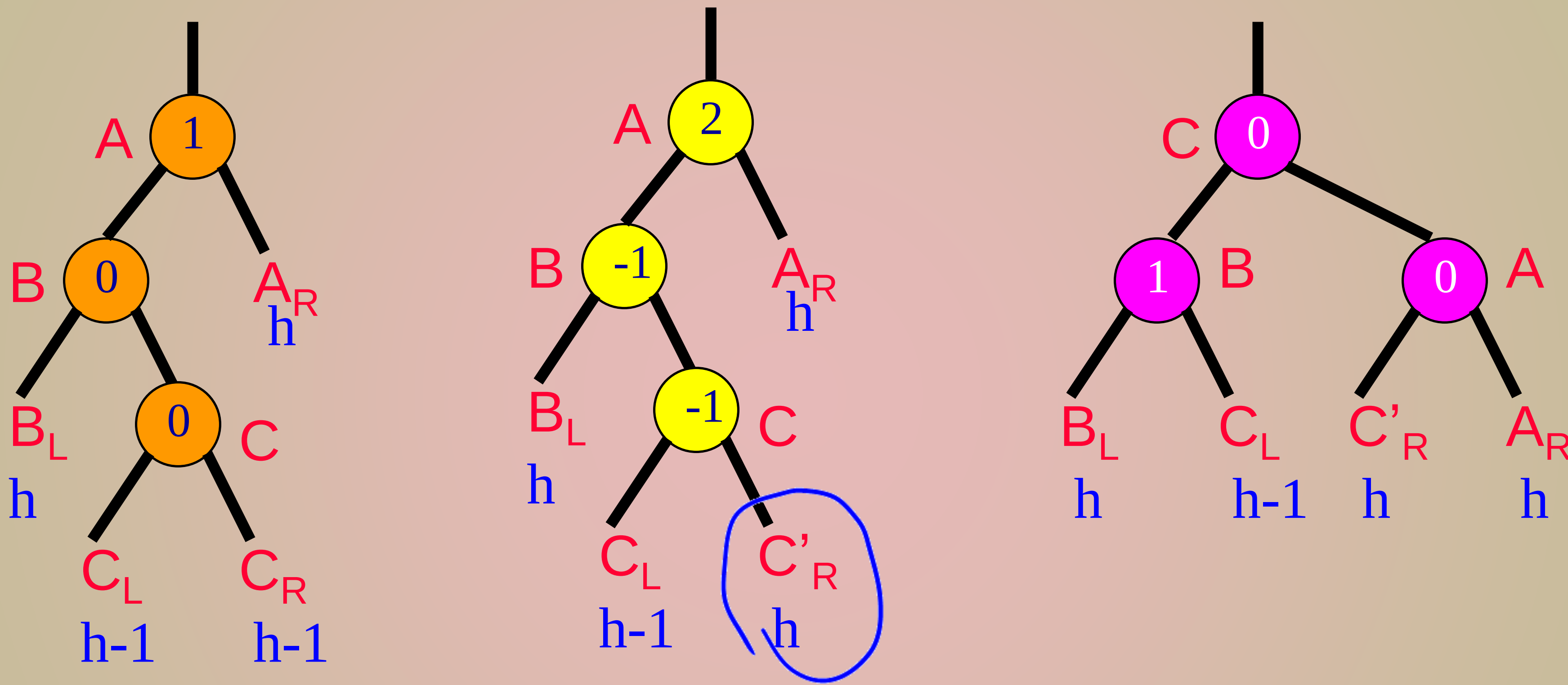
- Subtree height is unchanged.
- No further adjustments to be done.

LR Rotation (case 2)



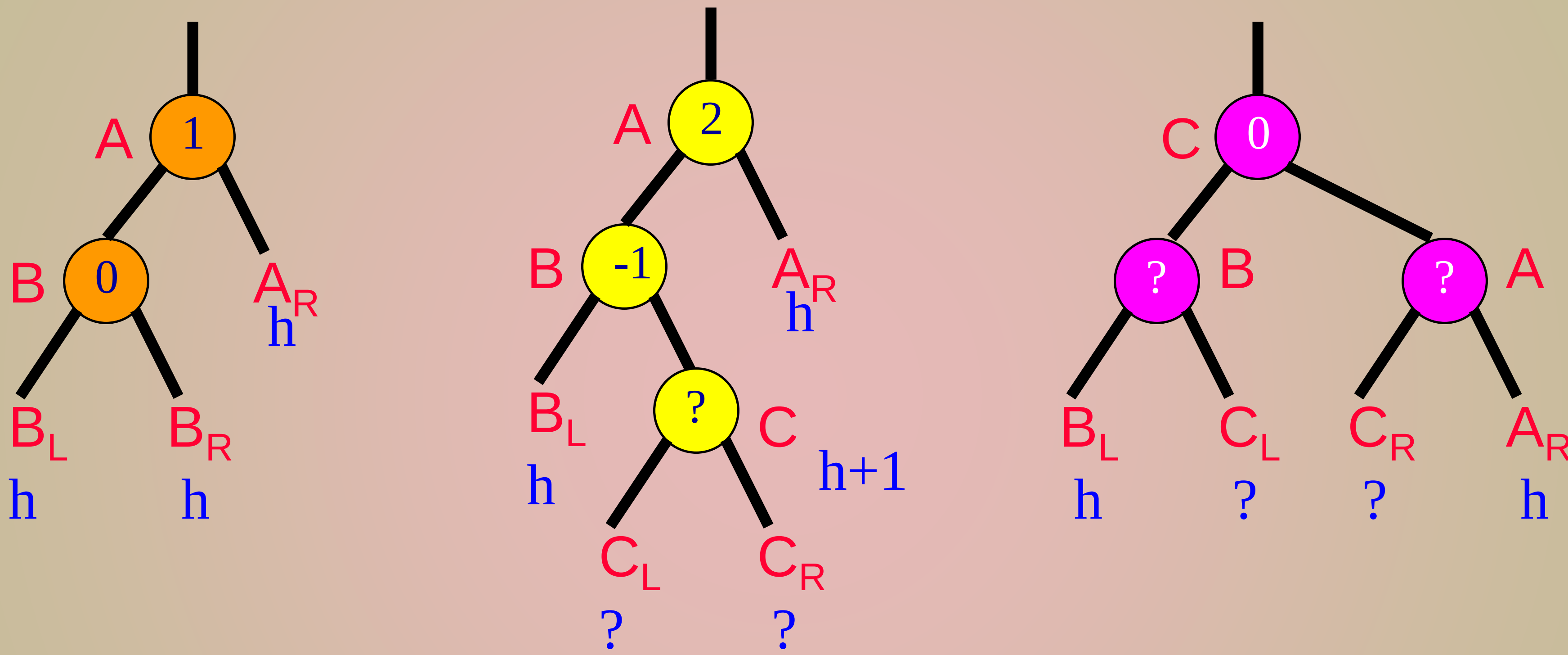
- Subtree height is unchanged.
- No further adjustments to be done.

LR Rotation (case 3)



- Subtree height is unchanged.
- No further adjustments to be done.

