Διάλεξη 13: Δέντρα ΙΙΙ - Ισοζυγισμένα Δέντρα, AVL Δέντρα

Στην ενότητα αυτή θα μελετηθούν τα εξής επιμέρους θέματα:

- Ισοζυγισμένα Δέντρα

- Υλοποίηση AVL-δέντρων

- Εισαγωγή Κόμβων και Περιστροφές σε AVL δέντρα

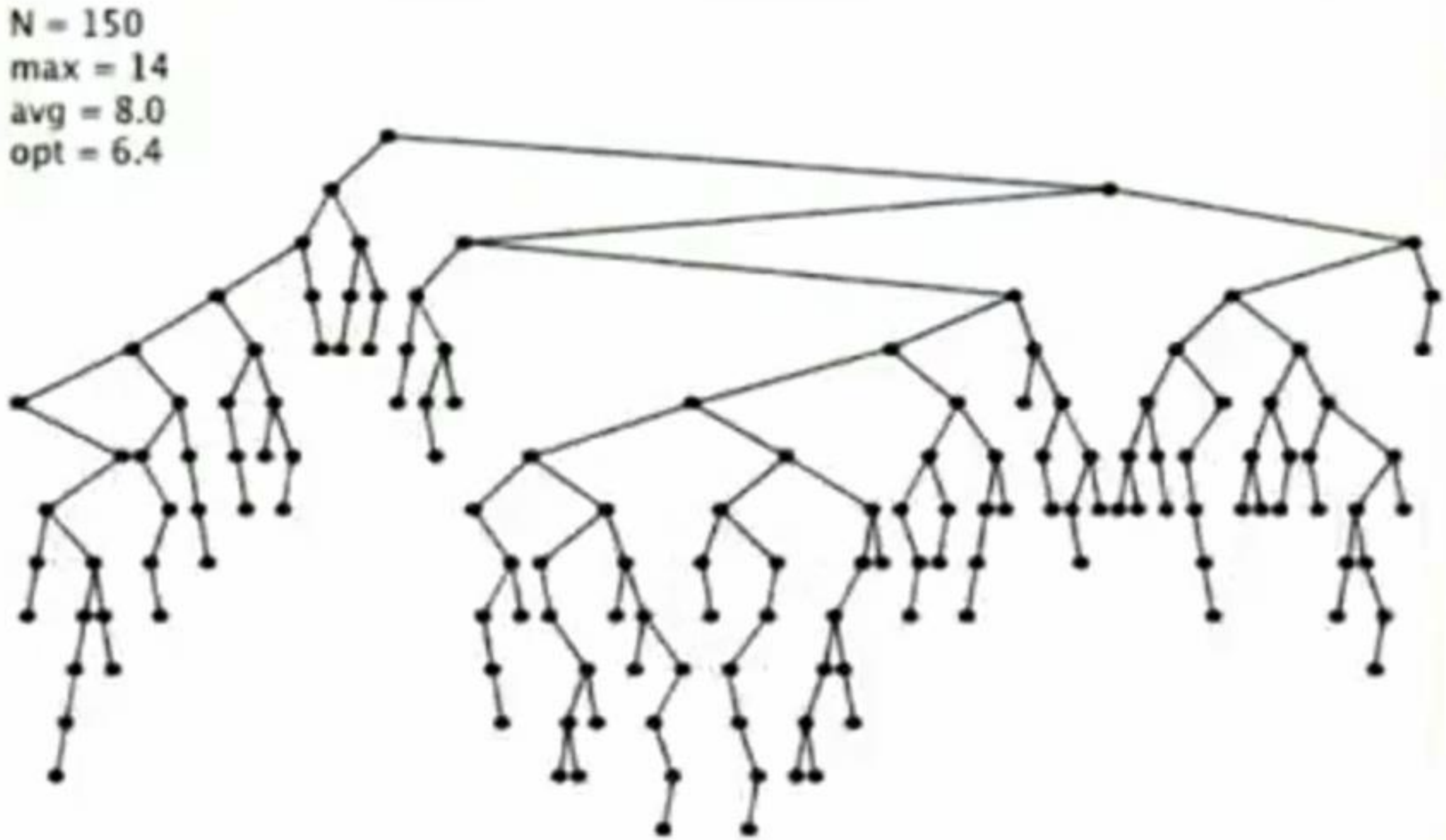
Διαγραφή σε BST (Hibbart Deletion)

```
public void deleteMin()
{
  if (isEmpty()) return;
  root = deleteMin(root);
}

private Node deleteMin(Node x) {
  if (x.left == null)
    return x.right;
  x.left = deleteMin(x.left);
  return x;
}


public void delete(Key key) {
  root = delete(root, key);
}
```

```
private Node delete(Node x, Key key) {
  if (x == null)
     return null;
  int cmp = key.compareTo(x.key);
  if (cmp < 0)
     x.left = delete(x.left, key);
  else if (cmp > 0)
     x.right = delete(x.right, key);
  else {
     if (x.right == null)
        return x.left;
     if (x.left == null)
        return x.right;
     Node t = x;
     x = min(t.right);
     x.right = deleteMin(t.right);
     x.left = t.left;
  }
  return x;
}
```

Code from: https://algs4.cs.princeton.edu/32bst/BST.java.html

Διαγραφή στα ΔΔΑ



Σενάριο: Συνεχής διαγραφή + εισαγωγή στοιχείων σε δυναμικό περιβάλλον
Το ύψος του ΔΔΑ τείνει στο $\sqrt{n}$

Εισαγωγή

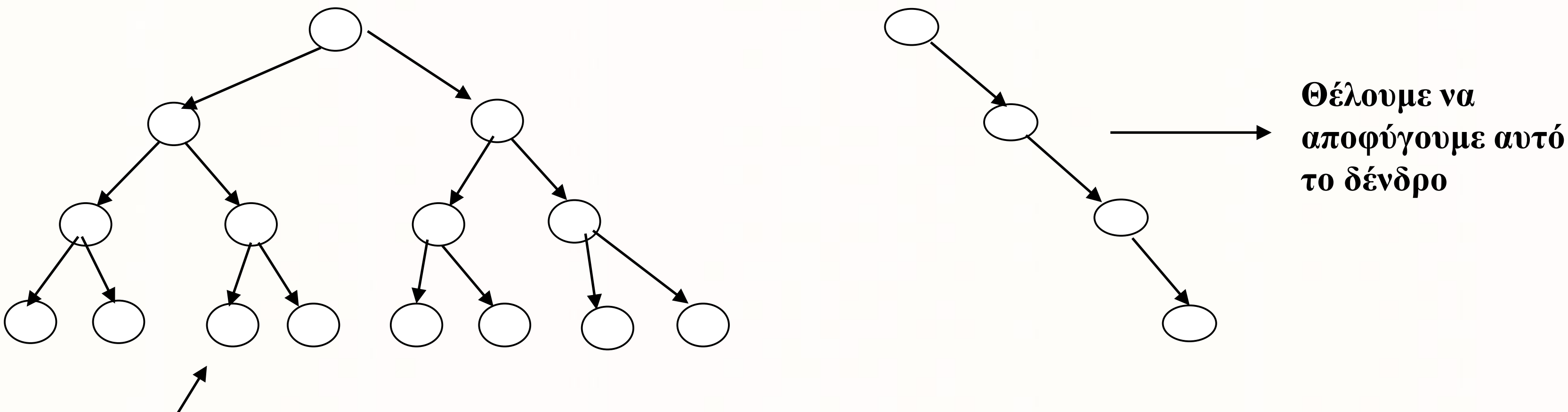
- Στην προηγούμενη διάλεξη μιλήσαμε για Δυαδικά Δένδρα Αναζήτησης (ΔΔΑ).

