

---

# Προηγμένα Θέματα Βάσεων Δεδομένων

Διδάσκων: Άγγελος Μιχάλας

# Περιεχόμενα

---

- B-δένδρα
- **B\*-δένδρα**
- **B<sup>+</sup>-δένδρα**
- Άλλες παραλλαγές των B-δένδρων

# B\*-δένδρα

---

- Η μέση τιμή του παράγοντα χρησιμοποίησης χώρου (69%),  $U$ , είναι δυνατό να αυξηθεί ακολουθώντας μια τεχνική κατά την εισαγωγή και διαγραφή κλειδιών:
  1. αντί να γίνει διάσπαση κατά την εισαγωγή ενός κλειδιού, να αποθηκευτούν κάποια κλειδιά σε γειτονικό κόμβο (αν υπάρχει χώρος) δηλαδή αναδιανομή
  2. αν είναι γεμάτοι οι γειτονικοί κόμβοι, τότε διασπώνται 2 κόμβοι (ο υπερχειλίζοντας και ένας γειτονικός) και προκύπτουν 3 κόμβοι.
- Προτάθηκε από τους *Bayer* και *McCreight*.

# B\*-δένδρα

---

- Μέση τιμή του παράγοντα **χρησιμοποίησης χώρου** σε ένα **B-δένδρο** είναι **69%**.
- Με το προηγούμενο τέχνασμα, η ελάχιστη τιμή του παράγοντα **χρησιμοποίησης χώρου** σε ένα **B\*-δένδρο** καθίσταται  $U_{\min}=67%$ , οπότε για τη **μέση τιμή** ισχύει  $E[U]=81%$ .
- ✚ Εκτός από την καλύτερη χρήση χώρου, ένα άλλο πλεονέκτημα είναι η ταχύτερη αναζήτηση.
- Μειονέκτημα είναι το αυξημένο κόστος κατά τις εισαγωγές και τις διαγραφές.

# B\*-δένδρα

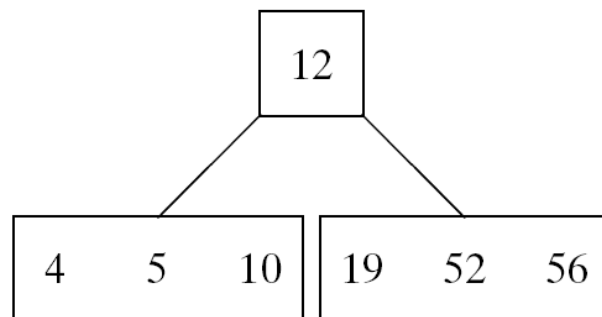
- **B\*-δένδρο τάξης  $m$**  είναι ένα δένδρο αναζήτησης  $m$  δρόμων όπου:
  - η ρίζα έχει το **ελάχιστο** 2 παιδιά (**1 κλειδί**) και το **μέγιστο**  $2 \lceil (2m-2)/3 \rceil + 1$  παιδιά ( **$2 \lceil (2m-2)/3 \rceil$  κλειδιά**),
  - οι εσωτερικοί κόμβοι, εκτός της ρίζας, έχουν το **ελάχιστο**  $\lceil (2m-2)/3 \rceil + 1$  παιδιά ( **$\lceil (2m-2)/3 \rceil$  κλειδιά**) και το μέγιστο  $m$  παιδιά ( **$m-1$  κλειδιά**),
  - ένας εσωτερικός κόμβος με  $k$  κλειδιά έχει  $k+1$  παιδιά, και
  - οι εξωτερικοί κόμβοι βρίσκονται στο ίδιο επίπεδο.

