Διάλεξη 14: Δέντρα IV - B-Δένδρα

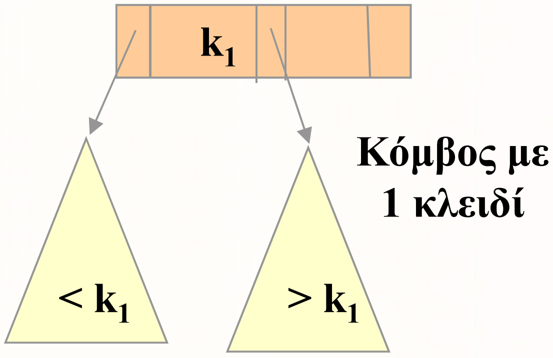
Στην ενότητα αυτή θα μελετηθούν τα εξής επιμέρους θέματα:

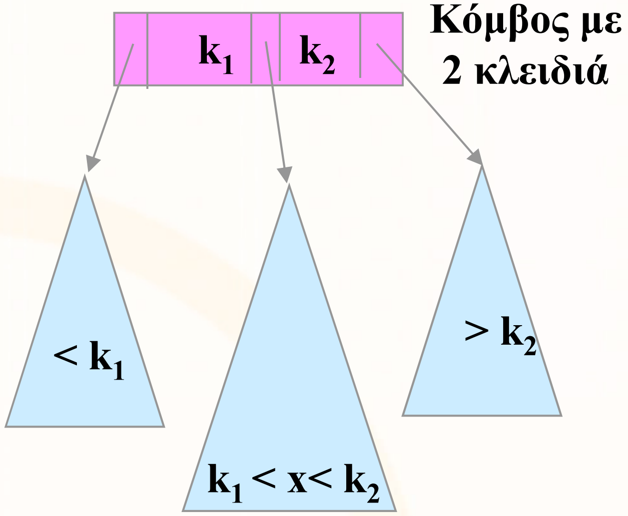
- 2-3 Δένδρα, Εισαγωγή και άλλες πράξεις
- Άλλα Δέντρα: B-δένδρα, B+-δέντρα, R-δέντρα

2-3 (2 ή 3 παιδιά) Δένδρα

- Γενίκευση των δυαδικών δένδρων αναζήτησης.

Ορισμός

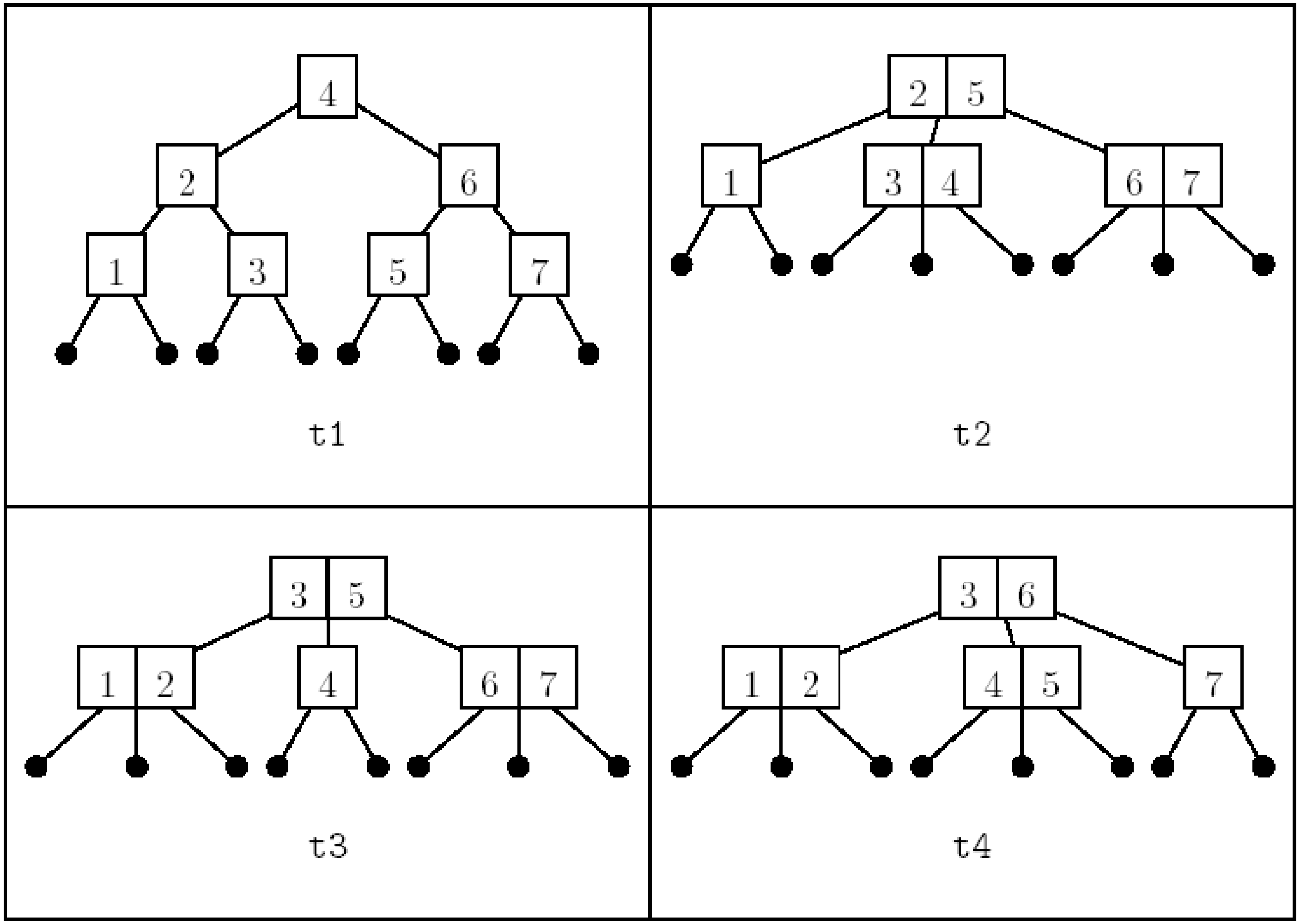




  - Κάθε κόμβος περιέχει **ένα ή δύο** κλειδιά.
  - Ένας εσωτερικός κόμβος u με <u>ένα κλειδί</u> , k_1 έχει δύο παιδιά (υπόδενδρα): το **αριστερό**, u.left, το οποίο περιέχει κλειδιά $< k_1$, και το **μεσαίο**, u.center, το οποίο περιέχει κλειδιά $> k_1$.
  - Ένας εσωτερικός κόμβος u με <u>δύο κλειδιά</u>, $k_1 < k_2$, έχει τρία παιδιά, το **αριστερό**, u.left, το **μεσαίο**, u.center, και το **δεξί**, u.right. Όλα τα κλειδιά του υποδένδρου u.left είναι $< k_1$, όλα τα κλειδιά του u.center είναι $k_1 <$ και $< k_2$ και όλα τα κλειδιά του u.right είναι $> k_2$.
  - <u>**Όλα τα φύλλα βρίσκονται στο ίδιο επίπεδο.**</u>

Σημείωση: Θεωρήστε ότι σε περίπτωση ισότητας με το κλειδί, η εισαγωγή θα γίνεται αριστερά (αντί στα δεξιά όπως ίσχυε με το ΔΔΑ)

Παραδείγματα 2-3 Δένδρων



- Όλα τα 2-3 δένδρα είναι **ισοζυγισμένα**. Δηλαδή όλα τα φύλλα έχουν την ίδια απόσταση από την ρίζα

Υλοποίηση 2-3 Δένδρων

- Ένας κόμβος 2-3 Δένδρου μπορεί να παρασταθεί ως μια εγγραφή με 6 πεδία:
  1. **`numkeys`**, τύπου int, που δηλώνει τον αριθμό των κλειδιών που περιέχει ο κόμβος,
  2. **`key1`**, όπου αποθηκεύεται το **πρώτο κλειδί**,
  3. **`key2`**, όπου αποθηκεύεται το **δεύτερο κλειδί**, αν υπάρχει,
  4. **`left`**, τύπου δείκτη, που δείχνει στο **αριστερό παιδί** του κόμβου,
  5. **`center`**, τύπου δείκτη, που δείχνει στο **μεσαίο παιδί** του κόμβου,
  6. **`right`**, τύπου δείκτη, που δείχνει στο **δεξί παιδί** του κόμβου αν υπάρχει.