- Αυτά έχουν **<u>ύψος</u>** ίσο με $\log_2 n$ (στην καλύτερη περίπτωση) και n-1 στην χειρότερη περίπτωση.

- Άρα για να βρούμε αν υπάρχει ένα στοιχείο στο δένδρο (δηλαδή για να κάνουμε μια αναζήτηση) χρειαζόμαστε να περάσουμε από $\mathbf{\log_2 n}$ (καλύτερη περίπτωση) ή **n** στοιχεία (χειρότερη περίπτωση).

Ιδέα Λύσης

- Βασικά θέλουμε να περιορίσουμε το ύψος του δένδρου όσο το δυνατό περισσότερο. Αυτό μπορεί να γίνει με:
  1. Σωστή αξιοποίηση όλων των παιδιών.



Το **τέλειο** δένδρο είναι η ιδανική περίπτωση!

2. Να αυξήσουμε τον αριθμό των παιδιών σε κάθε κόμβο χωρίς να αυξηθεί πάρα πολύ. Γιατί;

   Διότι θα καταλήξουμε σε ένα πίνακα στο τέλος οπότε η αναζήτηση κάποιου στοιχείου θα πάρει O(n) χρόνο

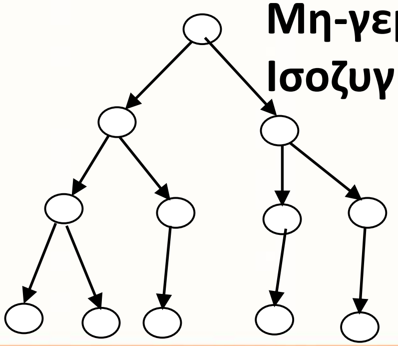
Ισοζυγισμένα Δένδρα

- Ένα τέλειο δένδρο προϋποθέτει ότι υπάρχει ο κατάλληλος αριθμός κόμβων. π.χ. ένα τέλειο δυαδικό δένδρο πρέπει να έχει

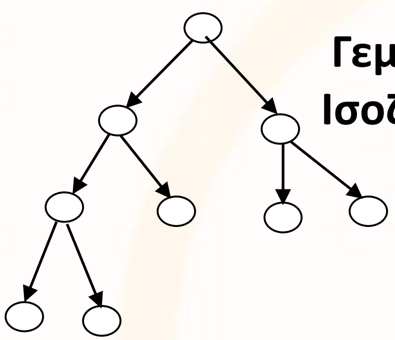
  1 ή 2 ή 4 ή 8 …. κόμβους
- Για αυτό περιοριζόμαστε στο να κρατάμε το δένδρο **<u>ισοζυγισμένο</u>** αντί τέλειο.

<u>Ισοζυγισμένο Δένδρο (Balanced Tree)</u>

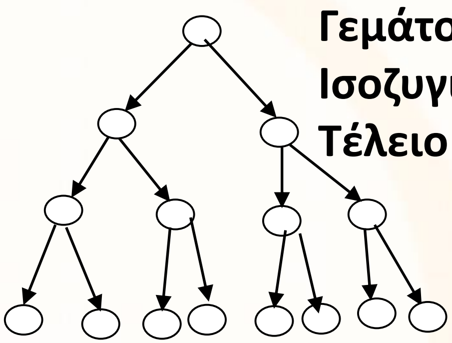
Ένα δένδρο στο οποίο όλα τα **φύλλα** έχουν το ίδιο **βάθος**.



Μη-γεμάτο, Ισοζυγισμένο



Γεμάτο, Μη-Ισοζυγισμένο



Γεμάτο, Ισοζυγισμένο, Τέλειο

AVL Δένδρα

- Είναι δυνατό να οργανώσουμε ένα δυαδικό δένδρο αναζήτησης έτσι ώστε το ύψος του να είναι το μικρότερο δυνατό; (τάξεως $O(\log_2 n)$)

- **Ιδέα:** για να έχουμε μικρό ύψος, αν u είναι ένας κόμβος του δένδρου τότε και τα δύο του υπόδενδρα έχουν περίπου τον ίδιο αριθμό κόμβων.

1. **Πρώτη προσπάθεια:** για κάθε κόμβο και τα δυο του υπόδενδρα έχουν το ίδιο ύψος… ⇒ τέλεια δένδρα ⇒ δένδρα με αριθμό κόμβων 2h+1-1.

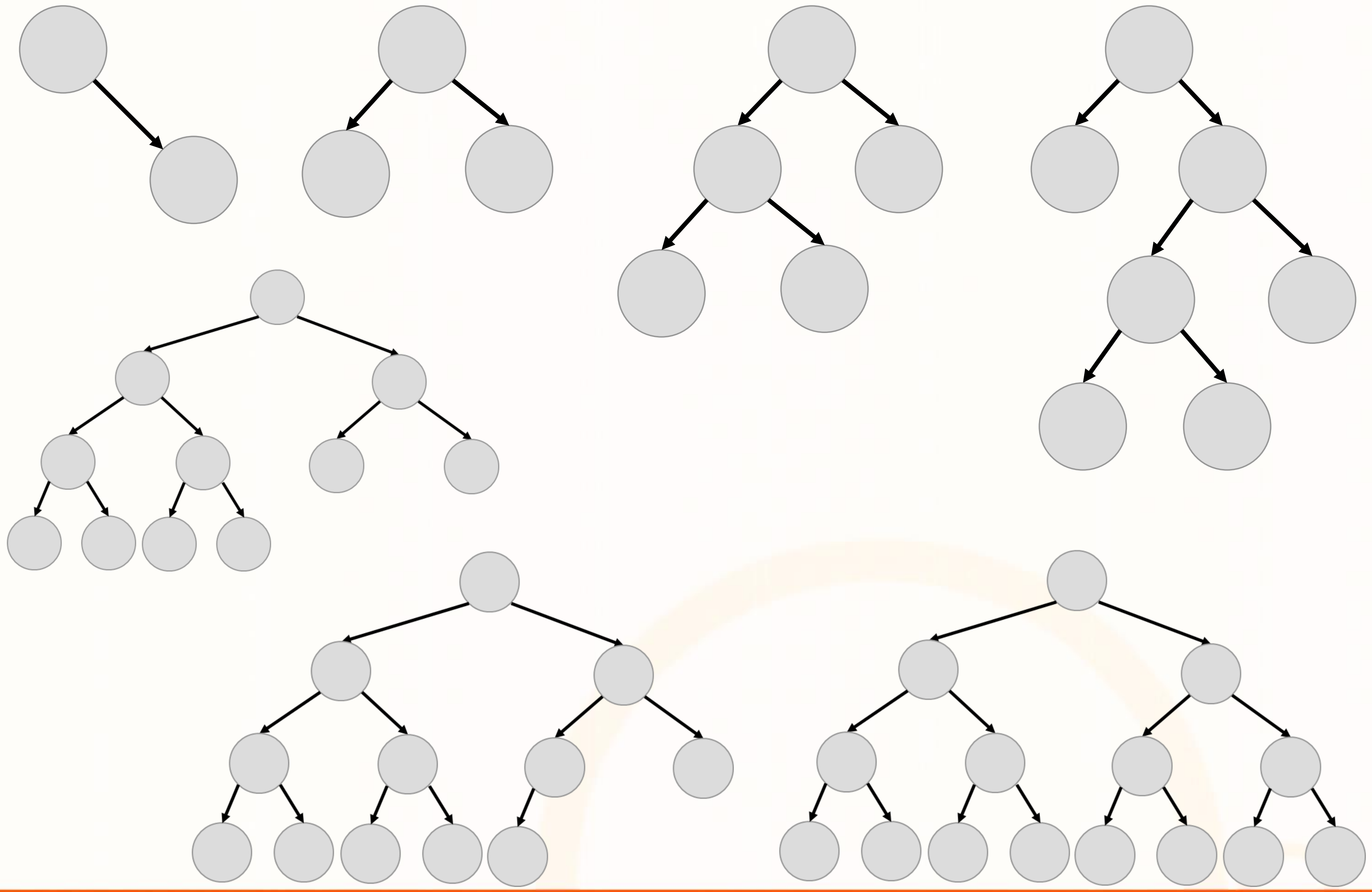
AVL Δένδρα

2. **Δεύτερη προσπάθεια:** Ένα δυαδικό δένδρο είναι **AVL-δένδρο** (Adelson-Velskii and Landis) αν για κάθε κόμβο του u τα ύψη των παιδιών του u διαφέρουν το πολύ κατά 1.
   (υποθέτουμε το ύψος του κενού δένδρου=–1)

