



Διάλεξη 24: Β-Δένδρα

Στην ενότητα αυτή θα μελετηθούν τα εξής επιμέρους θέματα:

- Εισαγωγή & Ισοζυγισμένα Δένδρα
- 2-3 Δένδρα, Εισαγωγή και άλλες πράξεις
- Β-δένδρα

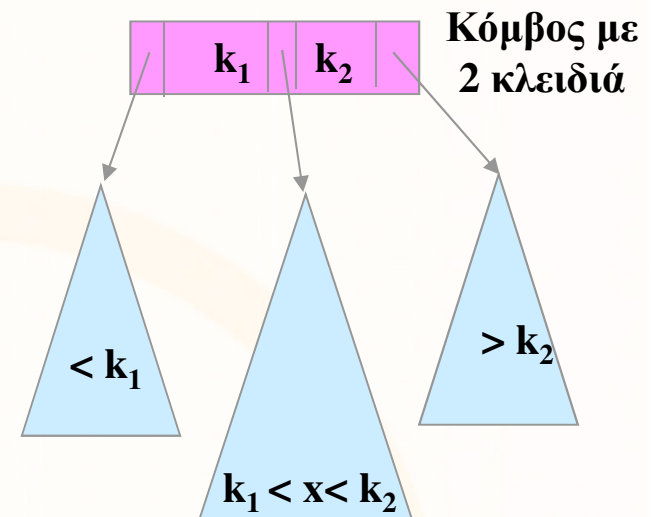
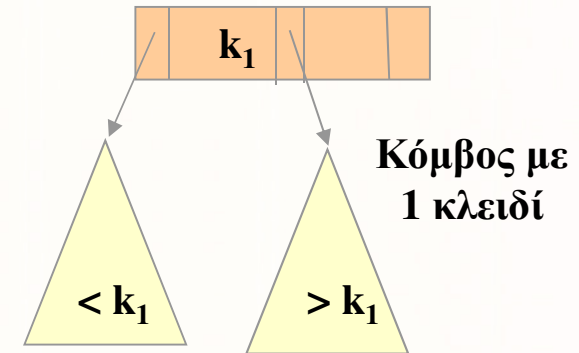
Διδάσκων: Παναγιώτης Ανδρέου

2-3 (2 ή 3 παιδιά) Δένδρα

- Γενίκευση των δυαδικών δένδρων αναζήτησης.

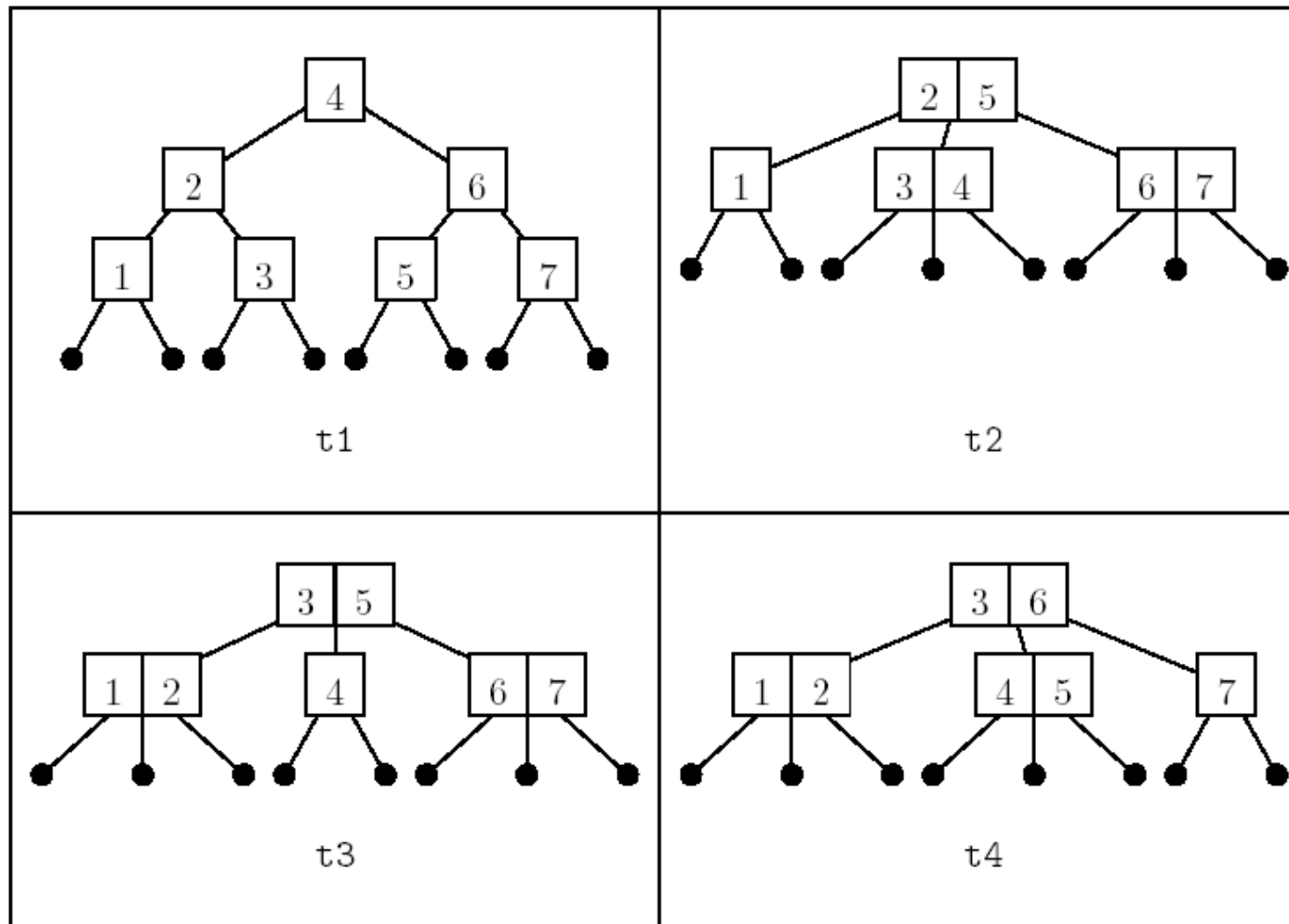
Ορισμός

- Κάθε κόμβος περιέχει **ένα ή δύο** κλειδιά.
- Ένας εσωτερικός κόμβος u με ένα κλειδί, 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 δύο κλειδιά, $k_1 < k_2$, 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 $k_1 < x < k_2$ και όλα τα κλειδιά του $u.right$ είναι $> k_2$.