LR Rotation (all 3 cases)

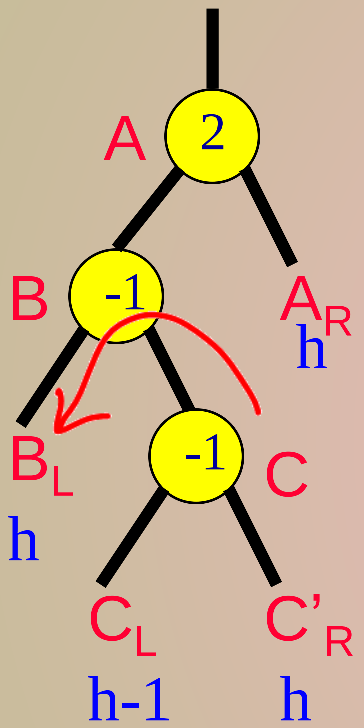


- Subtree height is unchanged.
- No further adjustments to be done.

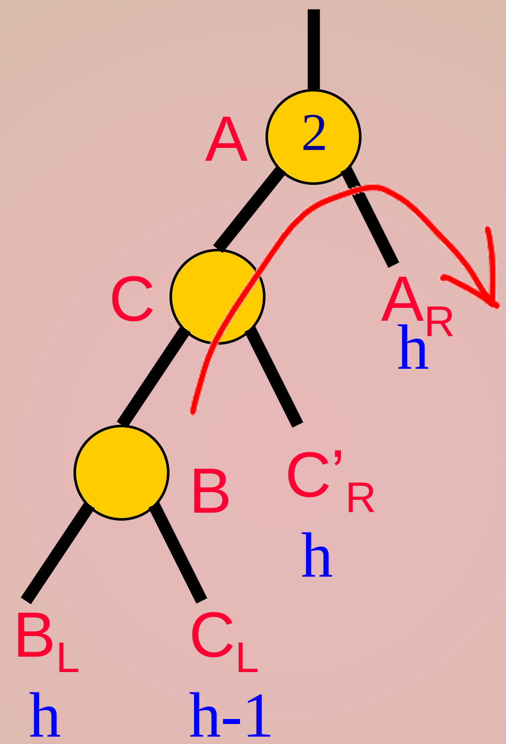
Single & Double Rotations

- Single
  - LL and RR
- Double
  - LR and RL
  - LR is RR followed by LL
  - RL is LL followed by RR

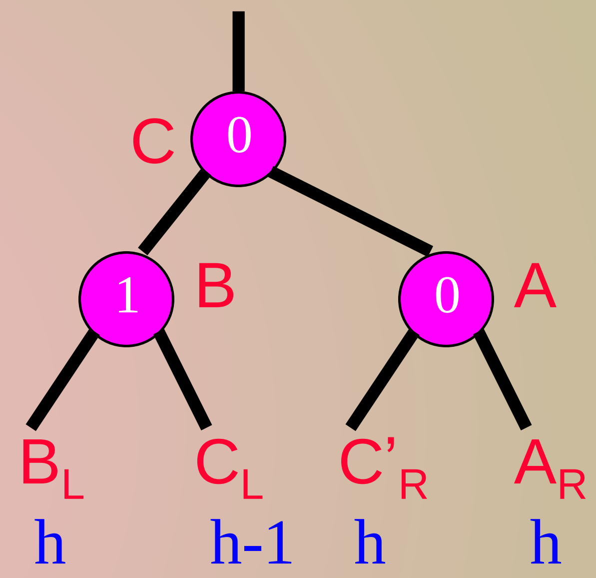
LR$_A$ Is RR$_B$ + LL$_A$



After insertion.

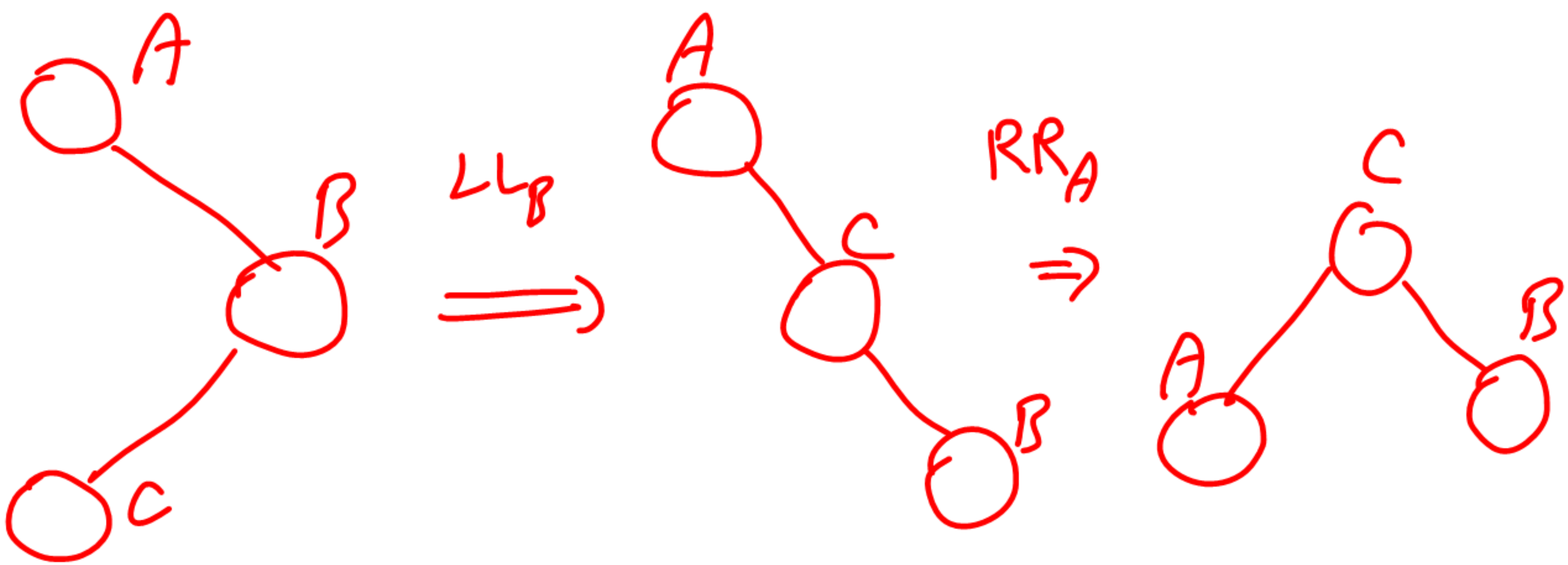


After RR rotation.