Θα δούμε ότι:

- το ύψος ενός AVL-δένδρου με n κόμβους είναι O(log n)
- διαδικασίες εισαγωγής και εξαγωγής κόμβων μπορούν να διατυπωθούν έτσι ώστε η AVL-συνθήκη να διατηρείται.
- Ένα πλήρες δένδρο είναι και AVL-δένδρο.

Παραδείγματα AVL δένδρων ... ή όχι;

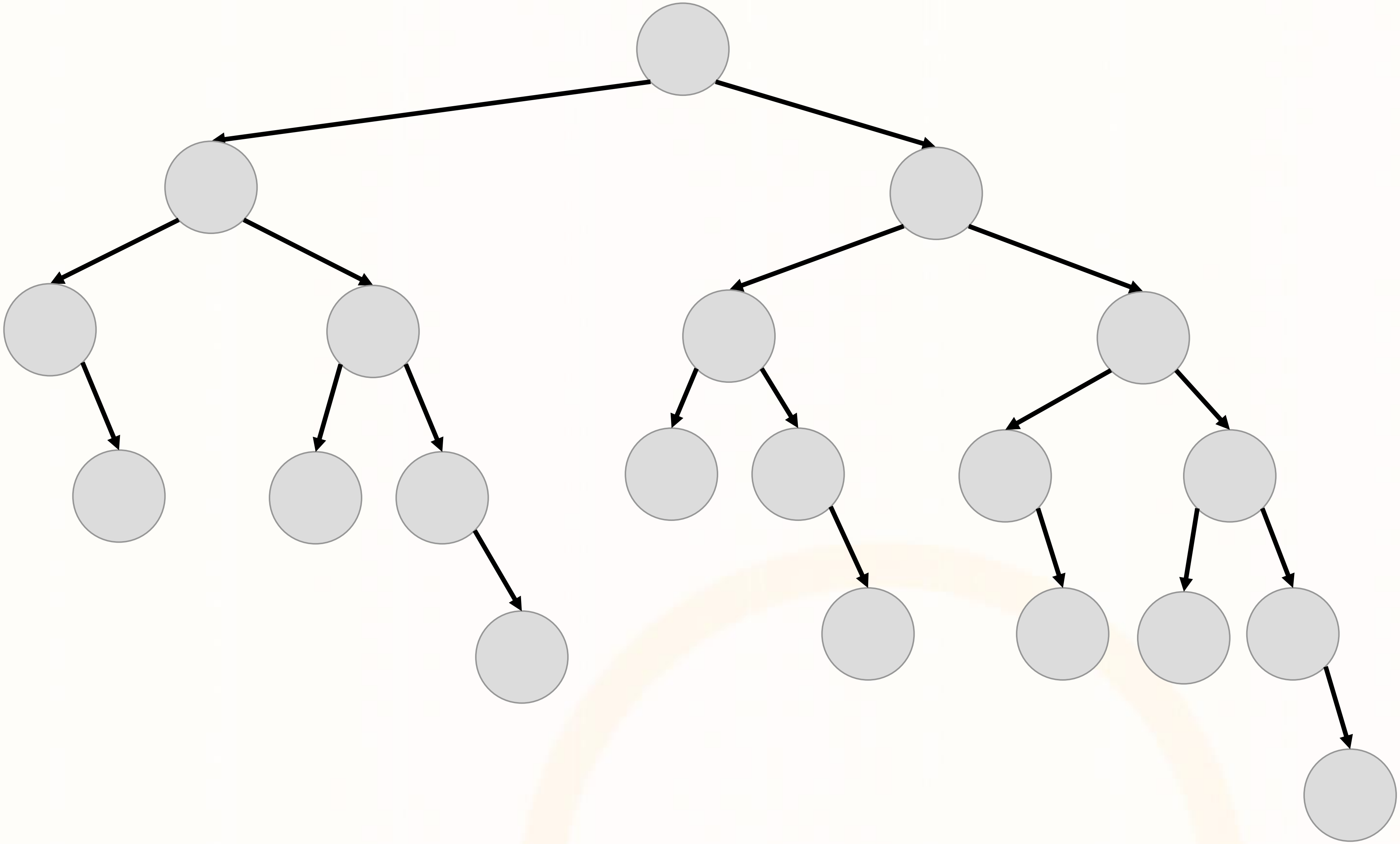


Το ύψος ενός AVL-δένδρου

- Έστω ότι N(h) είναι ο ελάχιστος αριθμός κόμβων ενός AVL-δένδρου ύψους h.
- Έχουμε: N(0) = 1, N(1) = 2.
- Για h ≥ 2, ένα AVL-δένδρο πρέπει:
  - να έχει μια ρίζα,
  - ένα από τα δύο του υπόδενδρα να έχει ύψος h-1,
  - τα ύψη των δύο υποδένδρων να διαφέρουν, το πολύ κατά 1.
- Άρα, N(h) = N(h-1) + N(h-2) + 1
- Η αναδρομική σχέση θυμίζει Fibonacci. Όπως αυτή, η N μεγαλώνει εκθετικά, δηλ. $N(h) \in \Theta\left(2^h\right)$. Επομένως:

Θεώρημα: Το ύψος ενός AVL-δένδρου με n κόμβους είναι της τάξης $\Theta(\log n)$

Ένα μικρότερο AVL-δένδρο ύψους 5

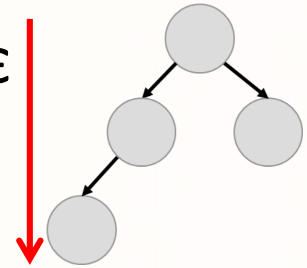


Υλοποίηση AVL δέντρων

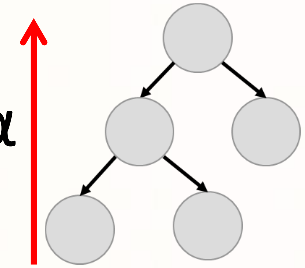
- Η παράσταση ενός κόμβου AVL-δένδρου είναι παρόμοια με αυτή ενός κόμβου ΔΔΑ, με την προσθήκη ενός επιπλέον πεδίου, που καταγράφει το ύψος του δένδρου που ριζώνει στον συγκεκριμένο κόμβο.
- Δηλαδή, ένας κόμβος μπορεί να υλοποιηθεί ως μια εγγραφή **AVLNode** με τέσσερα πεδία.
  - **key:** το κλειδί κόμβου
  - **height (int):** το ύψος του κόμβου
  - **left (pointer):** δείχνει το αριστερό υπόδενδρο που ριζώνει στον συγκεκριμένο κόμβο
  - **right (pointer):** δείχνει το δεξί υπόδενδρο που ριζώνει στον συγκεκριμένο κόμβο
- Το πεδίο **height** χρησιμοποιείται για τη διακρίβωση ανισοζυγίας (με σύγκριση των ανάλογων πεδίων των παιδιών κάθε κόμβου).