- Όλα τα φύλλα βρίσκονται στο ίδιο επίπεδο.



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;
```

Εισαγωγή κόμβου σε ένα 2-3 δένδρο

Η εισαγωγή κάποιου κλειδιού k σε ένα 2-3 Δένδρο μπορεί να χωριστεί στις ακόλουθες τρεις λογικές 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)

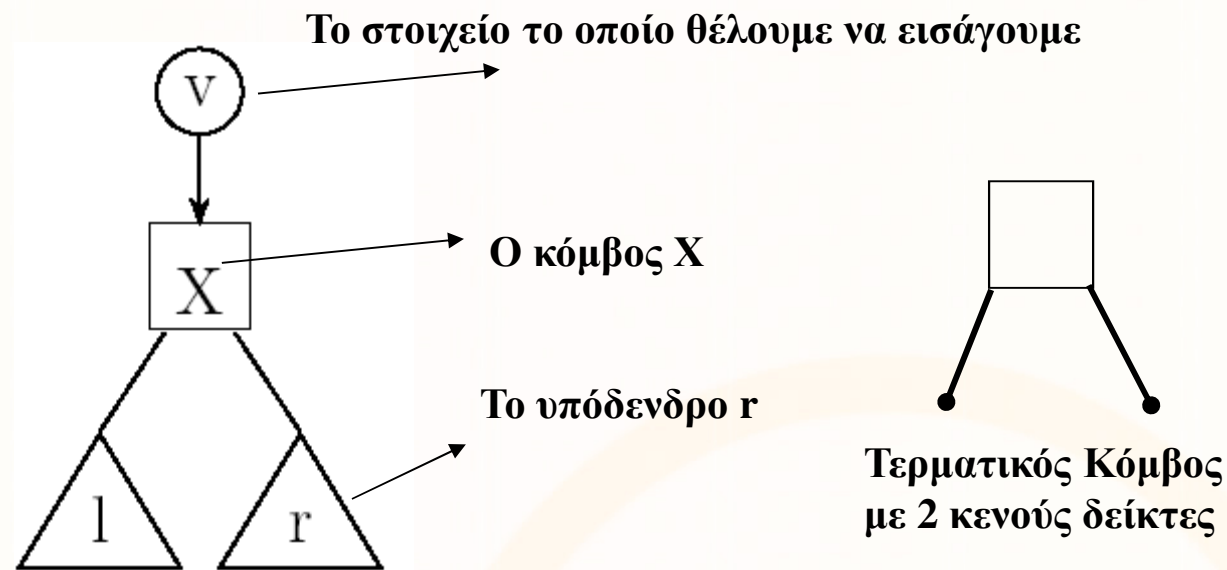
- Όταν φτάσουμε σε φύλλο τότε προσπαθούμε να κάνουμε την εισαγωγή
- Αν δεν έχει αρκετό χώρο τότε διασπάται το φύλλο και «ανασηκώνουμε» (kick up) αναδρομικά το μεσαίο στοιχείο στον πατέρα.

3. Ανοδική Φάση (Upward Phase)

- Σε αυτή την φάση κάνουμε την εισαγωγή αναδρομικά στον γονέα αν δεν πετύχαμε εισαγωγή στην τερματική φάση.