# B\*-δένδρα

B-δένδρο 7<sup>ης</sup> τάξης

4 5 10 12 19 52

Η ρίζα έχει min=1, max=6 κλειδιά

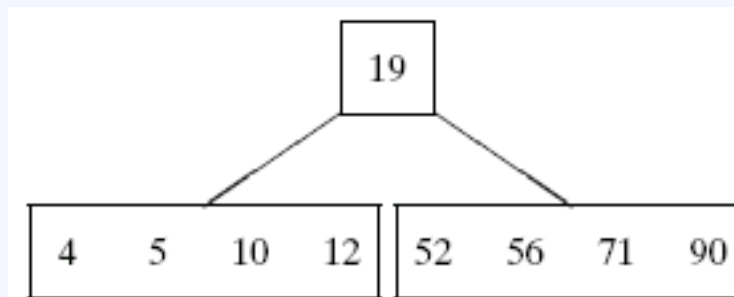


Οι κόμβοι έχουν min=3, max=6 κλειδιά

B\*-δένδρο 7<sup>ης</sup> τάξης

4 5 10 12 19 52 56 71

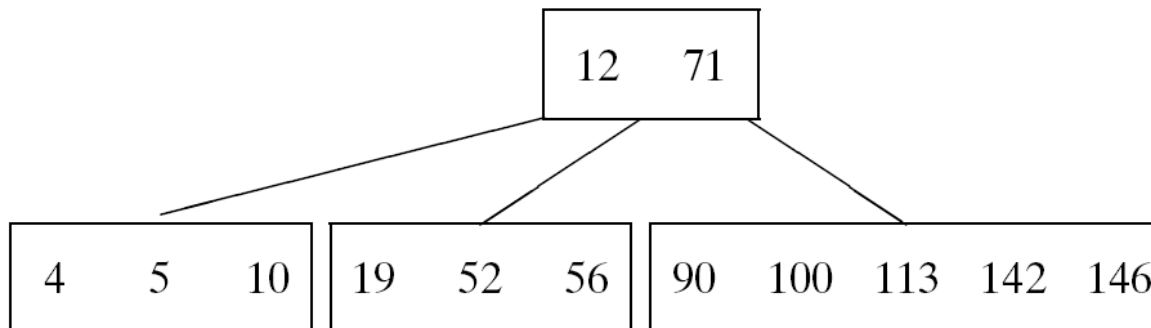
Η ρίζα έχει min=1, max=8 κλειδιά



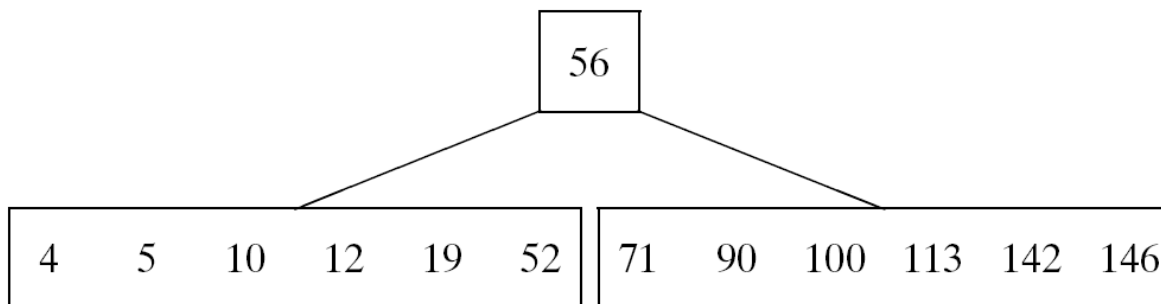
Οι κόμβοι έχουν min=4, max=6 κλειδιά

# B\*-δένδρα

B-δένδρο



B\*-δένδρο



- Εισαγωγή 100, 113, 142, 146 στο B\*-Δένδρο
  - + 71, 90 στο B-Δένδρο

# B\*-δένδρα

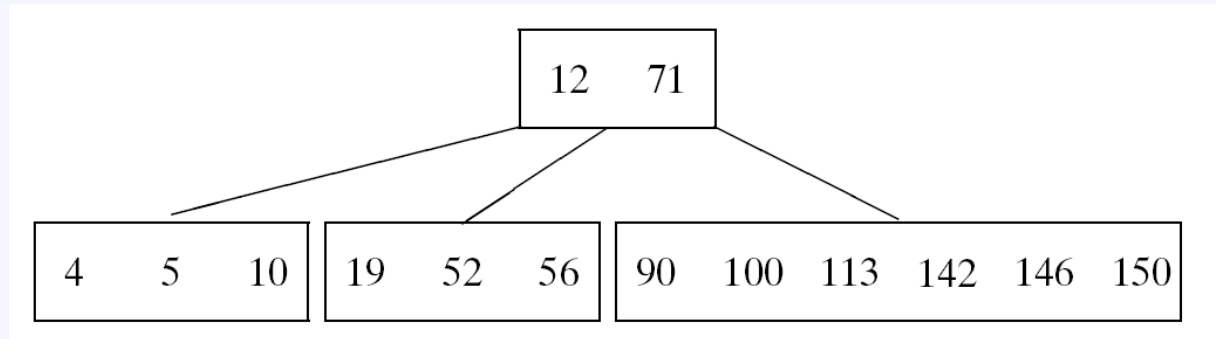
---

- Σε περίπτωση υπερχειλίσης, αν δεν υπάρχει χώρος σε γειτονικό κόμβο, τότε:
  1. δημιουργείται νέος κόμβος,
  2. γίνεται αναδιανομή στους τρεις κόμβους,
    - Τα κλειδιά που υπεισέρχονται στη διαδικασία της αναδιανομής είναι **τα κλειδιά των δύο κόμβων, το κλειδί από το ανώτερο επίπεδο και το εισαγόμενο κλειδί**
  3. τα κλειδιά που θα ανέλθουν στο ανώτερο επίπεδο είναι τα:
    - **$\alpha = (2m+1)/3$ -οστό και  $\beta = (2m+1+\alpha)/2$ -οστό.**
    - Π.χ. για  $m=7$   $\alpha=5$  και  $\beta=10$