Εισαγωγή κόμβου

1. Η **εισαγωγή** κόμβου γίνεται όπως ακριβώς και σε ένα δυαδικό δένδρο αναζήτησης, με τη διαφορά ότι **καταγράφουμε τη διαδρομή που ακολουθείται** (από τη ρίζα προς τα φύλλα).



2. Στη συνέχεια, ακολουθούμε τη διαδρομή προς τα πίσω και δίνουμε στα πεδία height των κόμβων τις νέες τους τιμές.

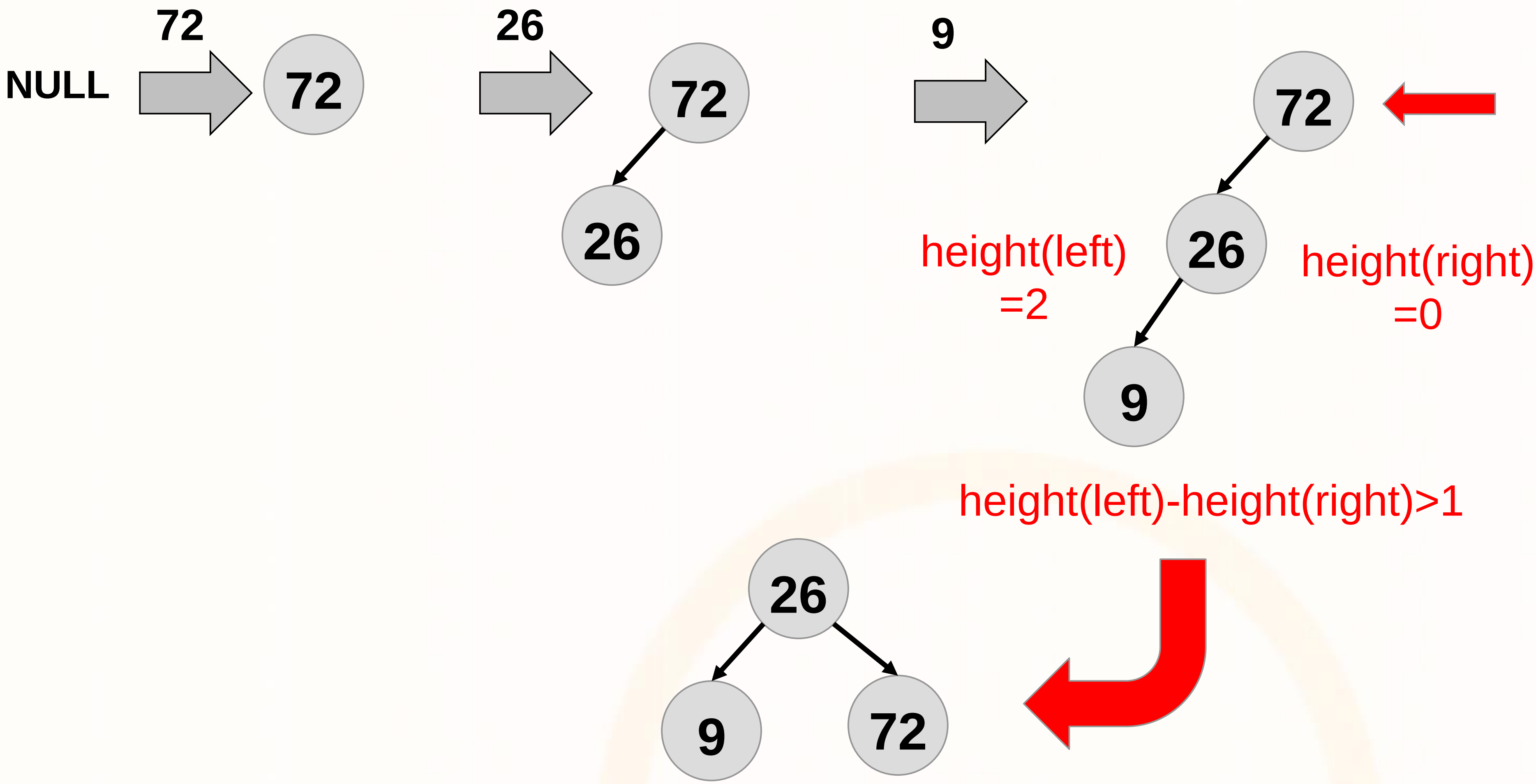


3. Αν αυτό προκαλέσει κάποια **ανισοζυγία**, δηλαδή, αν έχει σαν αποτέλεσμα κάποιος κόμβος να έχει παιδιά που το ύψος τους διαφέρει κατά τιμή >1, τότε ***αναπροσαρμόζουμε*** τα υποδένδρα ώστε το δένδρο να γίνει ξανά AVL.

Οι αναπροσαρμογές ονομάζονται ***περιστροφές*** (rotations)

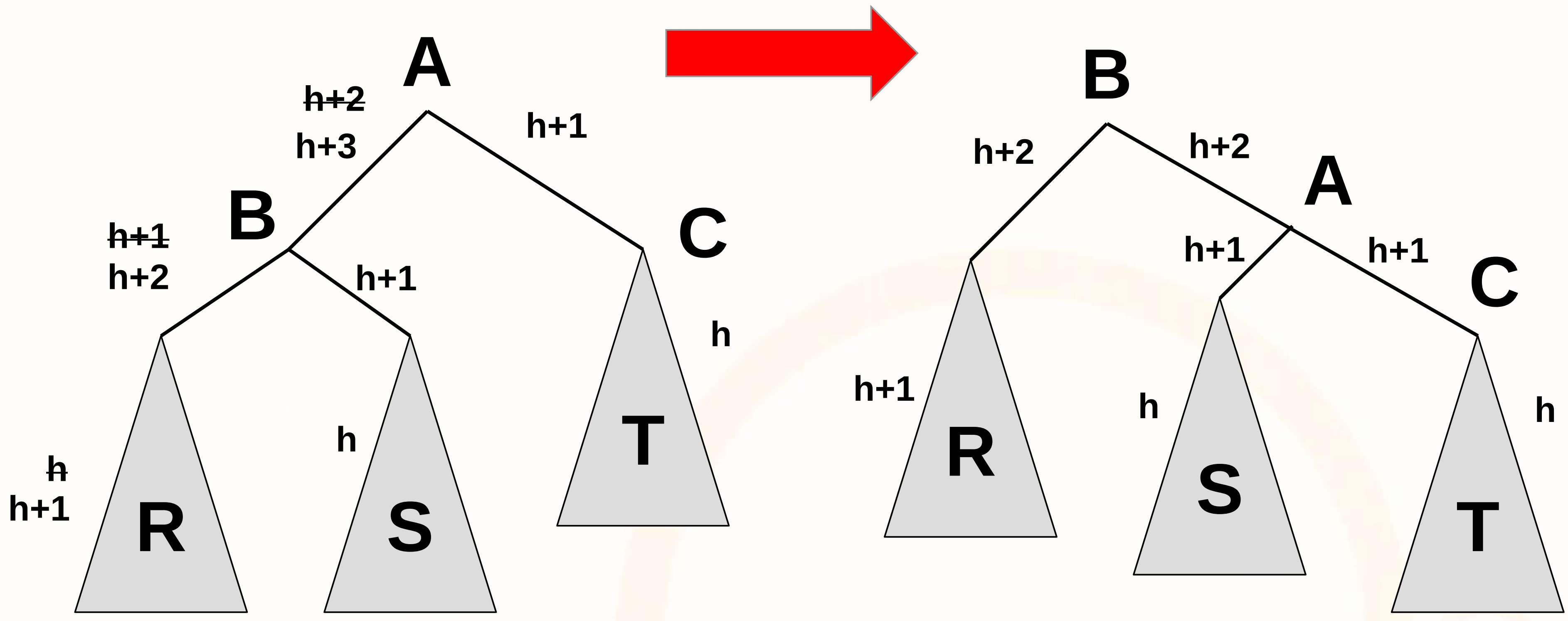
Παράδειγμα

- Εισαγωγή 72,26,9 στο κενό AVL δένδρο με αυτή τη σειρά:



Αριστερή Περιστροφή

- **Πριν την εισαγωγή:** τα δένδρα R,S,T έχουν το ίδιο ύψος, h.
- **Μετά την εισαγωγή:** έστω ότι ο κόμβος εισάγεται στο δένδρο R με αποτέλεσμα το ύψος του να γίνει h+1.
- Η αριστερή περιστροφή υλοποιεί το εξής:



Διαδικασία Α-περιστροφής

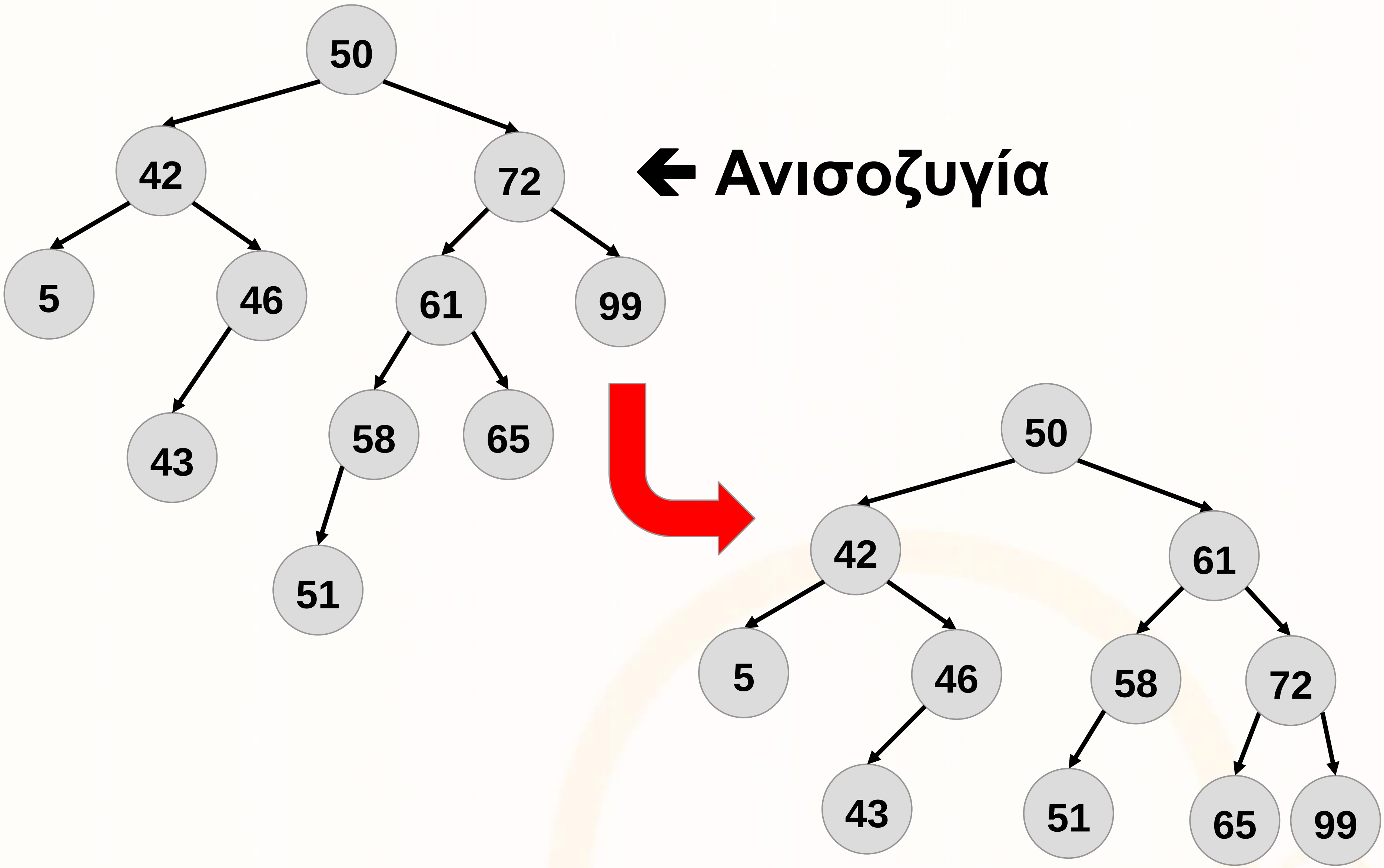
Αριστερή περιστροφή του (A,B) σημαίνει

1. A.left = B.right
2. B.right = A
3. A.height = C.height + 1
4. B.height = C.height + 2

- Πριν την περιστροφή ο A ήταν ο πατέρας του B, και μετά, ο B είναι ο πατέρας του A.
- Το δένδρο παραμένει δυαδικό δένδρο αναζήτησης.
- Μετά την περιστροφή το δένδρο είναι AVL:

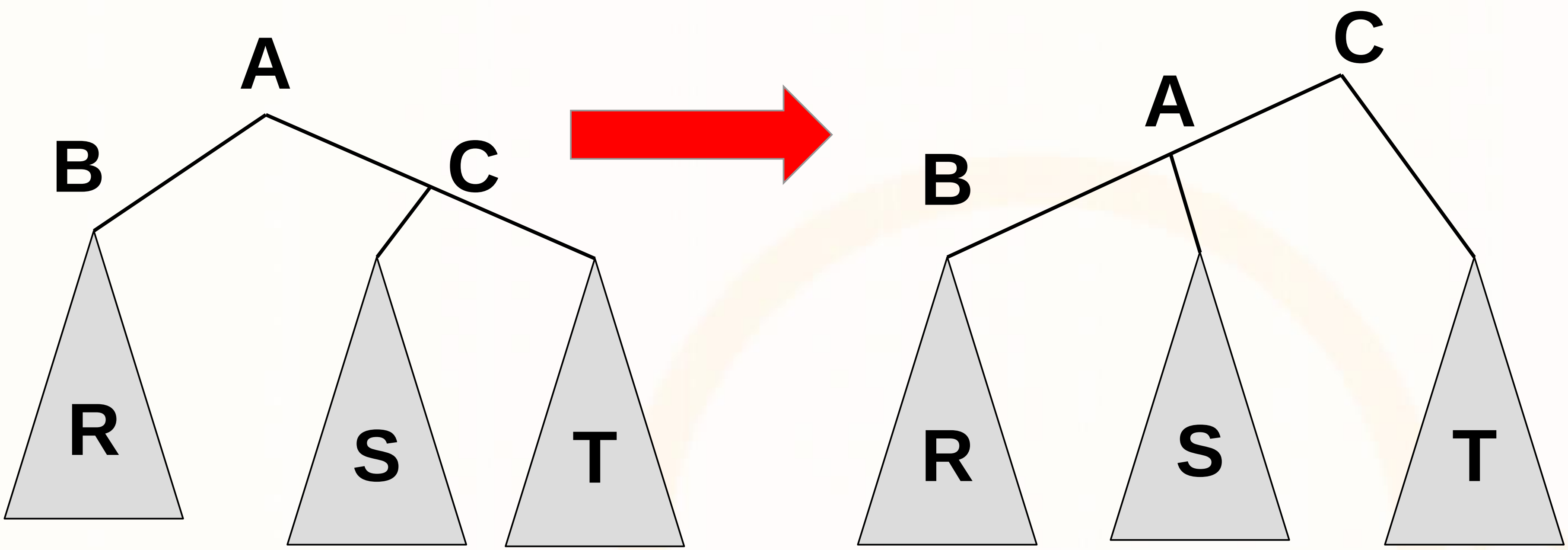
  A.height = h + 1 = ύψος του R.