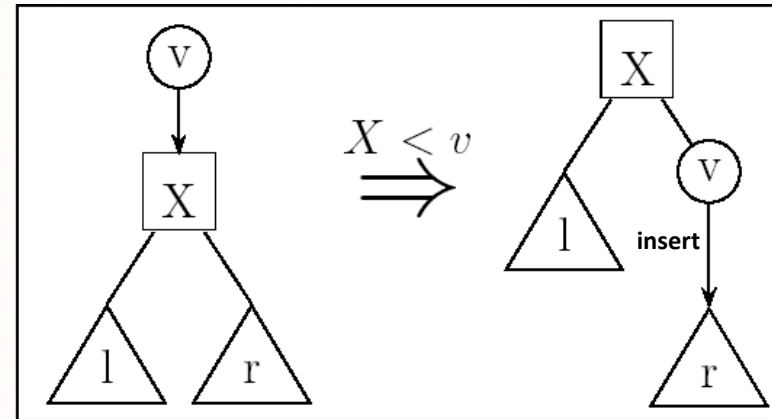
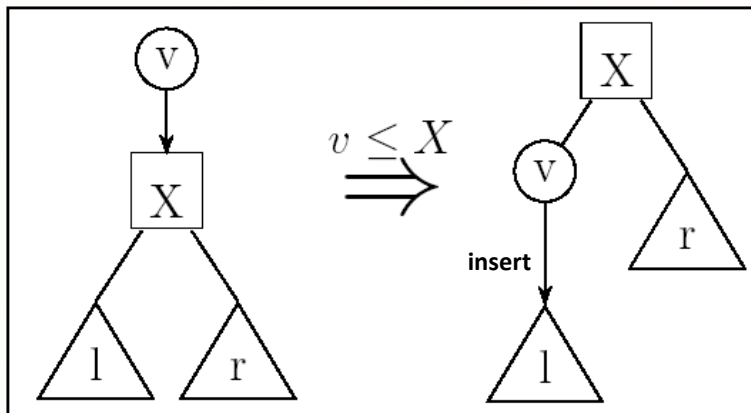
Περιγραφή Κάποιων Συμβολισμών

- Στις επόμενες διαφάνειες θα χρησιμοποιήσουμε τους ακόλουθους συμβολισμούς

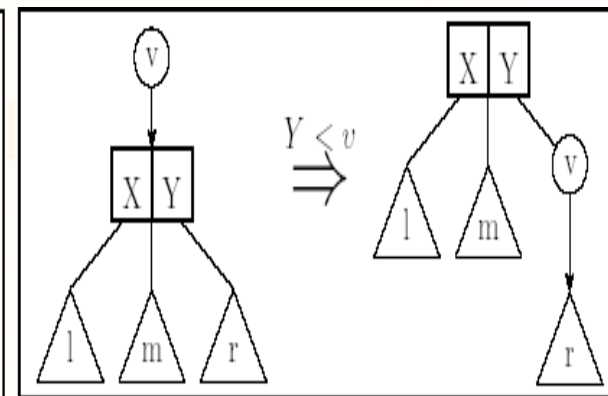
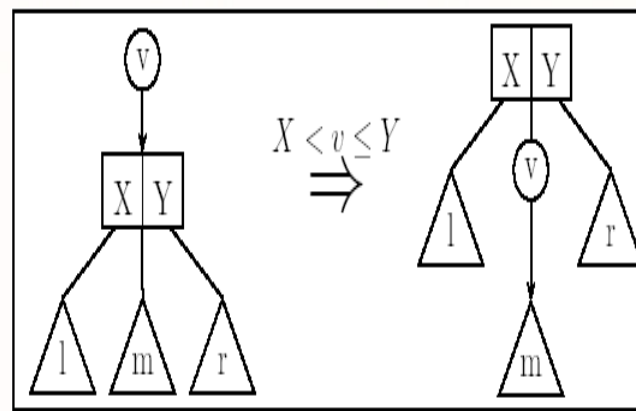
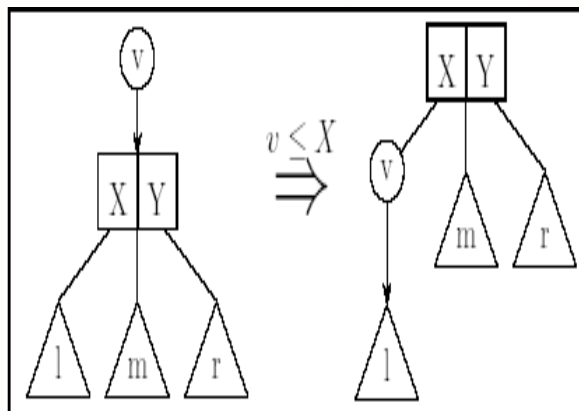


1) Καθοδική Φάση (Downward Phase)