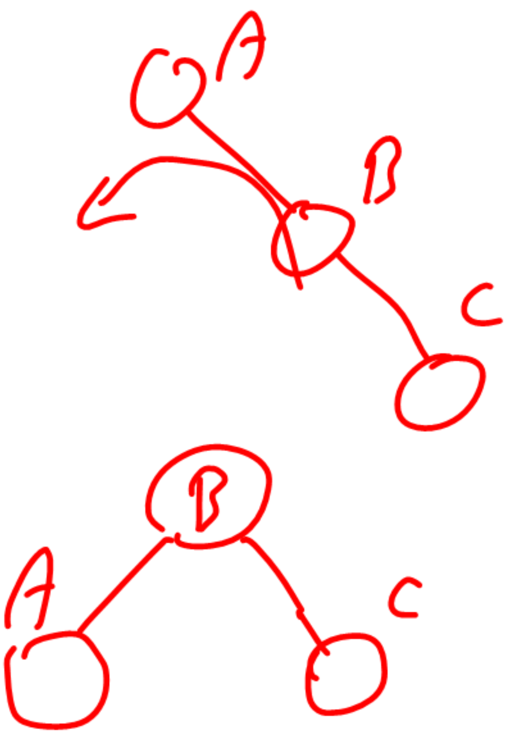
After LL rotation.