- Ένα δένδρο αναπαρίσταται ως ένας δείκτης σε κόμβο 2-3 δένδρου **(που δείχνει στη ρίζα**) και επιτρέπει τις πράξεις εισαγωγής, διαγραφή και αναζήτησης.

  NODE *root;

Ιδιότητες 2-3 δένδρου

- Αν N(h) είναι ο μικρότερος αριθμός κλειδιών ενός 2-3 δένδρου ύψους h, τότε

  $$N(0) = 1, \quad N(h) = 1 + 2 \cdot N(h-1)$$

- Αν M(h) είναι ο μεγαλύτερος αριθμός κλειδιών ενός 2-3 δένδρου ύψους h, τότε

  $$M(0) = 2, \quad M(h) = 2 + 3 \cdot M(h-1)$$

- Ο αριθμός κλειδιών ενός 2-3 δένδρου ύψους h είναι το πολύ $3^{h+1} - 1$ και το λιγότερο $2^{h+1} - 1$.
- *Επομένως, το ύψος ενός 2-3 δένδρου με n κόμβους είναι O(log n).*
- Η διαδικασία εύρεσης στοιχείου σε ένα 2-3 δένδρο είναι εύκολη (παρόμοια με αυτή ενός ΔΔΑ).

  Ο χρόνος εκτέλεσης της είναι τάξης $O(\log_2 n)$.

Εισαγωγή κόμβου σε ένα 2-3 δένδρο

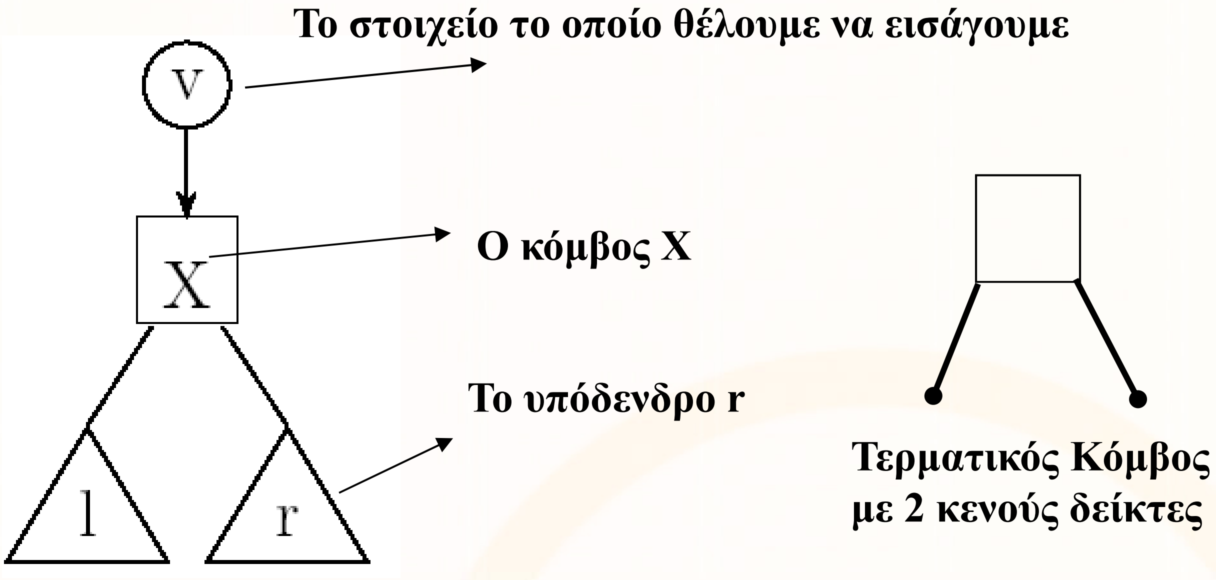
Η εισαγωγή κάποιου κλειδιού k σε ένα 2-3 Δένδρο μπορεί να χωριστεί στις ακόλουθες τρεις **<u>λογικές</u>** φάσεις

1. **Καθοδική Φάση (Downward Phase)**

   Σε αυτή την φάση διανύουμε αναδρομικά το δένδρο μέχρι να φθάσουμε σε τερματικό κόμβο (δηλαδή κάποιο φύλλο). Δηλαδή:
   - αν k<u.key1 τότε προχώρα στον κόμβο u.left
   - αν k>u.key2 τότε προχώρα στον κόμβο u.right
   - αν u.key1 < k < u.key2 τότε προχώρα στον κόμβο u.center.
2. **Τερματική Φάση (Terminal Phase)**
   - Όταν φτάσουμε σε φύλλο τότε προσπαθούμε να κάνουμε την εισαγωγή
   - Αν δεν έχει αρκετό χώρο τότε διασπάται το φύλλο και «ανασηκώνουμε» (kick up) αναδρομικά το μεσαίο στοιχείο στον πατέρα.
3. **Ανοδική Φάση (Upward Phase)**
   - Σε αυτή την φάση κάνουμε την εισαγωγή αναδρομικά στον γονέα αν δεν πετύχαμε εισαγωγή στην τερματική φάση.