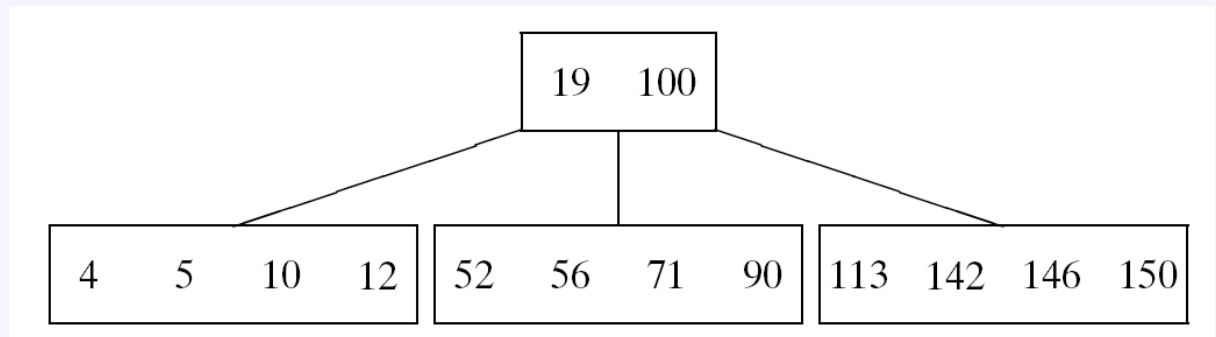


# B\*-δένδρα

B-δένδρο



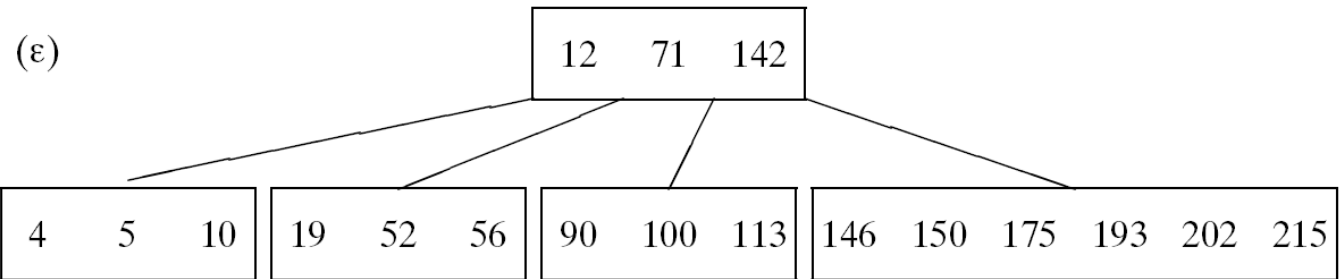
B\*-δένδρο



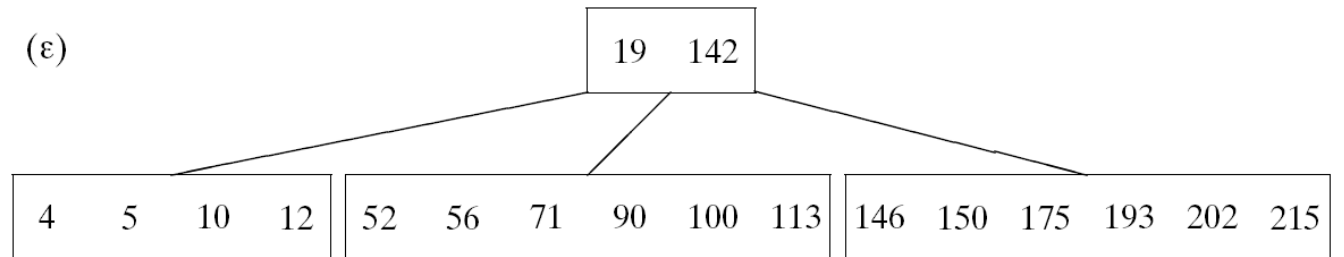
- Εισαγωγή 150

# B\*-δένδρα

B-δένδρο



B\*-δένδρο



- Εισαγωγή 175, 193, 202, 215

# B\*-δένδρα

---

- Η τεχνική της διάσπασης 2 κόμβων σε 3 μπορεί να γενικευθεί με διάσπαση 3 κόμβων σε 4, 4 κόμβων σε 5 κοκ.
- + Στις περιπτώσεις αυτές, η ελάχιστη τιμή του παράγοντα χρησιμοποίησης χώρου θα ισούται αντίστοιχα με 75%, 80% κοκ.
- + Καλύτερος χρόνος αναζήτησης έναντι του B-δένδρου
- + Γίνεται σπανιότερα η διάσπαση σε σχέση με το B-δένδρο

# B\*-δένδρα

---

- Το τίμημα έναντι του κέρδους σε καλύτερο παράγοντα χρησιμοποίησης χώρου είναι ο **αυξημένος χρόνος για εισαγωγές και διαγραφές**.

# B<sup>+</sup>-δένδρα

---

- Τα B<sup>+</sup>-δένδρα και οι παραλλαγές τους είναι η κύρια δομή που χρησιμοποιούν τα εμπορικά συστήματα διαχείρισης βάσεων δεδομένων.