$$RL_A = LL_B + RR_A$$

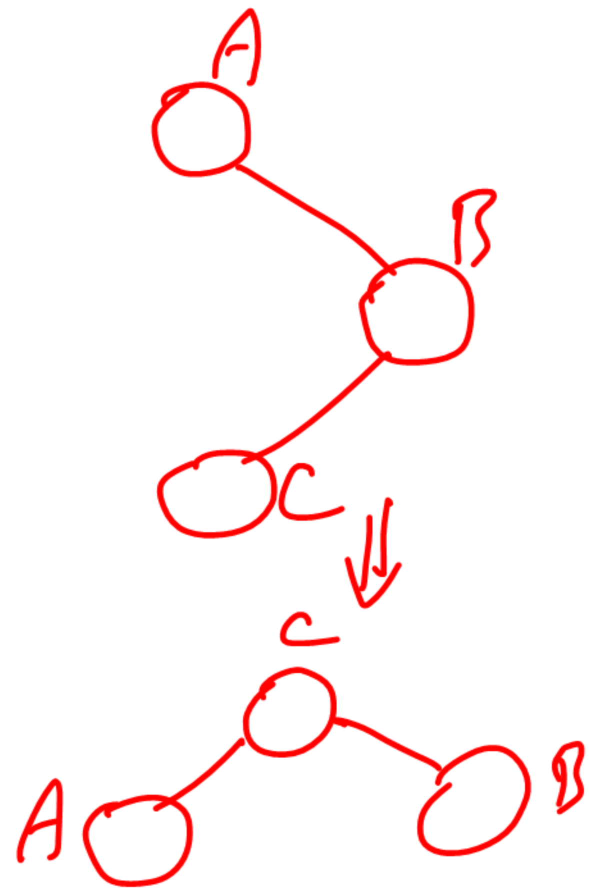


LL و LR گفته شد

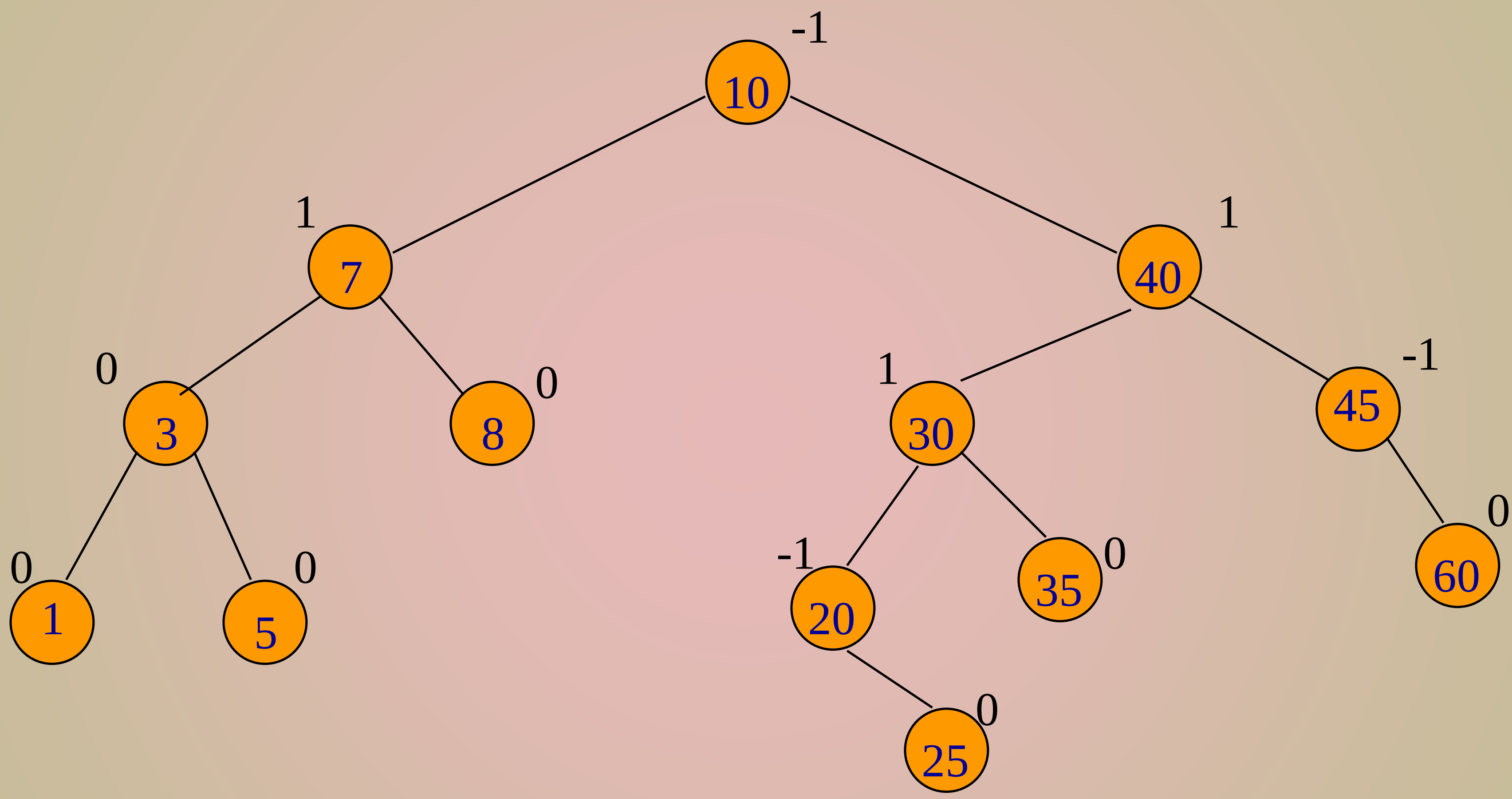
RR



RL

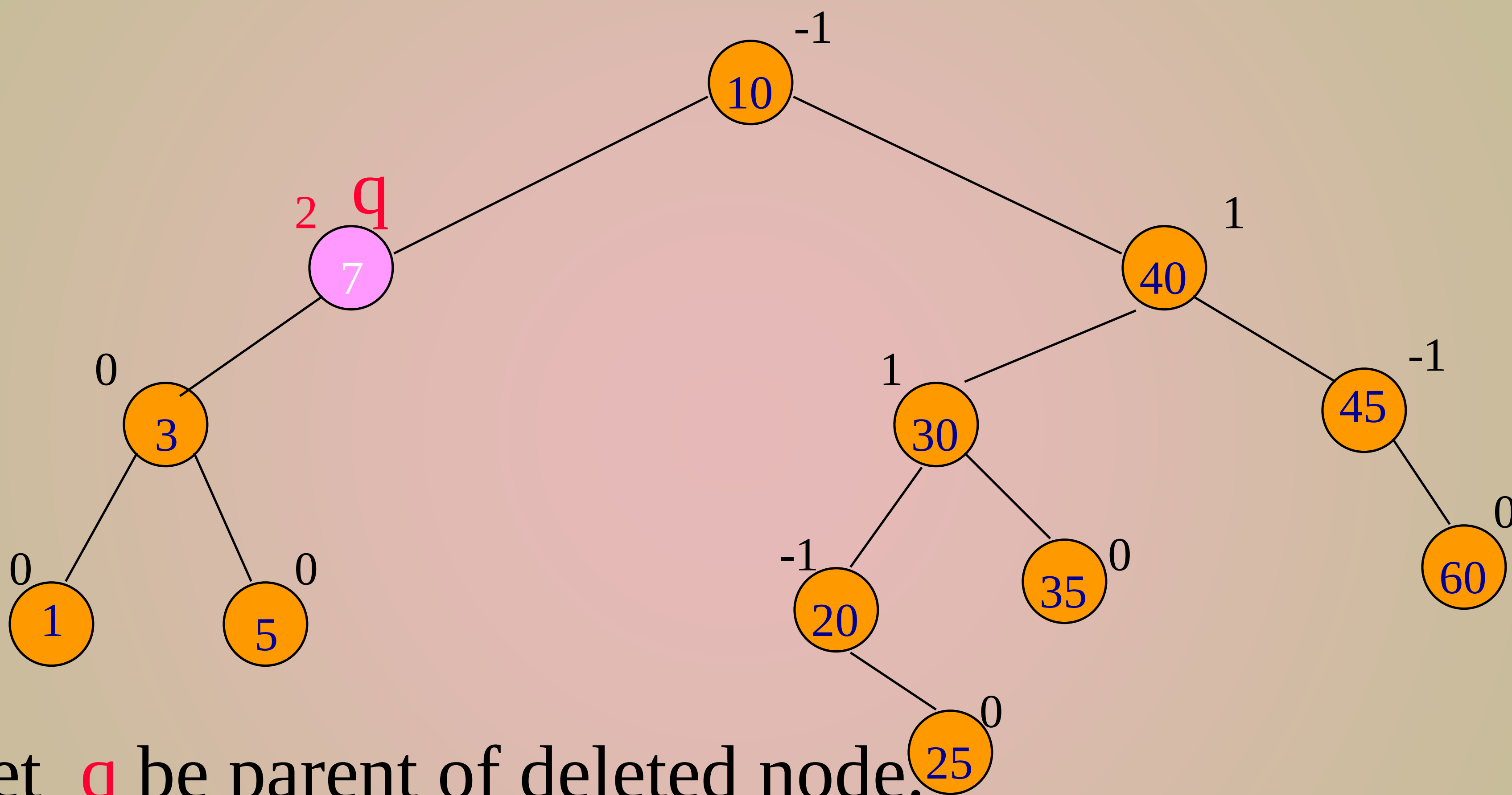


Remove An Element



Remove 8.

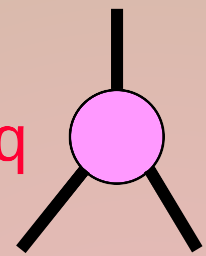
Remove An Element



- Let q be parent of deleted node.
- Retrace path from q towards root.

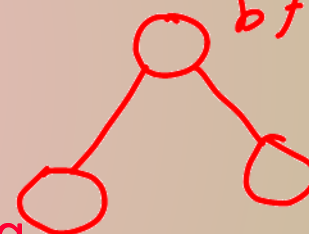
New Balance Factor Of q

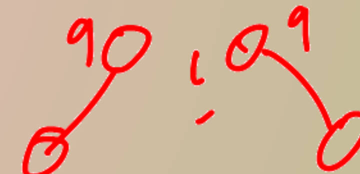
دقیقاً پدر گره حذف شده ← q



- Deletion from left subtree of q => bf--.
- Deletion from right subtree of q => bf++.
- New balance factor = 1 or –1 => ?
  - => no change in height of subtree rooted at q.
- New balance factor = 0 => ?
  - => height of subtree rooted at q has decreased by 1.
- New balance factor = 2 or –2 => ?
  - => tree is unbalanced at q.

قبلاً bf = 0



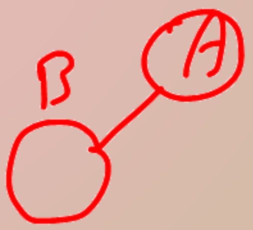


Imbalance Classification

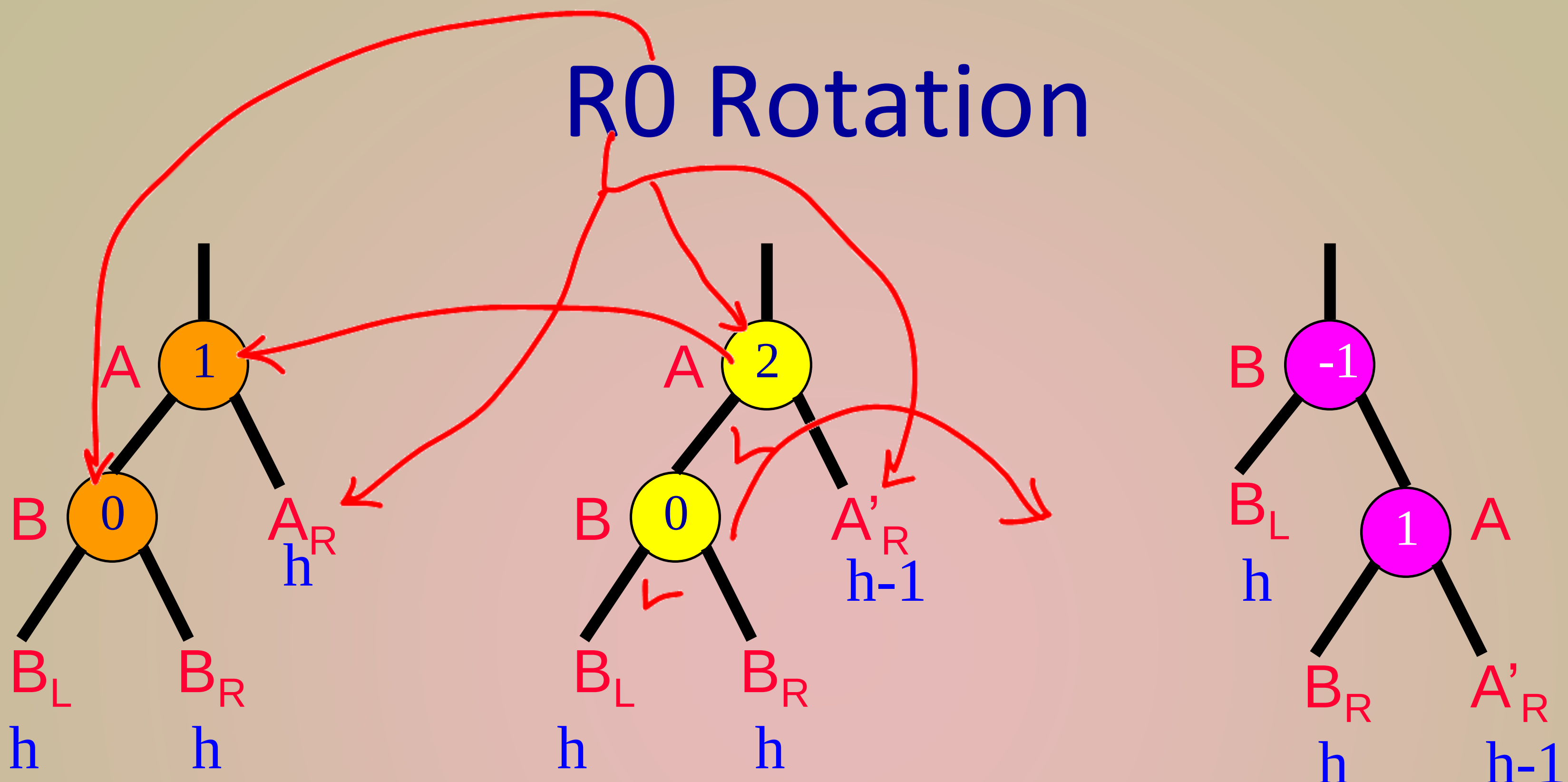
- Let A be the nearest ancestor of the deleted node whose balance factor has become 2 or –2 following a deletion.
- Deletion from left subtree of A => type L.
- Deletion from right subtree of A => type R.
- Type R => new bf(A) = 2.