Εισαγωγή του στοιχείου v – Η ρίζα περιέχει 1 στοιχείο



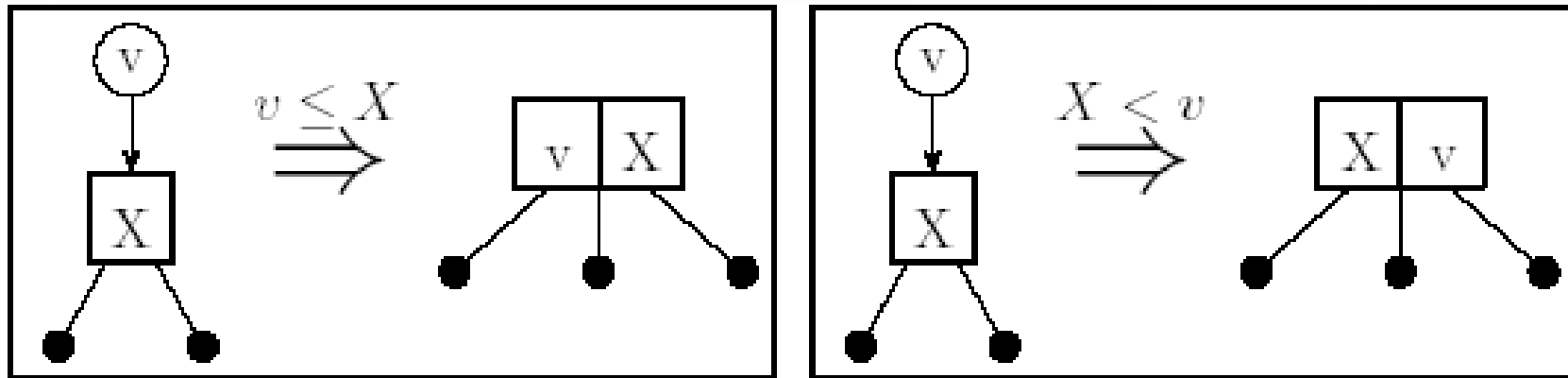
Εισαγωγή του στοιχείου v – η ρίζα περιέχει 2 στοιχεία



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

2) Τερματική Φάση (Terminal Phase)

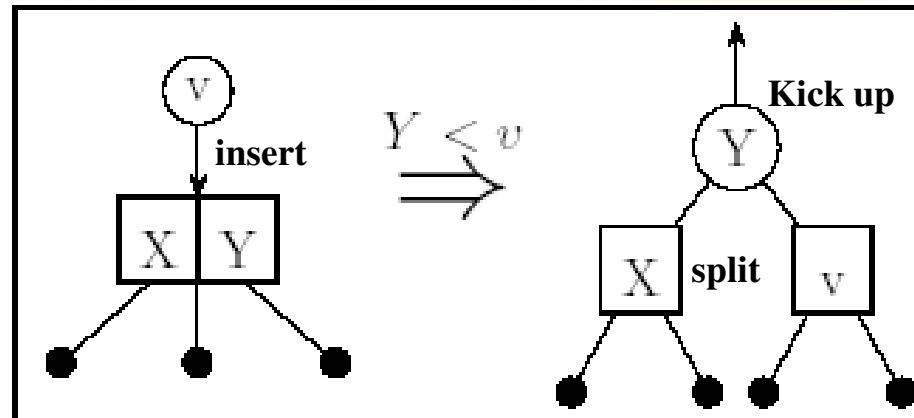
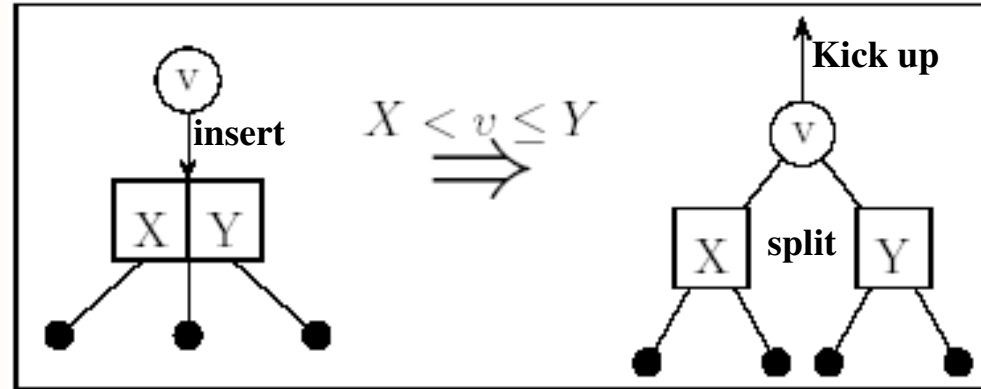
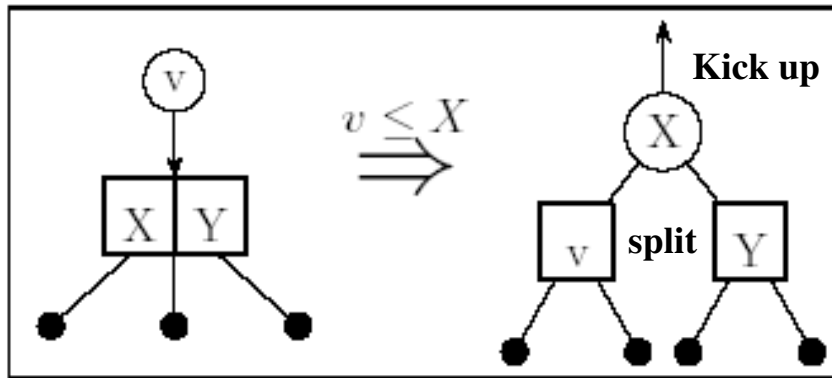