- Σε αντίθεση με τα B-δένδρα όπου κάθε τιμή του πεδίου αναζήτησης εμφανίζεται μια φορά σε κάποιο επίπεδο του δένδρου **στα B<sup>+</sup>-δένδρα οι δείκτες δεδομένων αποθηκεύονται μόνο στα φύλα.**
- Επομένως οι κόμβοι φύλα διαφέρουν από τους εσωτερικούς κόμβους.
- *Προτάθηκε από τον Knuth το 1973*

# B<sup>+</sup>-δένδρα

## Ορισμός B<sup>+</sup>-δένδρου (Εσωτερικοί κόμβοι)

1. Κάθε εσωτερικός κόμβος είναι της μορφής  $\langle P_1, K_1, P_2, K_2, \dots, K_{q-1}, P_q \rangle$   $q \leq p$  και κάθε  $P_i$  είναι δείκτης δένδρου
2. Σε κάθε εσωτερικό κόμβο  $K_1 < K_2 < \dots < K_{q-1}$ .
3. Κάθε εσωτερικός κόμβος έχει το πολύ  $p$  δείκτες δένδρου.
4. Κάθε εσωτερικός κόμβος εκτός από τη ρίζα έχει το λιγότερο  $\lceil p/2 \rceil$  δείκτες δένδρου. Ο κόμβος της ρίζας έχει τουλάχιστον δυο δείκτες δένδρου.
5. Ένας εσωτερικός κόμβος με  $q$  δείκτες,  $q \leq p$  έχει  $q-1$  τιμές πεδίου αναζήτησης