A bf(A) = 2
new

- So, old bf(A) = 1.
- So, A has a left child B.
  - bf(B) = 0 => R0.
  - bf(B) = 1 => R1.
  - bf(B) = –1 => R-1.



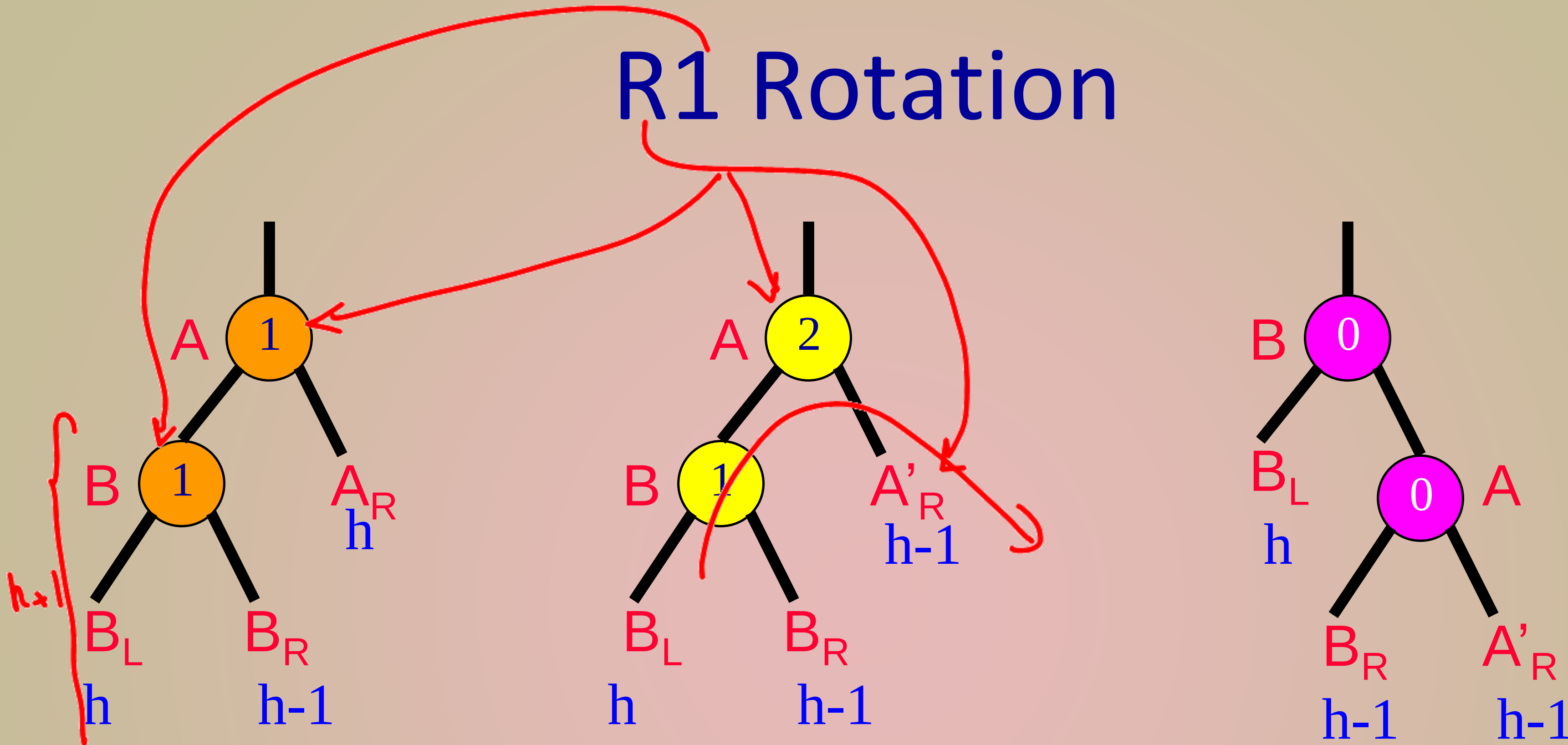
R0 Rotation



Before deletion. After deletion. After rotation.

- Subtree height is unchanged.
- No further adjustments to be done.
- Similar to LL rotation.