Όταν φτάσουμε σε τερματικό κόμβο και υπάρχει χώρος τότε μπορούμε να κάνουμε την εισαγωγή κατευθείαν. Δηλαδή:



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

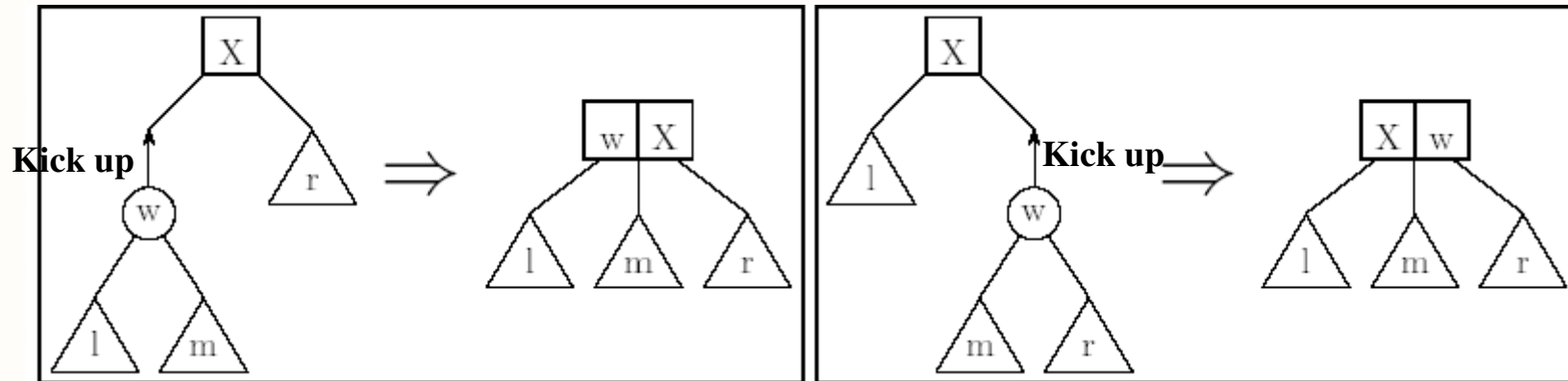
2) Τερματική Φάση (Terminal Phase) (συν.)

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



3: Ανοδική Φάση (Upward Phase)

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