Παράδειγμα Α-περιστροφής



Δεξιά Περιστροφή

- **Συμμετρική προς την αριστερή περιστροφή**
- **Πριν την εισαγωγή:** τα δένδρα R,S,T έχουν το ίδιο ύψος, h.
- **Μετά την εισαγωγή:** έστω ότι ο κόμβος εισάγεται στο δένδρο T με αποτέλεσμα το ύψος του να γίνει h+1.
- Η δεξιά περιστροφή υλοποιεί το εξής:



Διαδικασία Δ-περιστροφής

Δεξιά περιστροφή του (A,B) σημαίνει

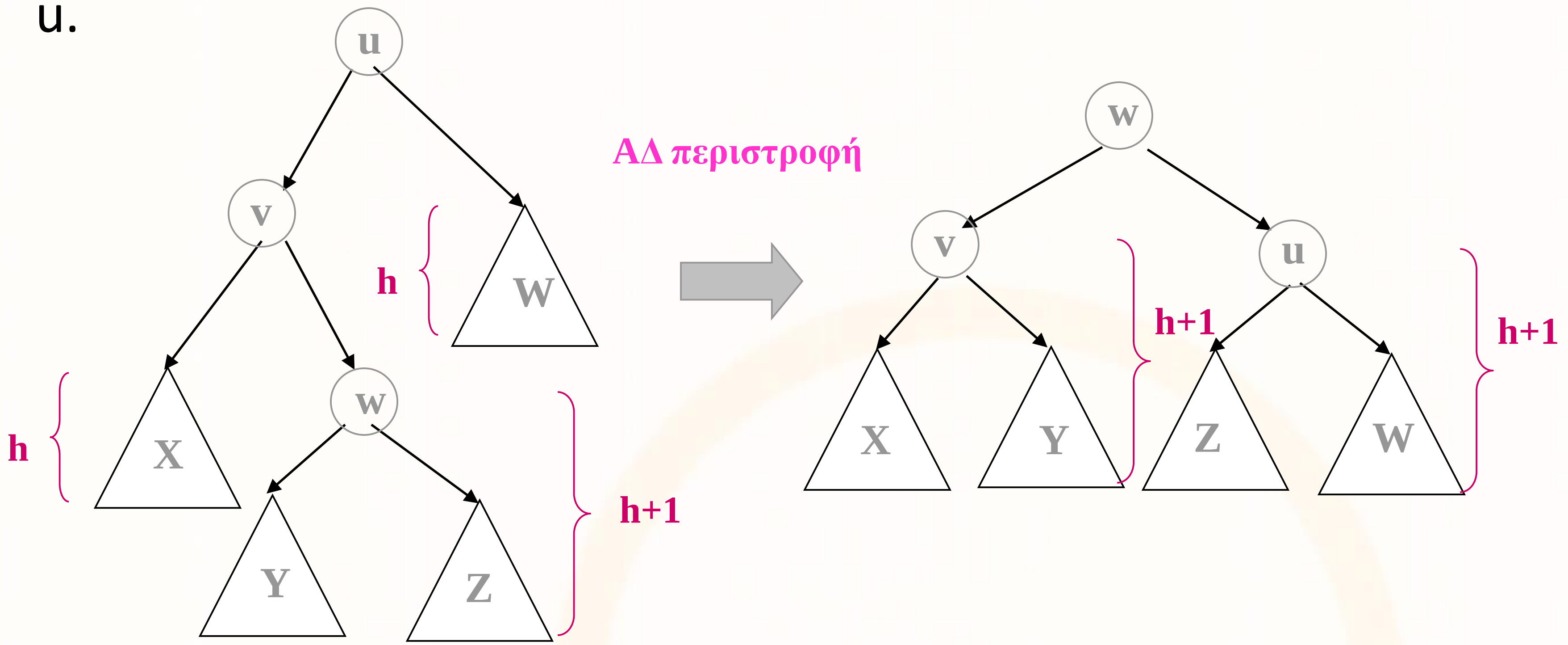
1. A.right= C.left
2. C.left = A
3. A.height = B.height + 1
4. C.height = B.height + 2

- Πριν την περιστροφή ο A ήταν ο πατέρας του C, και μετά, ο C είναι ο πατέρας του A.
- Το δένδρο παραμένει δυαδικό δένδρο αναζήτησης.
- Μετά την περιστροφή το δένδρο είναι AVL:

  C.height = h + 1 = ύψος του T.

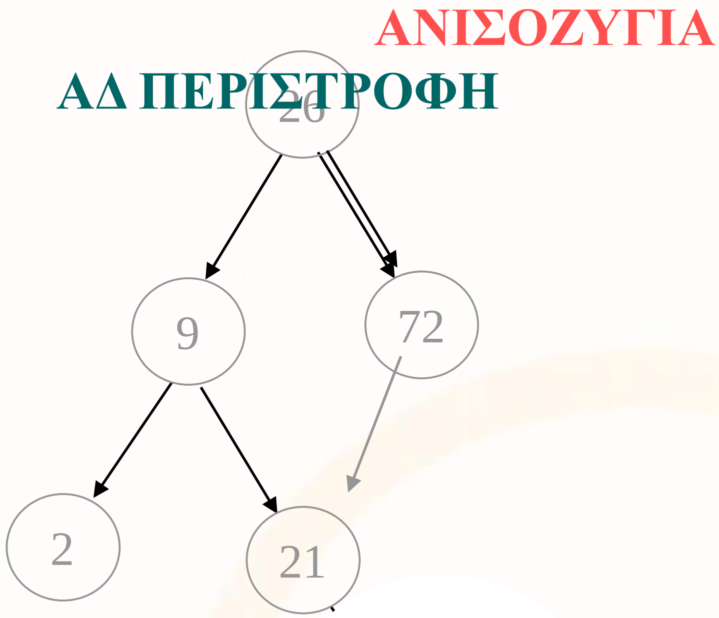
ΑΔ-Περιστροφή

- Τα δένδρα X και W έχουν ύψος h. Μετά από κάποια εισαγωγή, το w έχει ύψος h+1, προκαλώντας ανισοζυγία στο u.