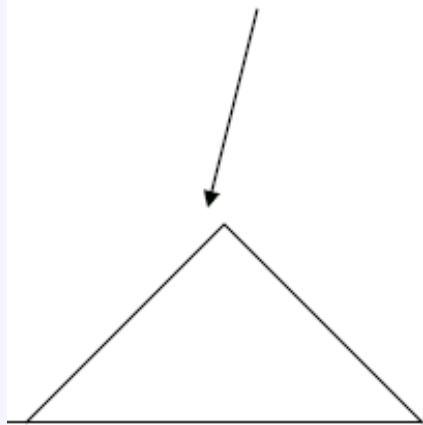
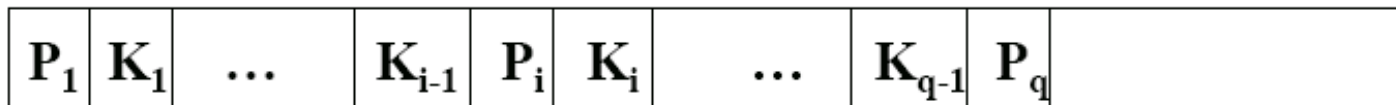
# B<sup>+</sup>-δένδρα

## Ορισμός B<sup>+</sup>δένδρου (Κόμβοι φύλα)

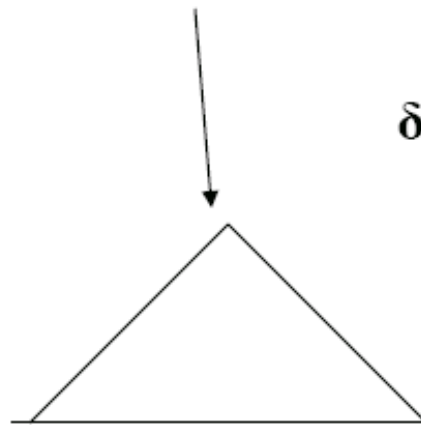
1. Κάθε κόμβος φύλο είναι της μορφής:  
 $\langle \langle K_1, Pr_1 \rangle, \langle K_2, Pr_2 \rangle, \dots, \langle K_{q-1}, Pr_{q-1} \rangle, P_{next} \rangle$   $q \leq p$  κάθε  $Pr_i$  είναι δείκτης δεδομένων και το  $P_{next}$  δείχνει τον επόμενο κόμβο φύλο του B<sup>+</sup>δένδρου.
2. Σε κάθε κόμβο  $K_1 < K_2 < \dots < K_{q-1}$ ,  $q \leq p$ .
3. Κάθε  $Pr_i$  είναι δείκτης δεδομένων που δείχνει την εγγραφή της οποίας η τιμή του πεδίου του ευρετηρίου είναι  $K_i$ .
4. Κάθε κόμβος φύλο έχει το λιγότερο  $\lceil p/2 \rceil$  τιμές.
5. Όλοι οι κόμβοι φύλα είναι στο ίδιο επίπεδο.