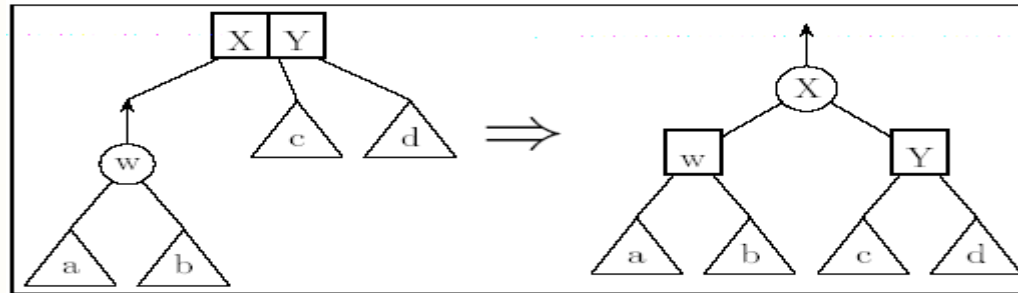


- Αν **ΔΕΝ** υπάρχει χώρος τότε πρέπει να διασπαστεί ο πατέρας και να **προαχθεί (kick-up)** το μεσαίο στοιχείο στον πατέρα του πατέρα (δες επόμενη διαφάνεια)

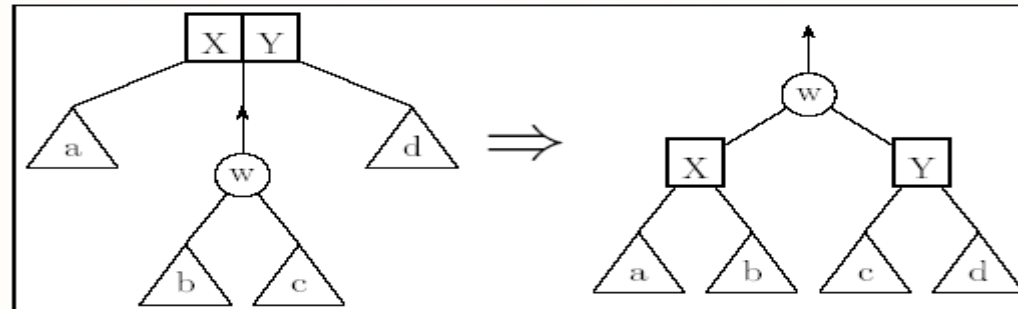
3: Ανοδική Φάση (Upward Phase) συνέχεια

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