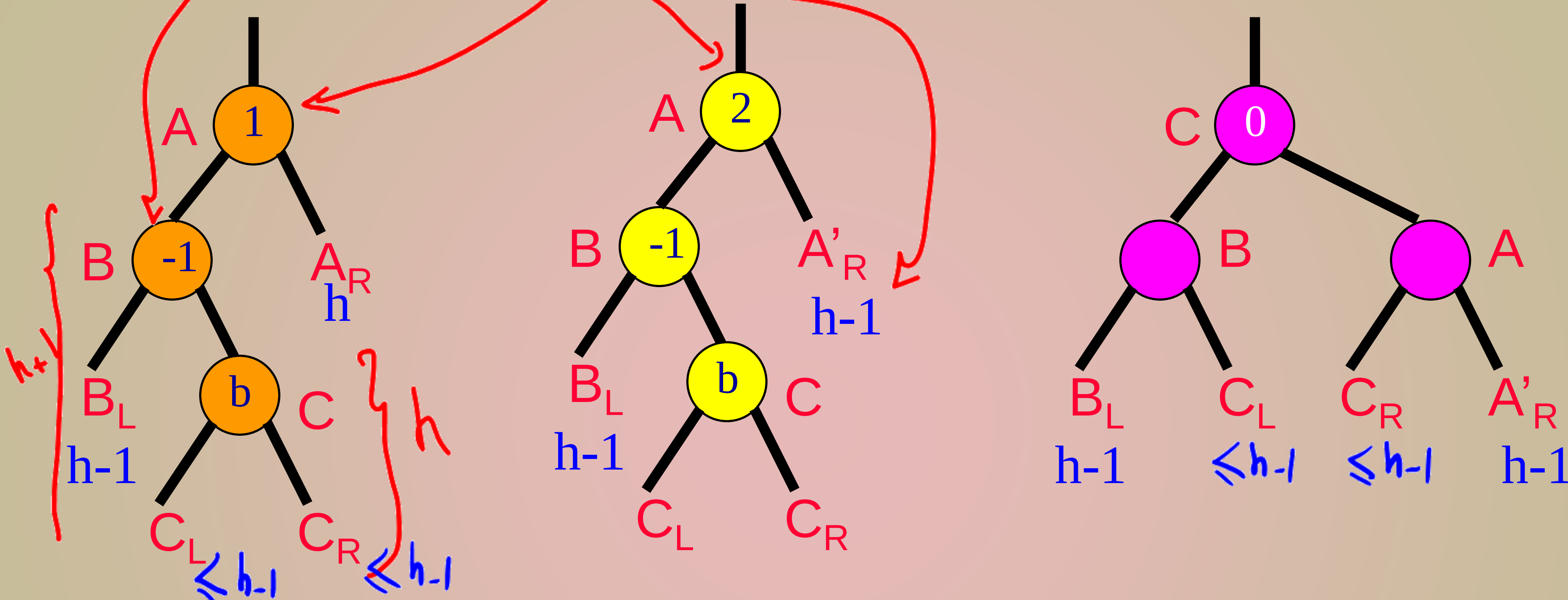
R1 Rotation



Before deletion. After deletion. After rotation.

- Subtree height is reduced by 1.
- Must continue on path to root.
- Similar to LL and R0 rotations.

R-1 Rotation



- New balance factor of A and B depends on b.
- Subtree height is reduced by 1.
- Must continue on path to root.
- Similar to LR.

R-1	R1	R0
LR	LL	LL
	Right	Right
L1	L-1	L0
RL	RR	RR
	Left	Left

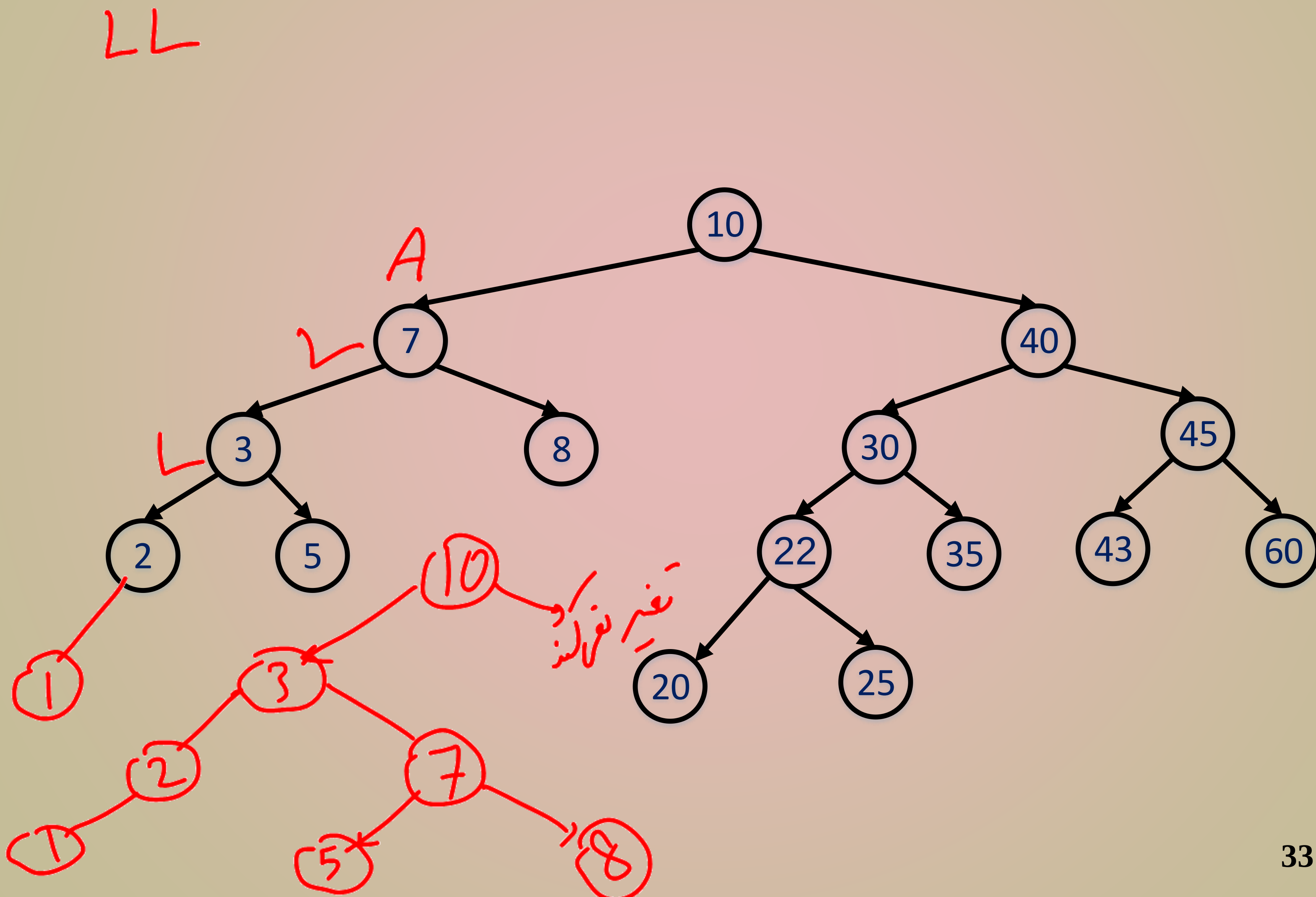
Number Of Rebalancing Rotations

- At most 1 for an insert.
- O(log n) for a delete.