# B<sup>+</sup>-δένδρα

## Δομή εσωτερικού κόμβου

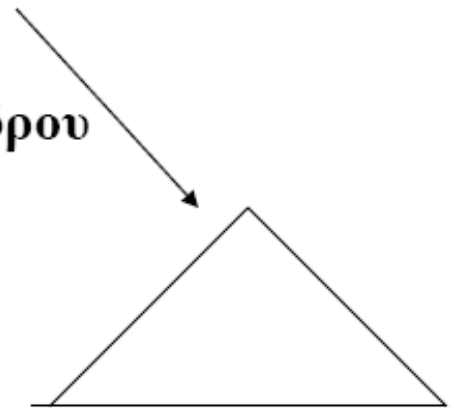


$$X \leq K_1$$



$$K_{i-1} \leq X \leq K_i$$

δείκτης δένδρου

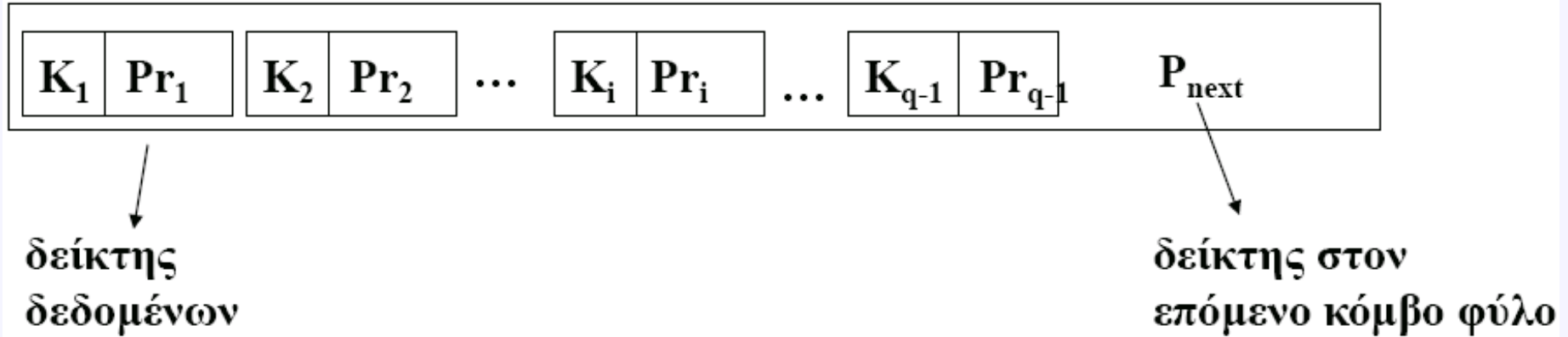


$$K_{q-1} < X$$



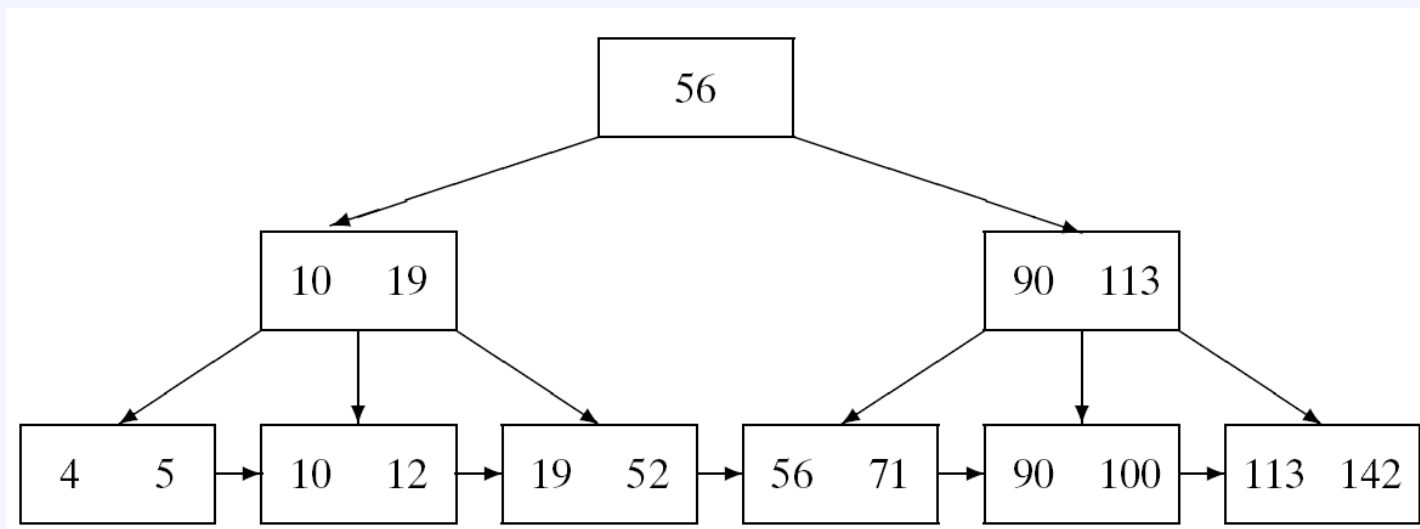
# B<sup>+</sup>-δένδρα

## Δομή κόμβου φύλου



# B<sup>+</sup>-δένδρα

- Το B<sup>+</sup>-δένδρο είναι ετερογενές, τα δεδομένα βρίσκονται μόνο στα φύλλα.
- Οι αριθμοί στα φύλλα δηλώνουν τα κλειδιά εγγραφών, ενώ στους εσωτερικούς κόμβους δηλώνουν απλά κλειδιά.
- Τα φύλλα περιέχουν δείκτες προς το επόμενο φύλλο για να διευκολύνεται η σειριακή επεξεργασία.