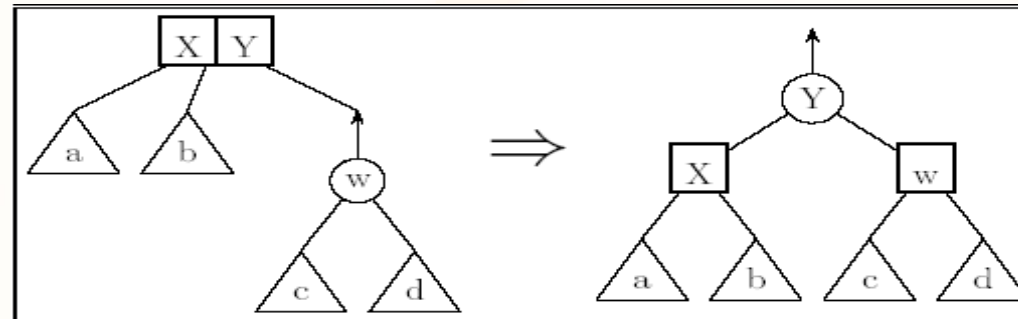
$w \leq X < Y$



$X < w \leq Y$



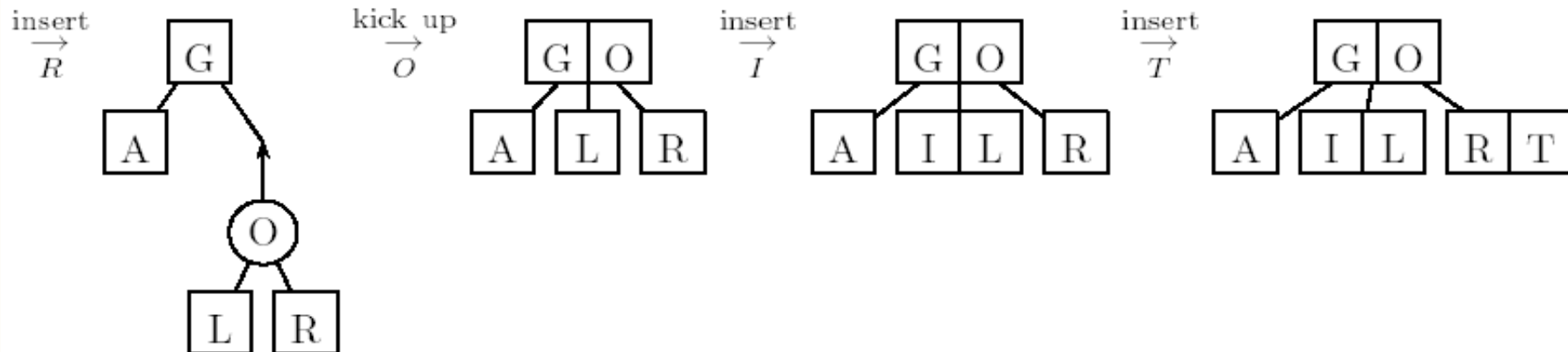
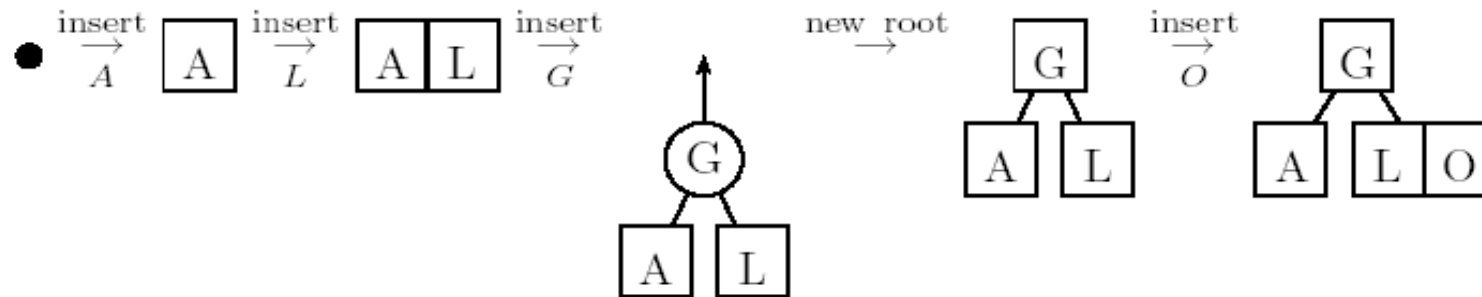
$Y < w$



Ολοκληρωμένο Παράδειγμα Εισαγωγής

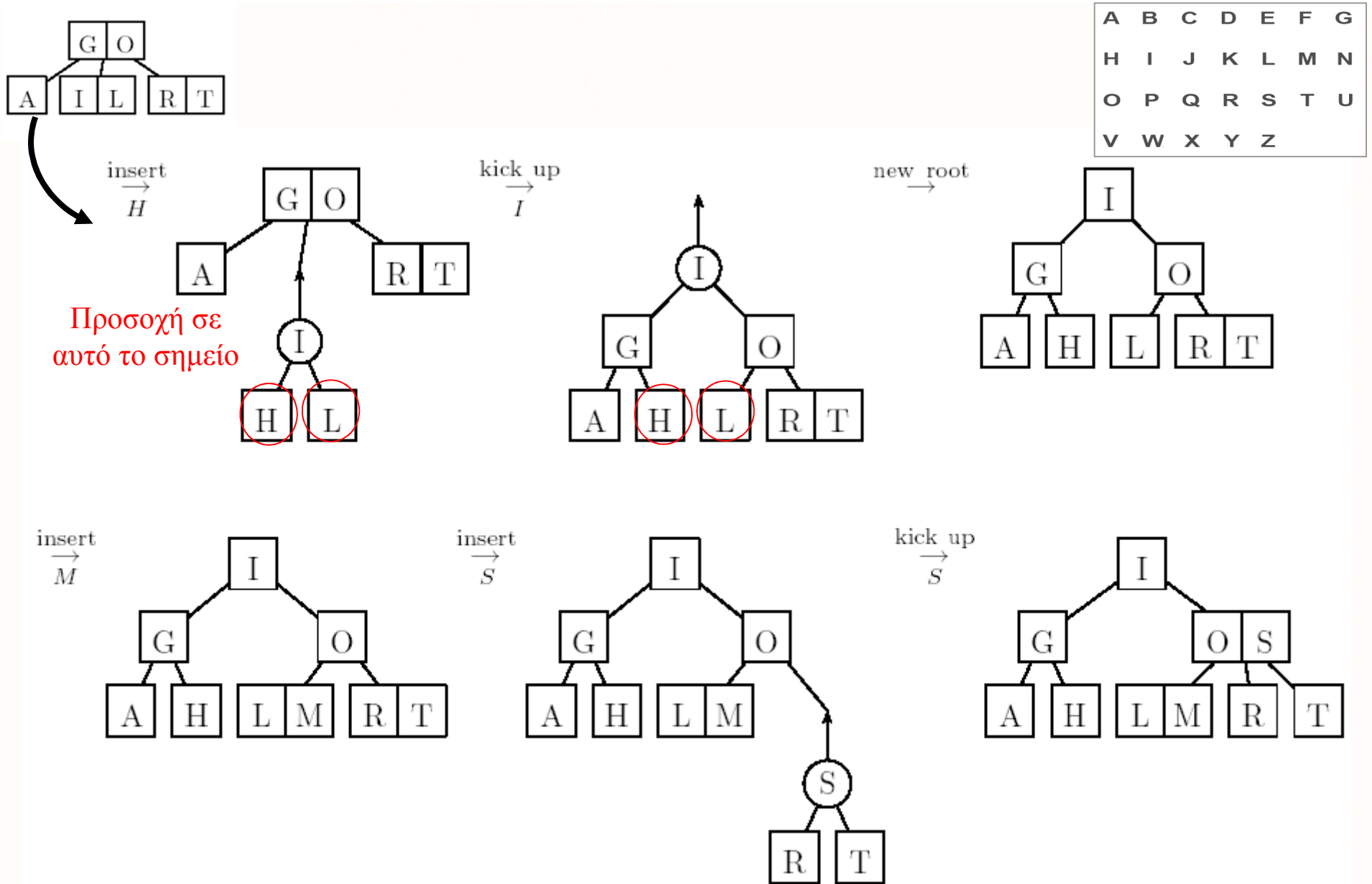
- Παράδειγμα εισαγωγής της λέξης
“ALGORITHMS” (γράμμα-γράμμα)
σε ένα κενό 2-3 δένδρο.

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.)