Rotation Frequency

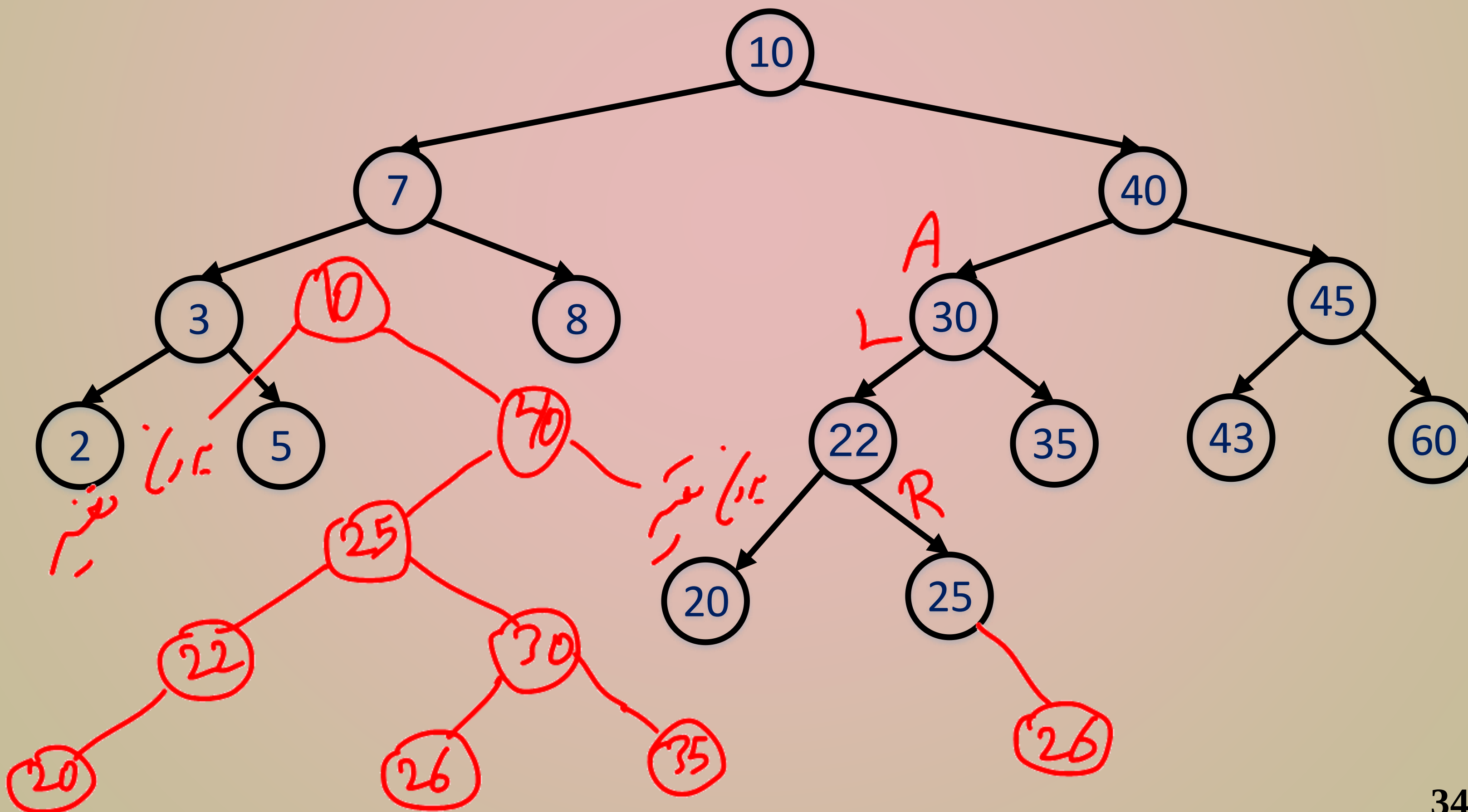
- Insert random numbers.
  - No rotation … 53.4% (approx).
  - LL/RR … 23.3% (approx).
  - LR/RL … 23.2% (approx).