Περιγραφή Κάποιων Συμβολισμών

- Στις επόμενες διαφάνειες θα χρησιμοποιήσουμε τους **ακόλουθους συμβολισμούς**



Το στοιχείο το οποίο θέλουμε να εισάγουμε

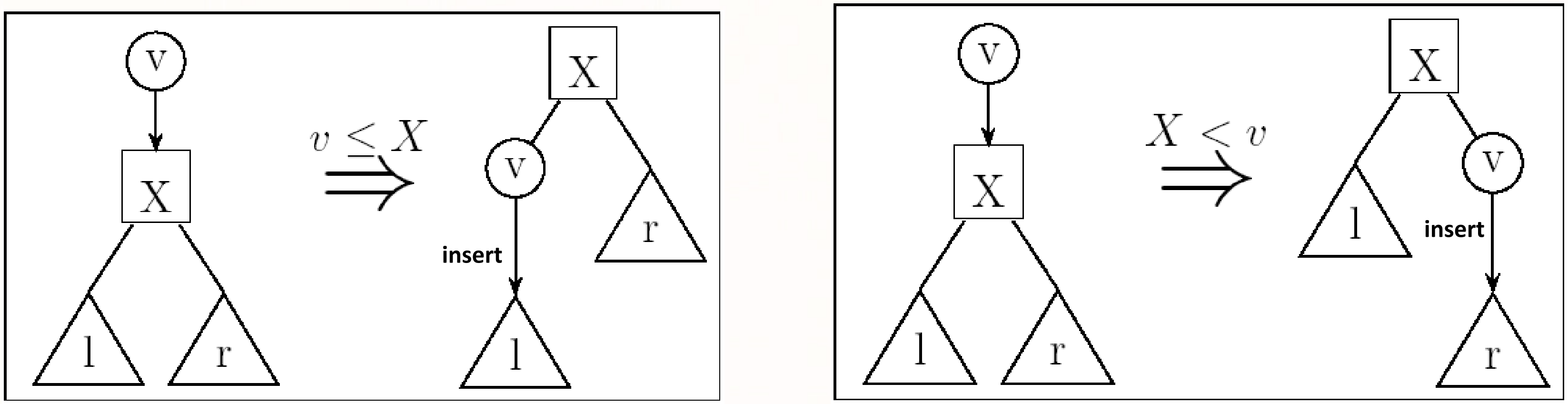
Ο κόμβος X

Το υπόδενδρο r

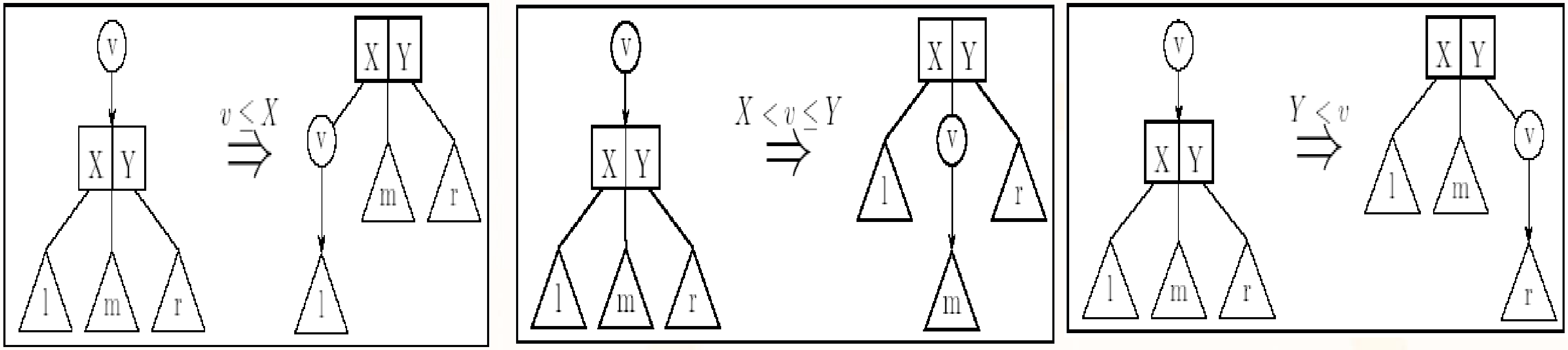
Τερματικός Κόμβος
με 2 κενούς δείκτες

1) Καθοδική Φάση (Downward Phase)

Εισαγωγή του στοιχείου v – Η ρίζα περιέχει 1 στοιχείο



Εισαγωγή του στοιχείου v – η ρίζα περιέχει 2 στοιχεία



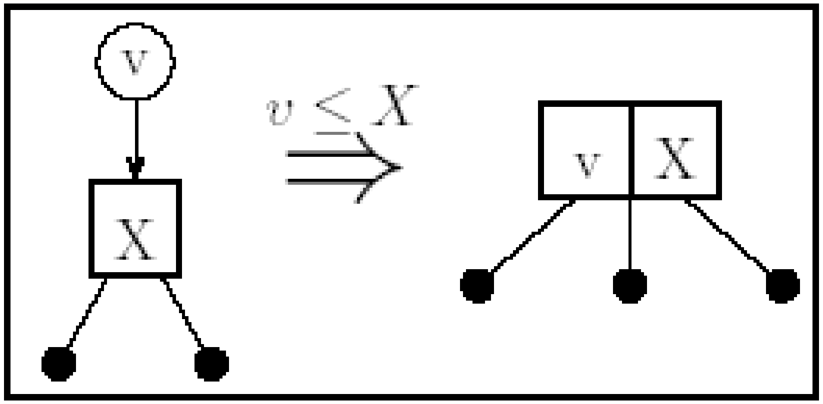
Προσοχή: **Σε αυτή την φάση απλά ζητάμε σε ένα από τα υποδένδρα να χειριστεί την εισαγωγή αλλά δεν κάνουμε την εισαγωγή**

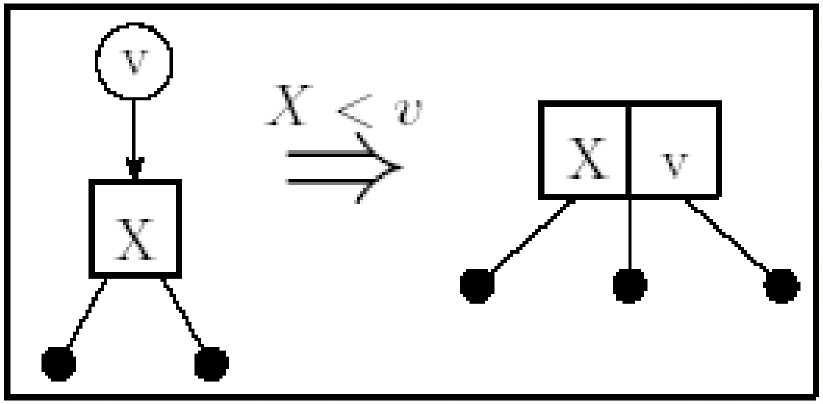
Καθοδική Φάση (συν.)

- Αν u.numkeys = 1, τότε
  - αν k<u.key1 τότε προχώρα στον κόμβο u.left
  - αν k>u.key1 τότε προχώρα στον κόμβο u.center
  - αν k=u.key1 τότε σταμάτα.
- Αν u.numkeys = 2, τότε
  - αν k<u.key1 τότε προχώρα στον κόμβο u.left
  - αν k>u.key2 τότε προχώρα στον κόμβο u.right
  - αν u.key1 < k < u.key2 τότε προχώρα στον κόμβο u.center.
  - διαφορετικά, σταμάτα.

2) Τερματική Φάση (Terminal Phase)

Α. Όταν φτάσουμε σε τερματικό κόμβο <u>και υπάρχει χώρος</u> τότε μπορούμε να κάνουμε την εισαγωγή κατευθείαν. Δηλαδή:

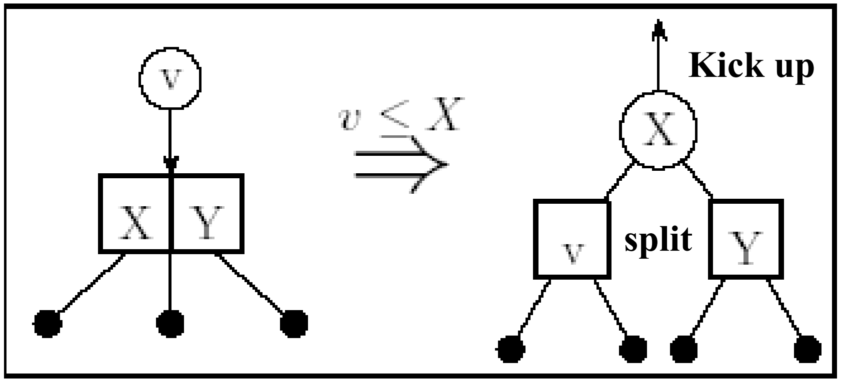


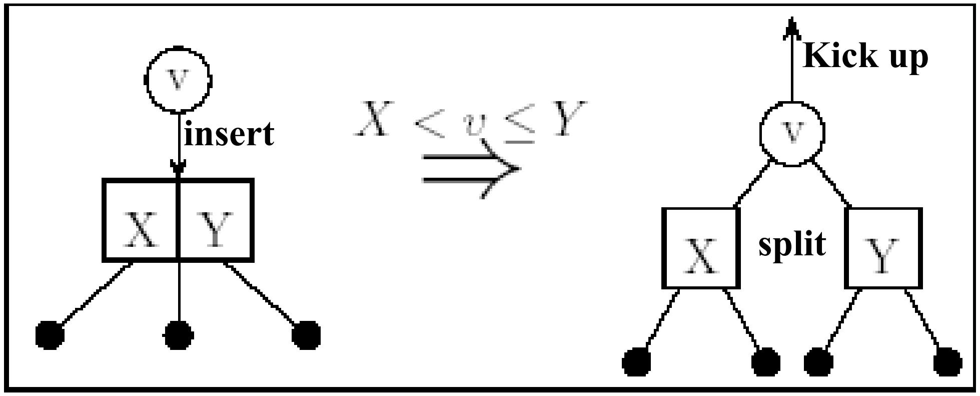


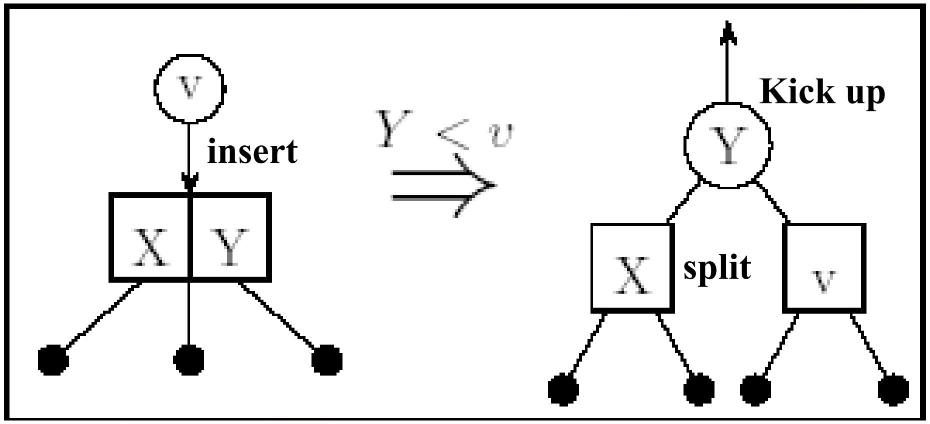
Αν **ΔΕΝ υπάρχει χώρος** τότε **διασπάται (split)** ο τερματικός κόμβος και **προάγεται (kick-up)** το **μεσαίο στοιχείο** στον **πατέρα** (δες επόμενη διαφάνεια)

2) Τερματική Φάση (Terminal Phase) (συν.)

Αναλυτικά οι υπό-φάσεις της **Διάσπασης (Split)** και **Προαγωγής (Kick-up)**





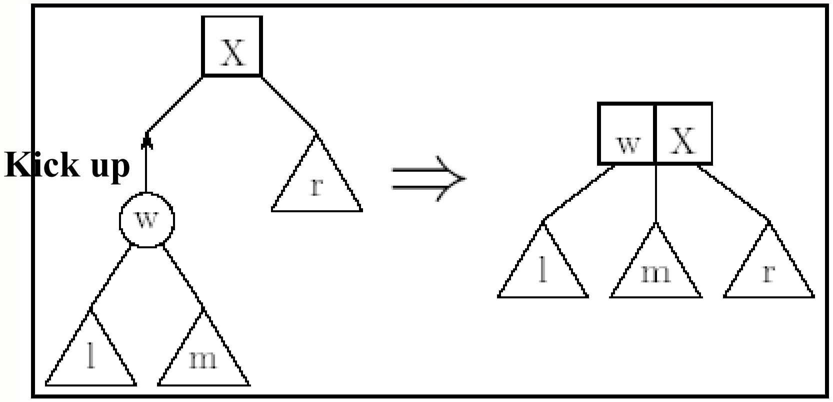


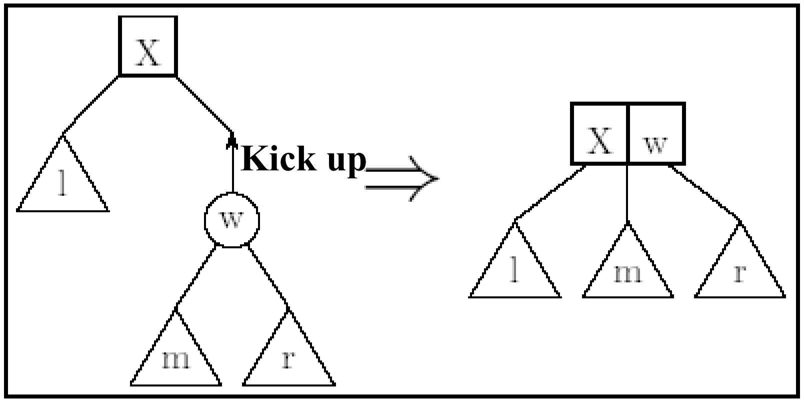
Τερματική Φάση (συν.)

- Έστω ότι προσθέτουμε το κλειδί k στο φύλλο u.
- *Αν u.numkeys = 1*, k_1= u.key1, τότε
  - `u.key1        = min (k₁, k)`,
  - `u.key2        = max (k₁, k)`,
  - `u.numkeys  = 2`
- *Αν u.numkeys = 2*, k_1 = u.key1, k_2 =u.key2, τότε
  - `u.key1        = min (k₁, k₂, k)`,
  - `u.numkeys  = 1,`
  - `p.numkeys  = 1,`
  - `p.key1        = max(k₁, k₂ ,k).`
  - προχώρα στην ανοδική φάση με παραμέτρους p, mid(k_1, k_2 ,k)

3: Ανοδική Φάση (Upward Phase)

- Όταν φτάσουμε στον πατέρα και **υπάρχει** χώρος τότε μπορούμε να κάνουμε την εισαγωγή κατευθείαν. Δηλαδή:



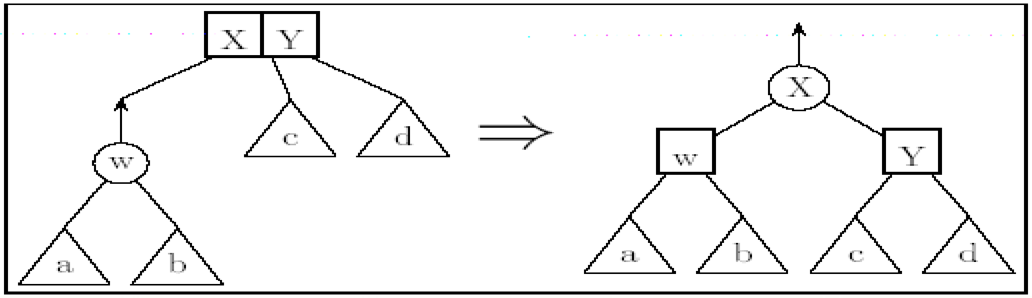


- Αν **ΔΕΝ υπάρχει χώρος** τότε πρέπει να διασπαστεί ο πατέρας και να **προαχθεί (kick-up)** το μεσαίο στοιχείο στον πατέρα του πατέρα (δες επόμενη διαφάνεια)

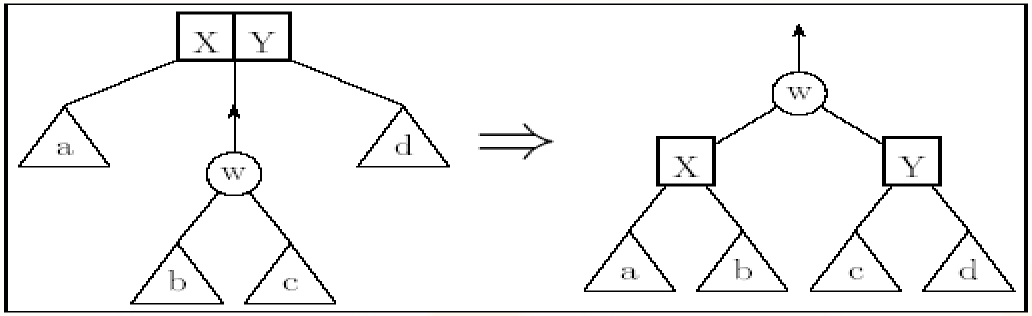
3: Ανοδική Φάση (Upward Phase) συνέχεια

- Όταν φτάσουμε στον πατέρα και <u>**δεν υπάρχει**</u> χώρος τότε αναδρομικά διασπάμε τον πατέρα μέχρι να εισαχθεί το στοιχείο. Αν διασπαστεί η ρίζα τότε αυξάνεται και το ύψος του δένδρου κατά 1! :