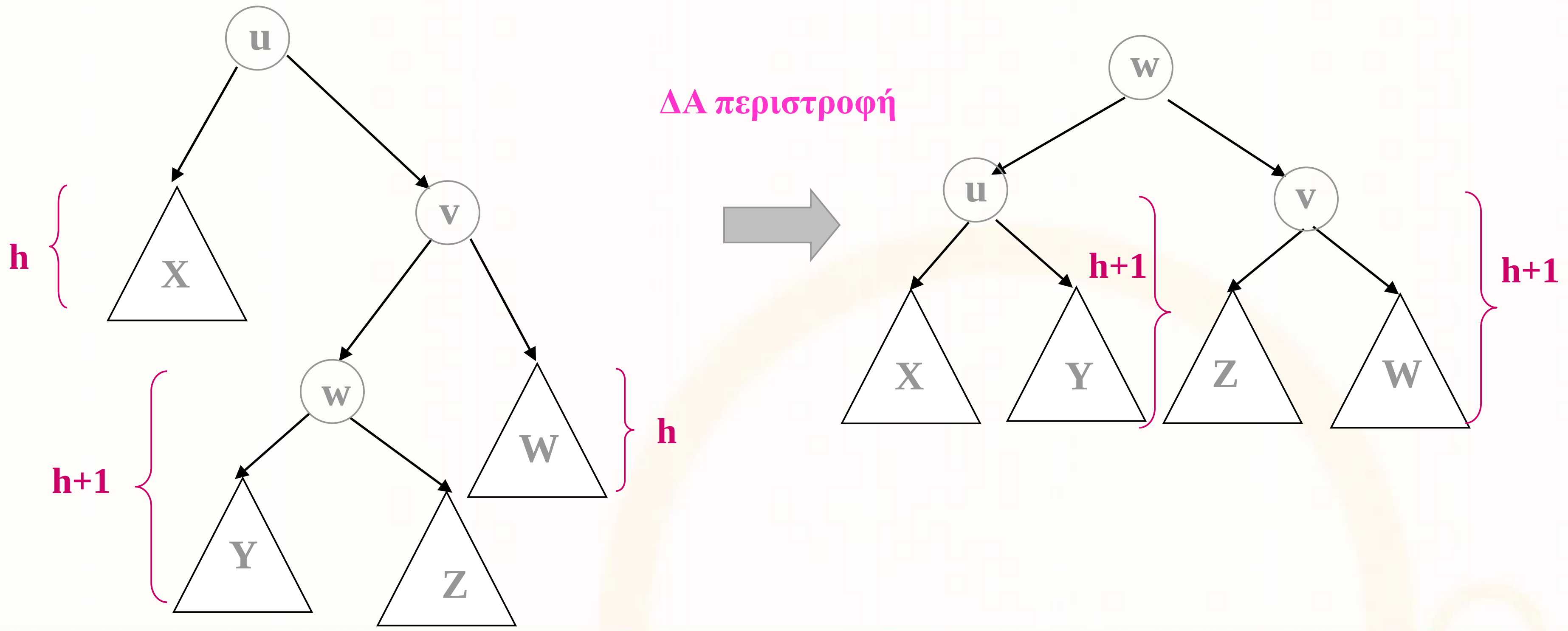
Παράδειγμα ΑΔ-περιστροφής

- Με την εισαγωγή των στοιχείων 72, 26, 9, 2, 21, 25 σε ένα AVL-δένδρο, δημιουργείται ανισοζυγία στον κόμβο 26.
- Με εφαρμογή ΑΔ περιστροφής έχουμε:

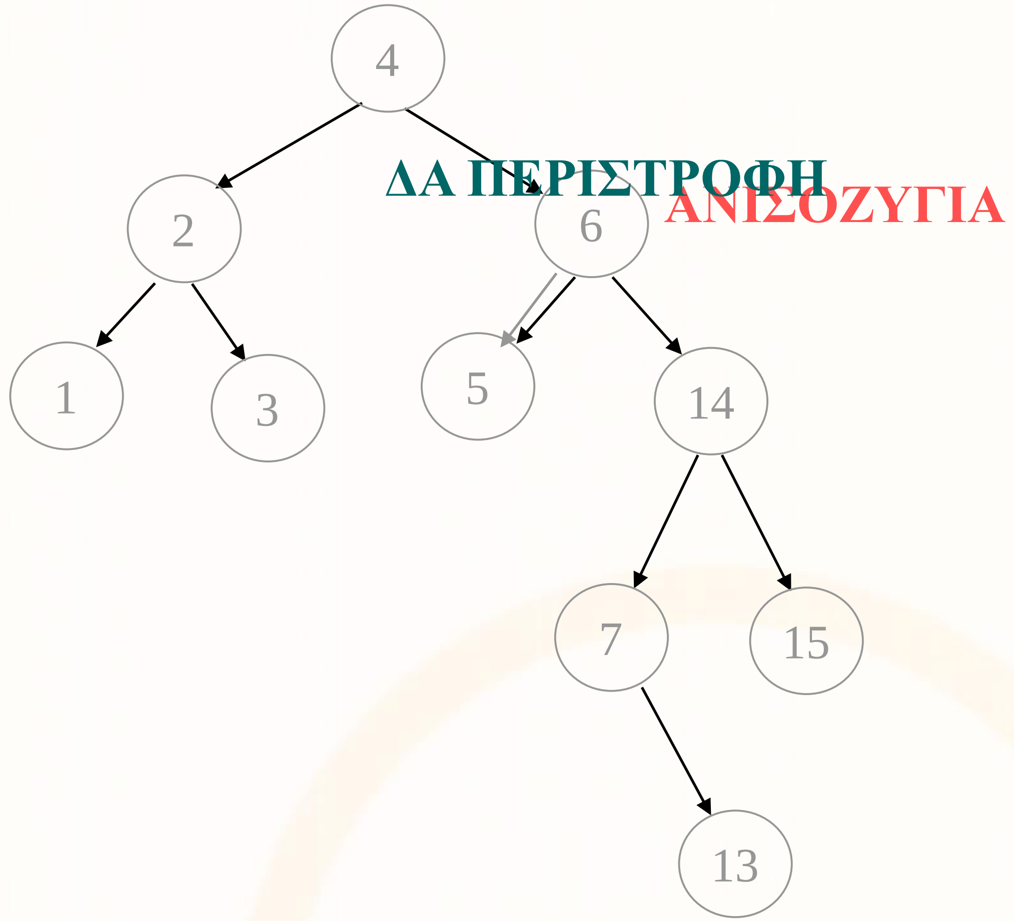


ΔΑ-Περιστροφή

- Τα δένδρα X και W έχουν ύψος h. Μετά από κάποια εισαγωγή, το w έχει ύψος h+1, προκαλώντας ανισοζυγία στο u.



Παράδειγμα ΔΑ περιστροφής



Διαδικασίες ΑΔ και ΔΑ-περιστροφής

ΑΔ περιστροφή του (u,v,w) υλοποιείται ως εξής:

1. v.right = w.left,
2. u.left = w.right,
3. w.left = v,
4. w.right = u,
5. v.height, u.height, w.height = …

ΔΑ περιστροφή του (u,v,w) υλοποιείται ως εξής:

1. v.left = w.right,
2. u.right = w.left,
3. w.left = u,
4. w.right = v, και
5. v.height, u.height, w.height = ….

- Οι περιστροφές δεν παραβιάζουν τη ΔΔΑ συνθήκη.
- Το δένδρο που δημιουργείται είναι AVL-δένδρο (οι κόμβοι v και u έχουν ύψος h+1).

Εφαρμογή περιστροφών

Όπως έχουμε περιγράψει η διαδικασία εισαγωγής κόμβου σε AVL-δένδρο γίνεται ως εξής:

1. Εισάγουμε το στοιχείο στο κατάλληλο φύλλο όπως ακριβώς σε ένα δυαδικό δένδρο αναζήτησης. Καταγράφουμε τη διαδρομή που ακολουθήσαμε, δηλαδή αν r είναι η ρίζα και u είναι το φύλλο που προσθέσαμε τότε παίρνουμε διαδρομή με μορφή:

$$u = v_1, v_1, \ldots, v_k = r$$

2. Ακολουθούμε τη διαδρομή προς τα πίσω και δίνουμε στα πεδία height των κόμβων τις νέες τους τιμές.
3. Αν σε κάποιο σημείο αυτό προκαλέσει ανισοζυγία και μόλις συμβεί αυτό, (δηλ. αν έχει σαν αποτέλεσμα κάποιοι κόμβοι να έχουν παιδιά που το ύψος τους διαφέρει κατά τιμή >1), τότε εφαρμόζουμε στον κόμβο αυτό, έστω vi , την κατάλληλη περιστροφή. Επιλέγουμε την περιστροφή ως εξής:

Εφαρμογή περιστροφών

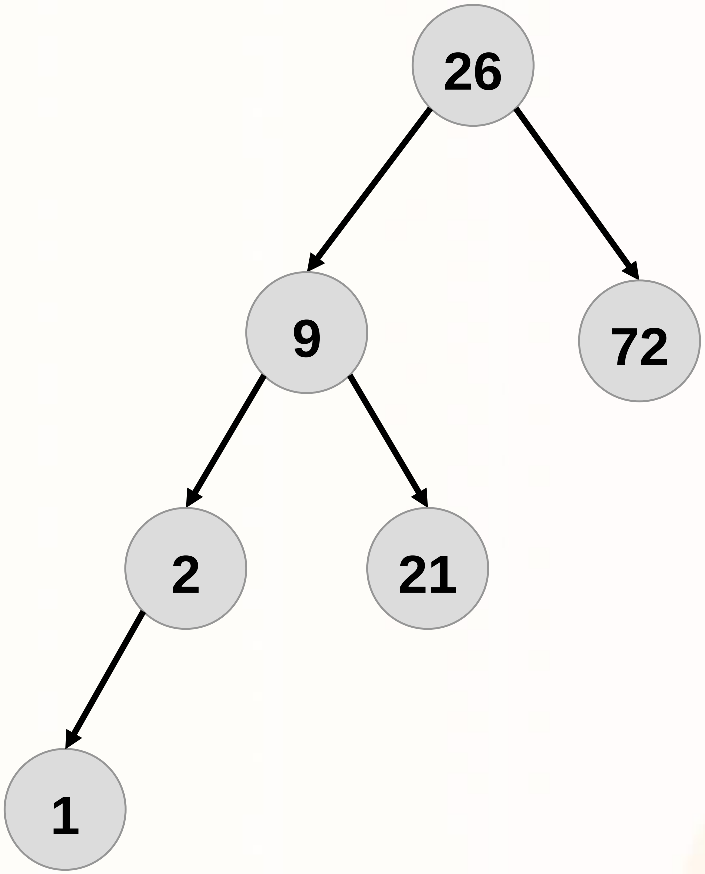
i. αν ο v_{i-1} είναι αριστερό παιδί του v_i και ο v_{i-2} αριστερό παιδί του v_{i-1} τότε εφαρμόζουμε την Α-περιστροφή,
ii. αν ο v_{i-1} είναι δεξιό παιδί του v_i και ο v_{i-2} δεξιό παιδί του v_{i-1} τότε εφαρμόζουμε τη Δ-περιστροφή,
iii. αν ο v_{i-1} είναι αριστερό παιδί του v_i και ο v_{i-2} δεξιό παιδί του v_{i-1} τότε εφαρμόζουμε την ΑΔ-περιστροφή,
iv. αν ο v_{i-1} είναι δεξιό παιδί του v_i και ο v_{i-2} αριστερό παιδί του v_{i-1} τότε εφαρμόζουμε τη ΔΑ-περιστροφή.

4. Ενημερώνουμε τον πατέρα του vi για το ποιο είναι το παιδί του ως
   - αποτέλεσμα της περιστροφής, ή, αν ο vi είναι η ρίζα του δένδρου,
   - ενημερώνουμε την καλούσα συνάρτηση για τη νέα ρίζα.

ΠΡΟΣΟΧΗ:
- Η περιστροφή εφαρμόζεται στον πιο χαμηλό κόμβο που παρουσιάζει ανισοζυγία.
- Ένα AVL-δένδρο είναι ΔΔΑ!

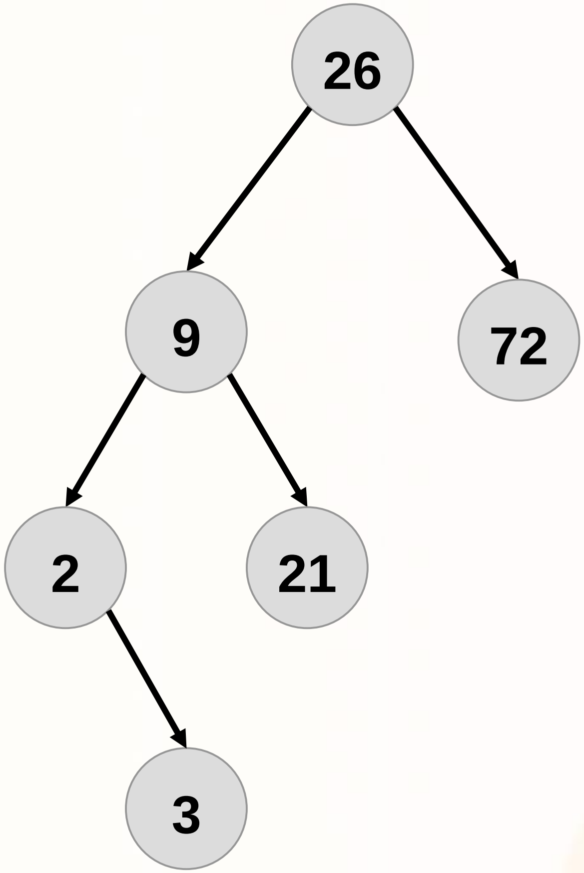
AVL Δέντρα: Ασκήσεις

Τι περιστροφή πρέπει να γίνει όταν εισαχθεί το 1;



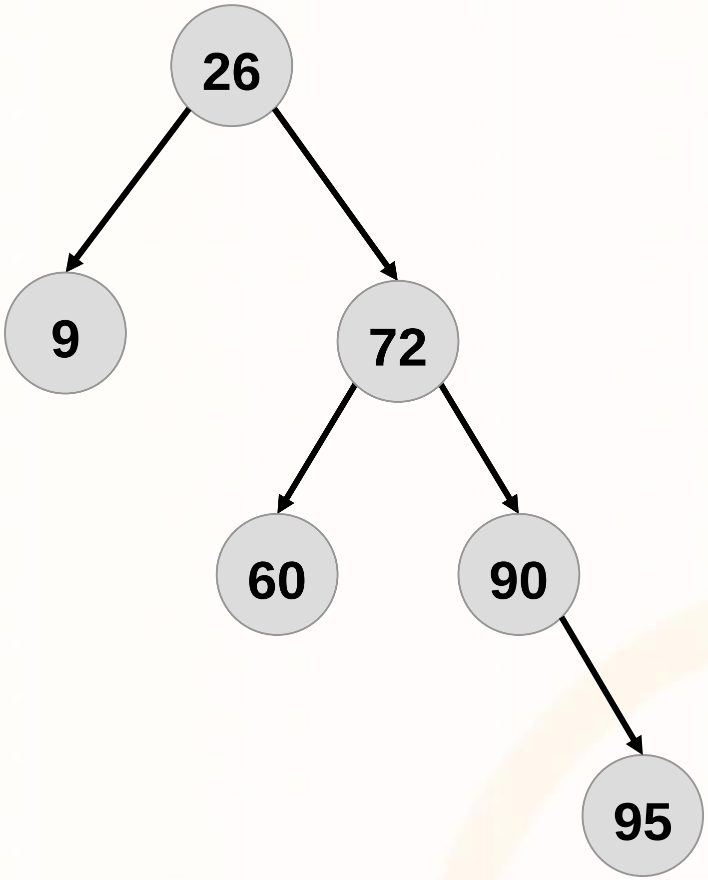
AVL Δέντρα: Ασκήσεις

Τι περιστροφή πρέπει να γίνει όταν εισαχθεί το 3;



AVL Δέντρα: Ασκήσεις

Τι περιστροφή πρέπει να γίνει όταν εισαχθεί το 95;



AVL Δέντρα: Ασκήσεις

Τι περιστροφή πρέπει να γίνει όταν εισαχθεί το 71;