# B<sup>+</sup>-δένδρα

---

- Η δομή υποδιαιρείται σε δύο λογικά μέρη, όπου το ανώτερο εξυπηρετεί το κατώτερο ως κατάλογος.
  - Ο τύπος των εσωτερικών κόμβων είναι διαφορετικός από τον τύπο των εξωτερικών.
- ✚ Τα B<sup>+</sup>-δένδρα είναι:
- ρηχά και εξαιρετικά πλατιά, και
  - πολύ αποτελεσματικά και χρησιμοποιούνται για ερωτήσεις διαστήματος (Για απλά ερωτήματα χρησιμοποιείται ο κατακερματισμός).

# B<sup>+</sup>-δένδρα

---

- Σημείωση: Θεωρούμε ότι τα δεδομένα βρίσκονται σε ταξινομημένη σειρά
  - Αυτό ονομάζεται ως **πρωτεύον ευρετήριο-κατάλογος (primary index)**
- **Δευτερεύον κατάλογος (secondary index):**
  - Δημιουργείται πάνω σε ένα attribute στο οποίο το αρχείο δεν είναι ταξινομημένο.

# B<sup>+</sup>-δένδρα

---

- **Ομαδοποιημένος (clustered)** λέγεται ο κατάλογος πρωτεύοντος κλειδιού, όπου οι εγγραφές είναι αποθηκευμένες σε γειτονικούς κάδους κατά απόλυτη τάξη του κλειδιού.
- **Πυκνός (dense)** λέγεται ο κατάλογος, όπου **κάθε κλειδί σχετίζεται με ένα συγκεκριμένο δείκτη** προς μία **εγγραφή** του ταξινομημένου αρχείου.

Θεωρούμε ότι ένας κάδος ταυτίζεται με μια σελίδα και ο όρος κάδος χρησιμοποιείται ισοδύναμα με την έννοια του κόμβου

# B<sup>+</sup>-δένδρα

---

- **Αραιός (sparse)** λέγεται ο κατάλογος, όπου **κάθε κλειδί σχετίζεται με ένα συγκεκριμένο δείκτη** προς ένα **block** του ταξινομημένου αρχείου.
- **Αραιός:** στο ανώτερο επίπεδο από τα φύλλα ο κατάλογος περιέχει μόνο την τιμή του μικρότερου κλειδιού κάθε κάδου.

# B<sup>+</sup>-δένδρα

---

- Αν το B<sup>+</sup>-δένδρο χρησιμοποιείται ως **κύριος κατάλογος**, τότε τα φύλλα περιέχουν **εγγραφές**.
- Αν το B<sup>+</sup>-δένδρο χρησιμοποιείται ως **δευτερεύον κατάλογος**, τότε τα φύλλα περιέχουν **κλειδιά και δείκτες προς εγγραφές**.