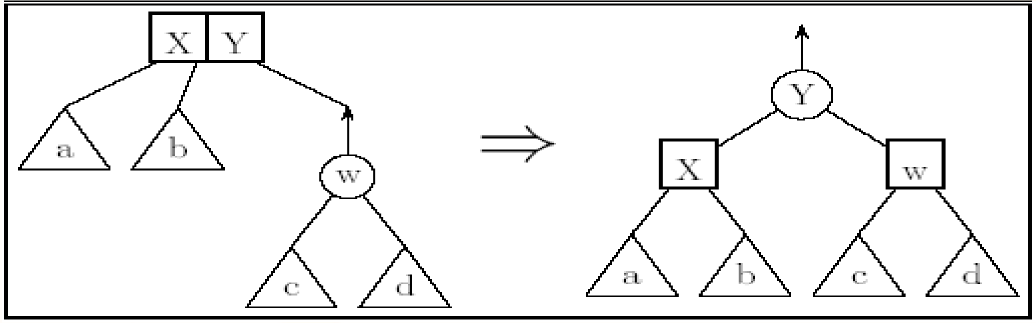
w <= X < Y



X < w <= Y



Y < w



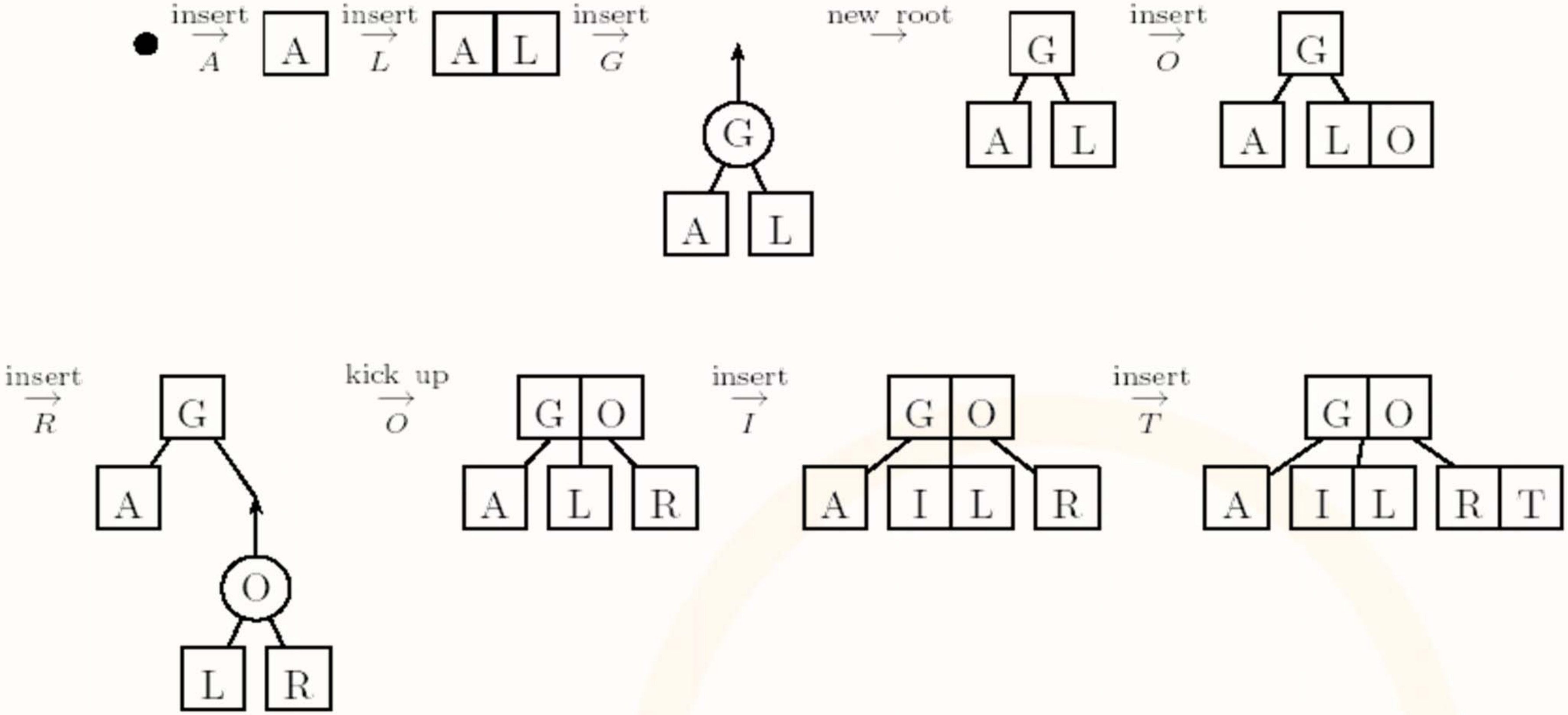
Ανοδική Φάση (συν.)

- Έστω ότι προσθέτουμε το κλειδί k και την αναφορά p στον κόμβο u.
- *<u>Αν u.numkeys = 1</u>*, k_1= u.key1, τότε
  - αν k > k_1 , u.key2 = k, u.right = p,
  - αν k < k_1 , u.key1 = k, u.key2 = k_1, u.right = u.center, u.center = p.
  - u.nunkeys = 2.
- <u>Αν u.numkeys = 2</u>, k_1= u.key1, k_2= u.key2, u.left = p_1, u.center = p_2 , u.right = p_3, τότε υπάρχουν 3 περιπτώσεις που εξαρτώνται από τη σχέση των κλειδιών k, k_1, k_2. Ας ασχοληθούμε με την περίπτωση k < k_1 (οι άλλες δύο είναι παρόμοιες). Τότε
  - u.key1 = k, u.center = p, u.numkeys = 1,
  - q.numkeys = 1, q.key1 = k_2, q.left = p2, q.center = p3
  - αν ο u είναι η ρίζα τότε προχώρα στη φάση 3, διαφορετικά επανάλαβε τη φάση 2(β) με παραμέτρους (q, k_1) στον πατέρα του u.