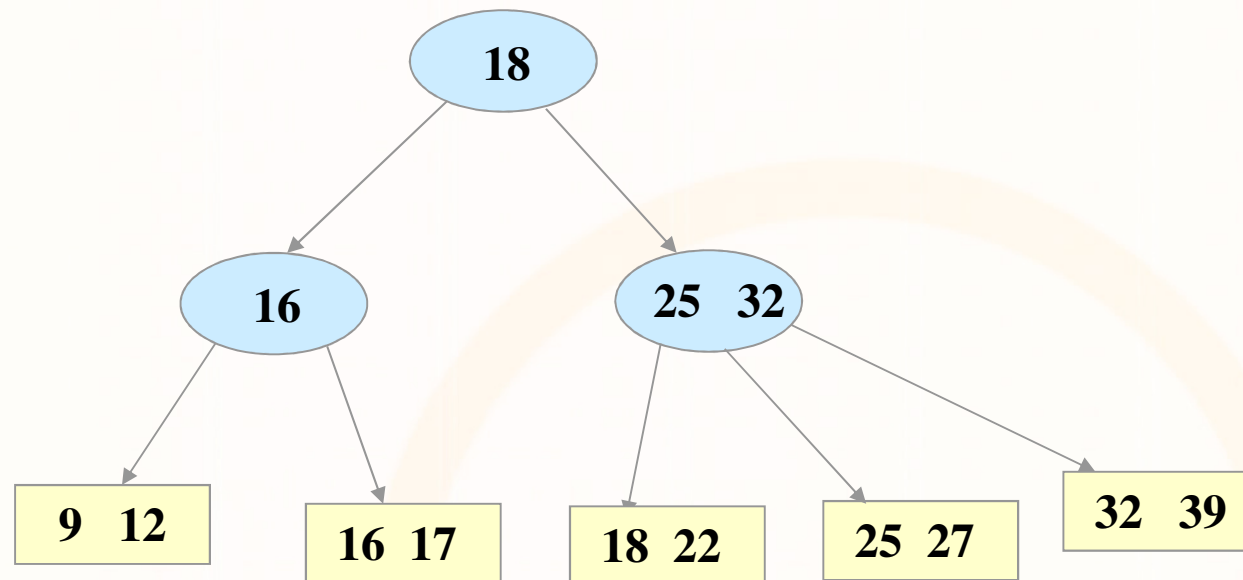


Διαδικασία Εξαγωγής Κόμβου

- 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 $O(\log n)$.**

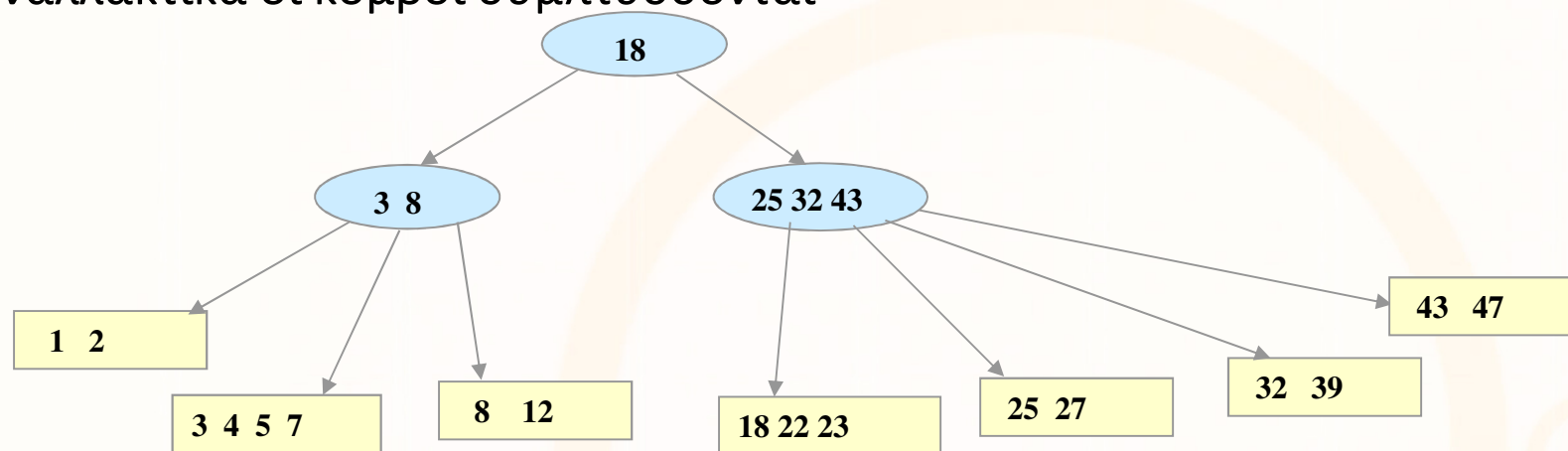
Παραλλαγή

- 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, \dots, p_n , έχει κλειδιά k_1, \dots, 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 συμπύσσονται



Ένα B-δένδρο ...

- τάξης m με n κλειδιά έχει ύψος
- 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!**