نمونه سوال امتحان: درج ۱

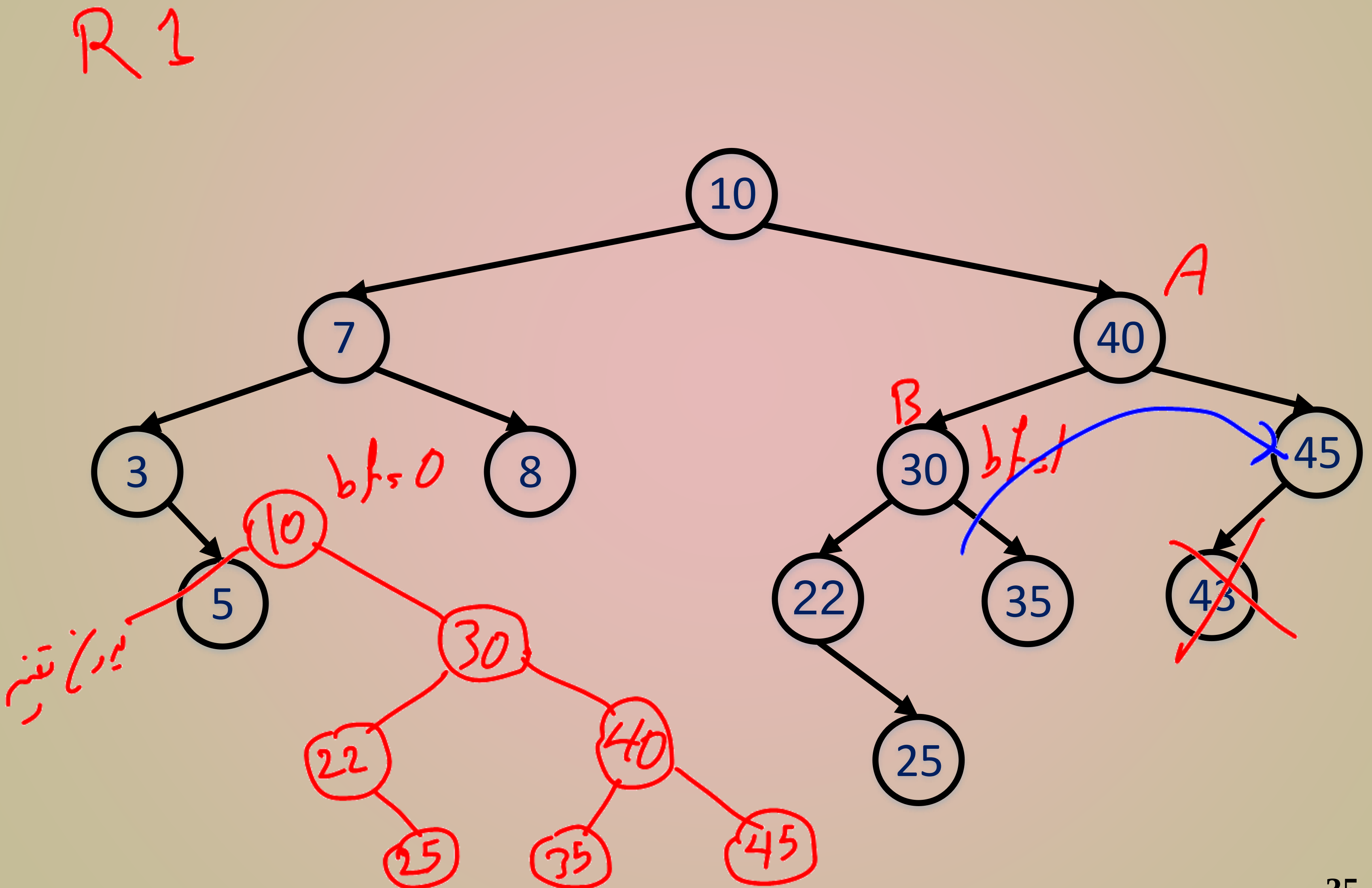


نمونه سوال امتحان: درج ۲۶

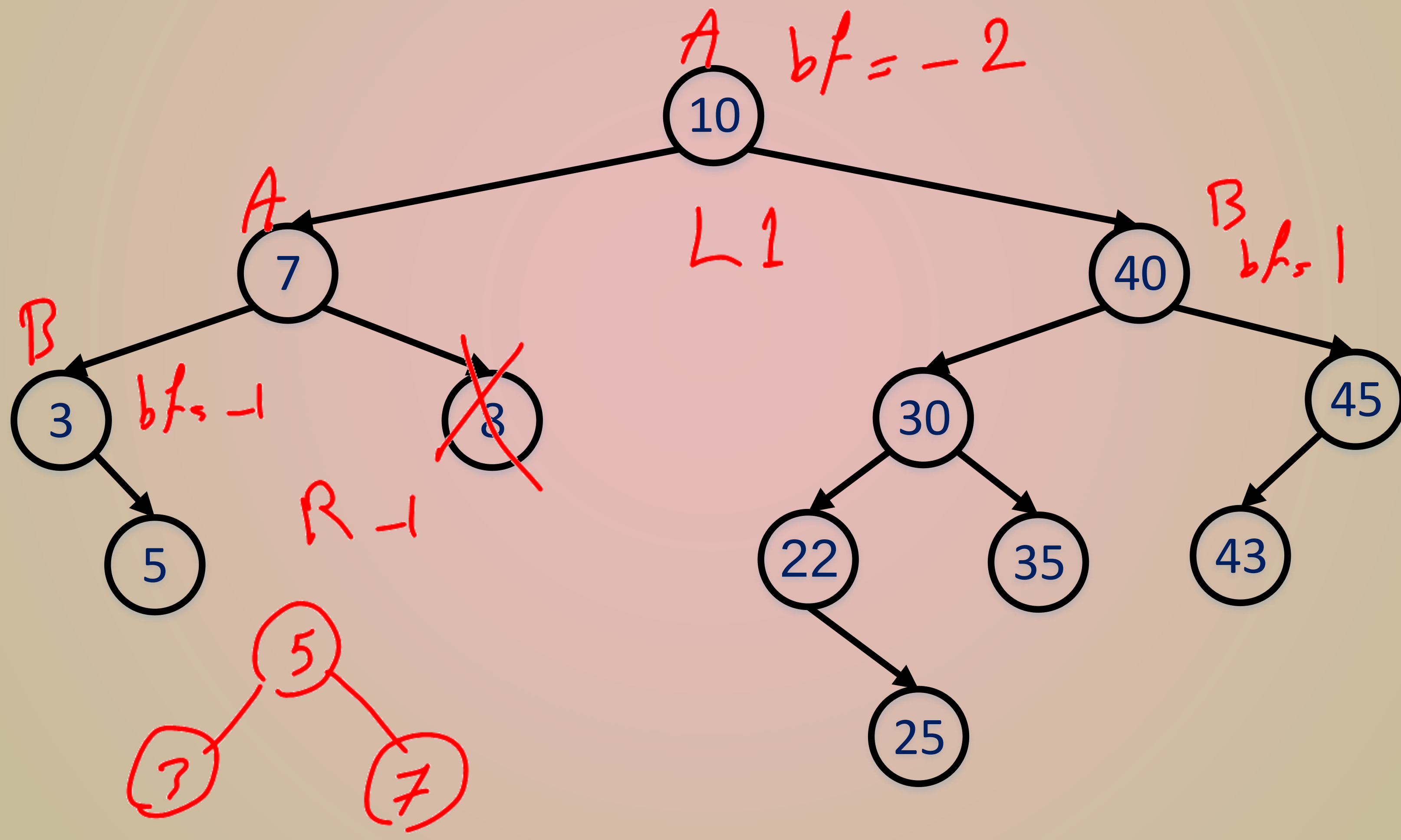
LR

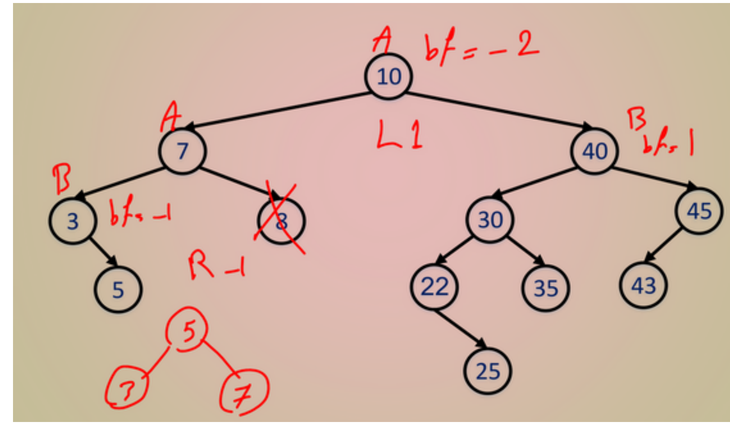


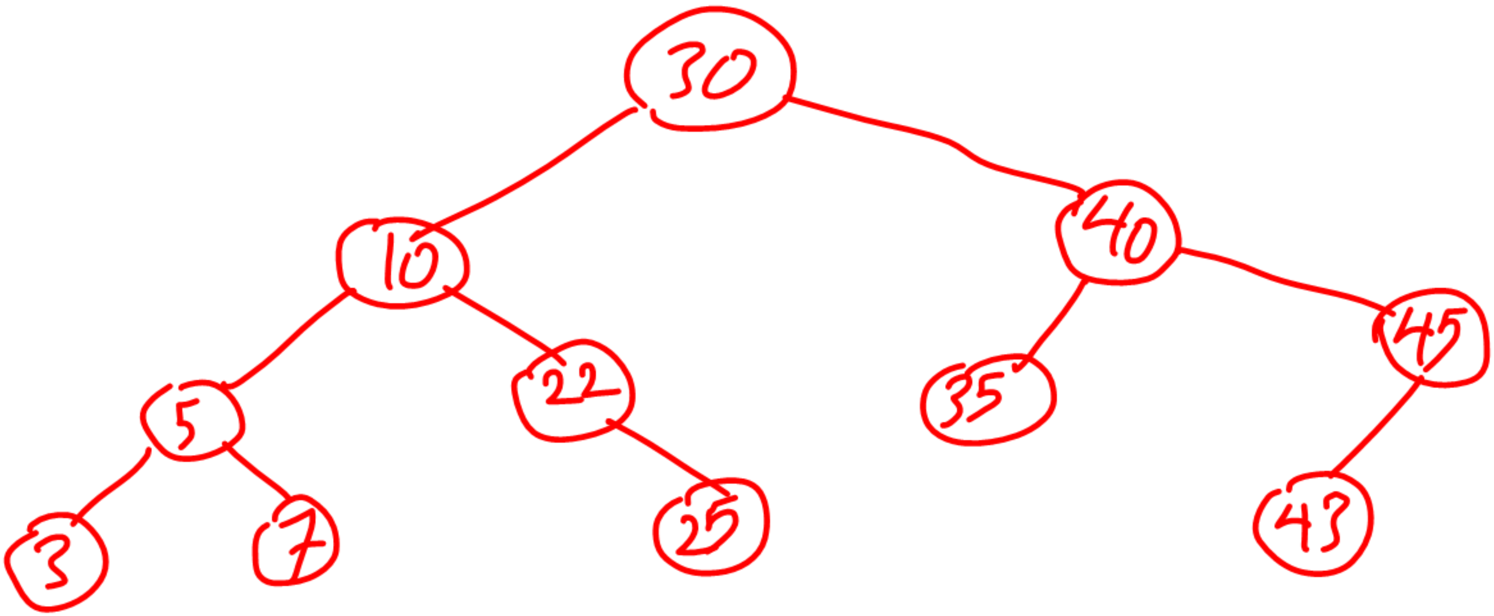
نمونه سوال امتحان: حذف ۴۳



نمونه سوال امتحان: حذف ۸




A bf = -2
10
A
7
L 1
B bf = 1
40
B
3
bf = -1
8
30
45
R -1
5
22
35
43
5
3
7
25


30
10
40
5
22
35
45
3
7
25
43

پاسخ نهایی