Ολοκληρωμένο Παράδειγμα Εισαγωγής

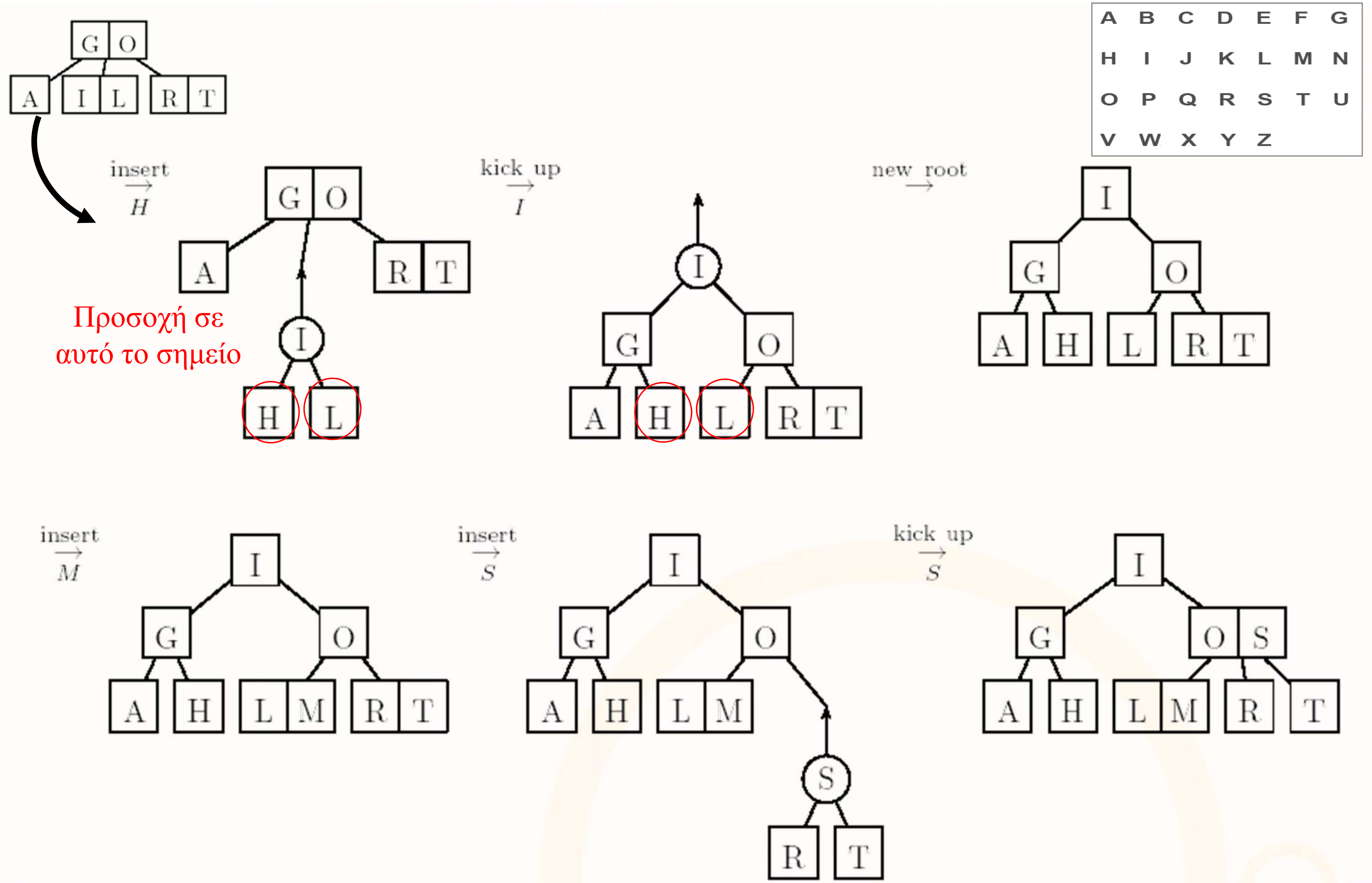
- Παράδειγμα εισαγωγής της λέξης “A L G O R I T H M S” (γράμμα-γράμμα) σε ένα κενό 2-3 δένδρο.

A	B	C	D	E	F	G
H	I	J	K	L	M	N
O	P	Q	R	S	T	U
V	W	X	Y	Z		



συνέχεια στην επόμενη διαφάνεια…

Ολοκληρωμένο Παράδειγμα Εισαγωγής (συν.)



Προσοχή σε αυτό το σημείο

Διαδικασία Εισαγωγής Κόμβου: Χρόνος Εκτέλεσης

- Η αναδρομική διαδικασία εισαγωγής κόμβου, παίρνει παραμέτρους:
  1. την αναφορά στον κόμβο όπου θα γίνει η εισαγωγή,
  2. το κλειδί που θέλουμε να προσθέσουμε (κατά την κάθοδο στο δένδρο),

  και επιστρέφει το ζεύγος (αναφορά σε παιδί, κλειδί) που θέλουμε να προσθέσουμε (κατά την άνοδο στο δένδρο).

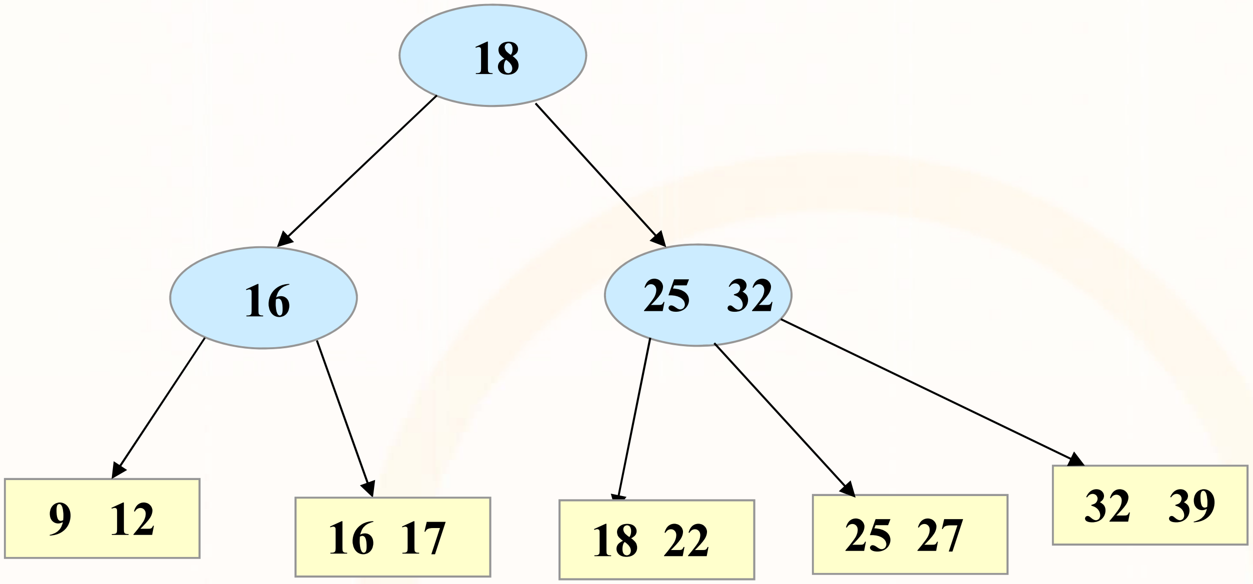
- Η διαδικασία εξαγωγής κλειδιών χρησιμοποιεί παρόμοιες ιδέες. Σ’αυτή όμως την περίπτωση, αντί *διάσπαση* κόμβου έχουμε *συγχώνευση* κόμβων.
- Όλες οι διαδικασίες έχουν χρόνο εκτέλεσης ανάλογο με το ύψος του δένδρου. Άρα είναι τάξης Θ(log n).

Διαδικασία Εξαγωγής Κόμβου

- Η διαδικασία εξαγωγής κλειδιών χρησιμοποιεί παρόμοιες ιδέες.
- Δηλαδή υπάρχουν πάλι οι τρεις φάσεις αλλά αντί ***διάσπαση*** κόμβου έχουμε ***συγχώνευση*** κόμβων.
- Όλες οι διαδικασίες έχουν χρόνο εκτέλεσης ανάλογο με το ύψος του δένδρου. **Άρα είναι τάξης O(log n).**

Παραλλαγή

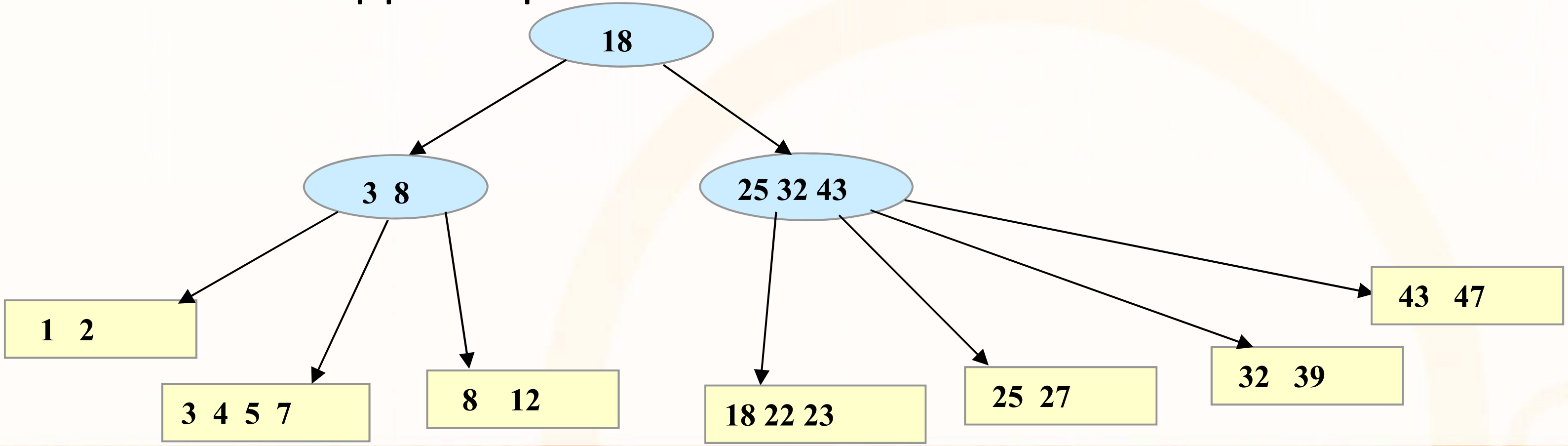
- **Οι πληροφορίες αποθηκεύονται μόνο στα φύλλα.**
- Τα **κλειδιά εσωτερικών κόμβων χρησιμεύουν μόνο για την αναζήτηση στο δένδρο.**
- Δηλαδή, για κάποιο μη-τερματικό κόμβο u, το u.key1 δηλώνει το μικρότερο κλειδί στο μεσαίο υπόδενδρο (u.middle), ενώ το u.key2 (αν υπάρχει) δηλώνει το μικρότερο κλειδί στο δεξί υπόδενδρο (u.right).



B-δένδρα

- Γενίκευση του **2-3 δένδρου**: μπορεί να **αποθηκεύει περισσότερα** από **2** κλειδιά σε κάθε κόμβο.

- **Ένα δένδρο τάξης m** ικανοποιεί τις πιο κάτω ιδιότητες:
  - Έχει μια **ρίζα** η οποία έχει **0** ή **περισσότερα παιδιά**.
  - Τα δεδομένα φυλάσσονται στα φύλλα και όλα τα **φύλλα** βρίσκονται στο ίδιο επίπεδο.
  - Ένας εσωτερικός κόμβος με δείκτες **p_0, p_1, …, p_n,** έχει κλειδιά **k_1,…,k_n,** όπου k_i είναι το μικρότερο κλειδί του υποδένδρου που δείχνεται από τον δείκτη p_i.
  - Όλοι οι **εσωτερικοί κόμβοι** (εκτός τη ρίζα) έχουν **από m/2**μέχρι **m** παιδιά, εναλλακτικά οι κόμβοι συμπτύσσονται

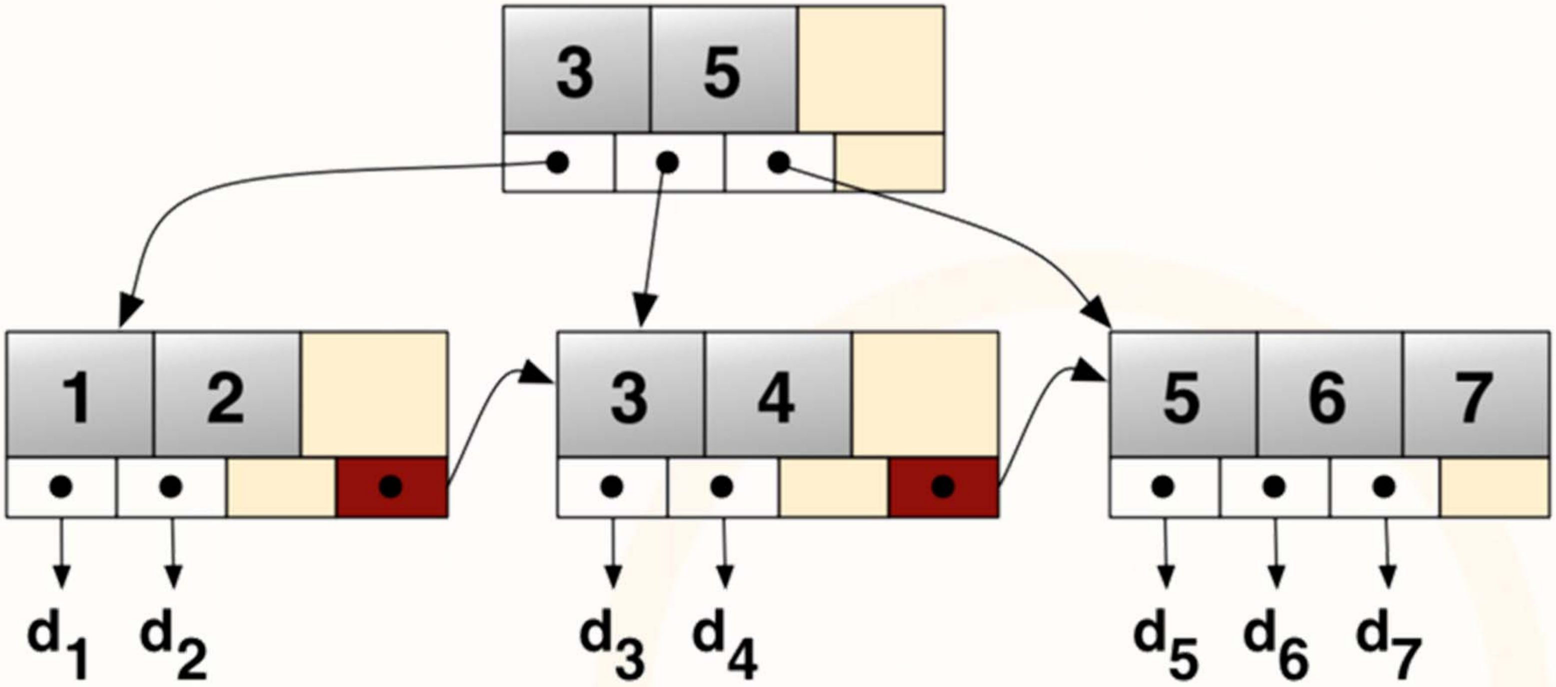


Ένα Β-δένδρο ...

- τάξης **m** με n κλειδιά έχει ύψος $O(\log_m n)$
- Έχει διαδικασίες παρόμοιες με αυτές ενός 2-3 δένδρου, οι οποίες επίσης μπορεί να **προκαλέσουν διάσπαση** ή **συγχώνευση** της ρίζας.
- Διάφορες παραλλαγές τους χρησιμοποιούνται σε βάσεις δεδομένων (Databases) όπου οι κόμβοι αποθηκεύονται σε δευτερεύουσα μνήμη του υπολογιστή
- Το **κόστος πρόσβασης** ενός κόμβου στον **μαγνητικό δίσκο** είναι πολύ μεγαλύτερο απ’ αυτό της επεξεργασίας του. Για αυτό η μείωση του **χρόνου αναζήτησης** είναι πολύ **σημαντική**!
- Η τιμή του m επιλέγεται βάσει του πόσα κλειδιά μπορούν να αποθηκευτούν σε ένα καταχώρημα (block) μνήμης.

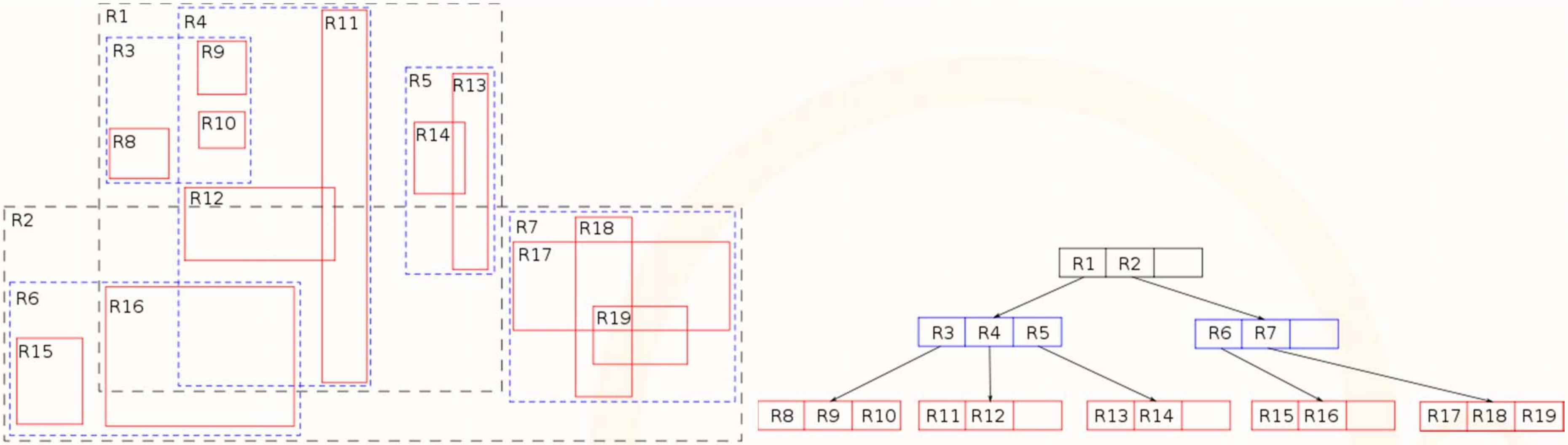
B+ Δέντρα

- Παραλλαγή του B-δέντρου όπου τα φύλλα είναι συνδεδεμένα μέσω αναφορών όπως σε μία συνδεδμένη λίστα
- Πλεονέκτημα όταν θέλουμε να δόύμε τις τιμές κάποιου block διαδοχικά



R-δέντρα

- Τα R-δέντρα χρησιμοποιούνται για χωρικές πληροφορίες, για παράδειγμα:
  - Γεωγραφικές συντεταγμένες
  - Ορθογώνια
  - Πολύγωνα
- Παραδείγματα Επερωτήσεων
  - Βρες όλα τα μουσεία που είναι 2km από την τοποθεσία μου
  - Επέστρεψε όλα τα τμήματα του δρόμου σε απόσταση 5km

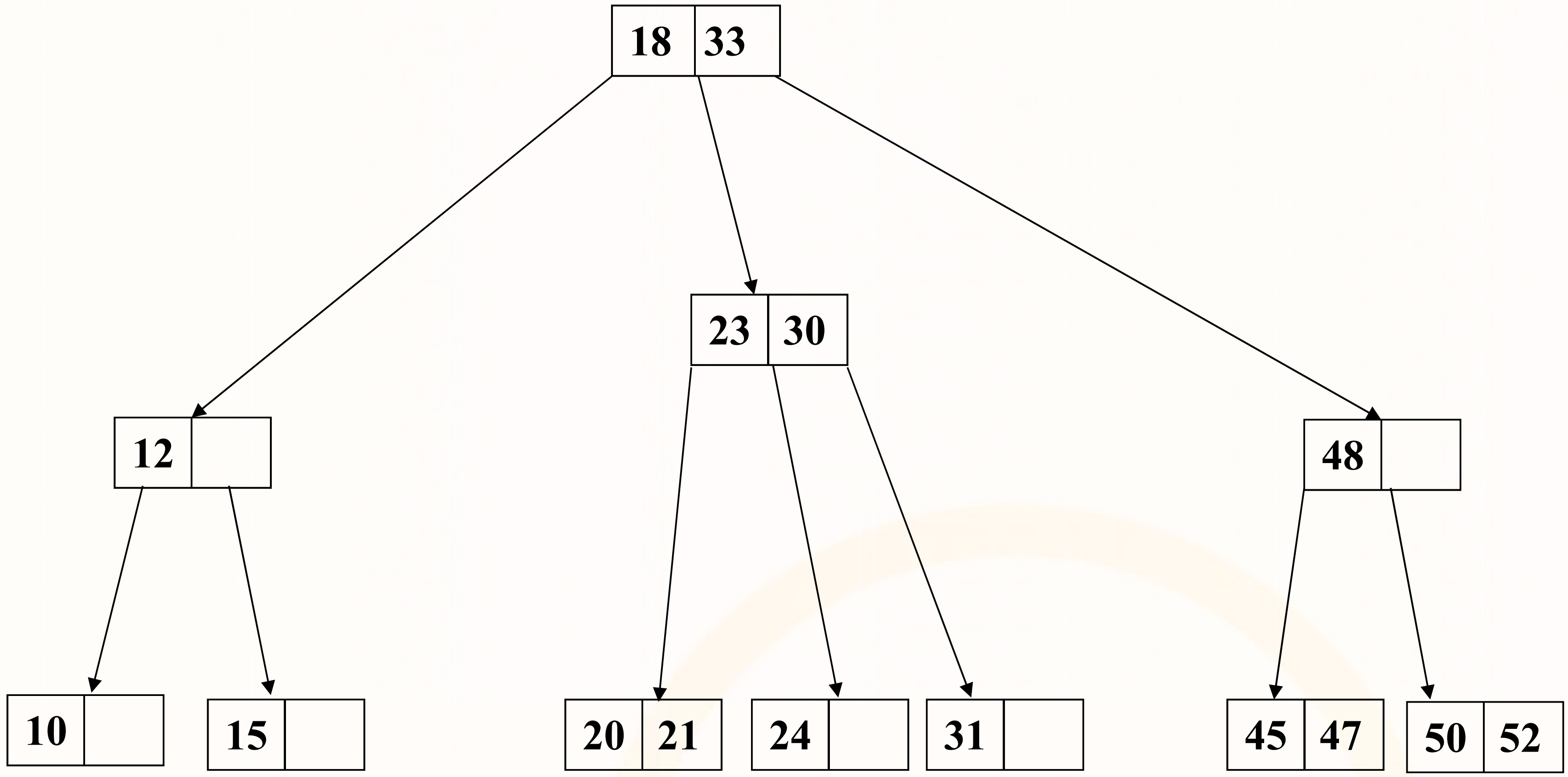


2-3 Δέντρα: Ασκήσεις

- Εισάγετε τα στοιχεία 65, 76, 71, 79, 82, 73, 84, 72, 77, 83 σε ένα 2-3 δέντρο

2-3 Δέντρα: Ασκήσεις

- Εισάγετε το στοιχείο 19 στο πιο κάτω 2-3 δέντρο



2-3 Δέντρα: Ασκήσεις

- Υλοποιήστε μία αποδοτική συνάρτηση που να επιστρέφει το ύψος ενός 2-3 δέντρου σε χρόνο τάξης Θ(h)