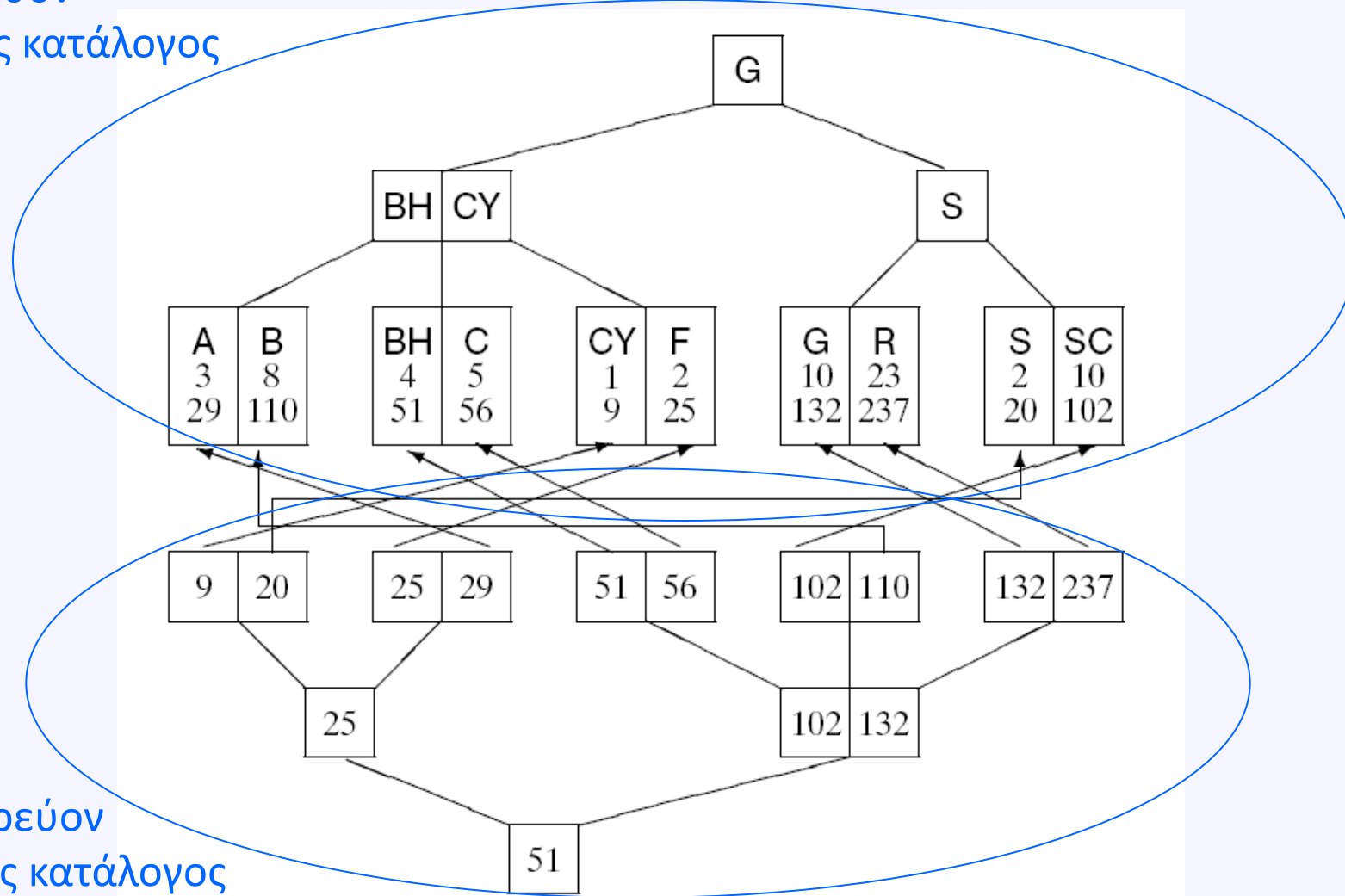
# B<sup>+</sup>-δένδρα

Χώρα	Πληθυσμός	Έκταση
Αλβανία (A)	3	29
Βοσνία-Ερζεγοβίνη (BH)	4	51
Βουλγαρία (B)	8	110
Ελλάδα (G)	10	132
Κροατία (C)	5	56
Κύπρος (CY)	1	9
ΠΓΔΜ (F)	2	25
Ρουμανία (R)	23	237
Σερβία-Μαυροβούνιο (SC)	10	102
Σλοβενία (S)	2	20



# B<sup>+</sup>-δένδρα

Πρωτεύον  
Αραιός κατάλογος



Δευτερεύον  
Πυκνός κατάλογος

# B<sup>+</sup>-δένδρα

---

- Είναι ευθύνη του σχεδιαστή των αρχείων να αποφασίσει το **πόσους** και **ποιους καταλόγους** θα δημιουργήσει.
  - στην περίπτωση που θα γίνει ερώτηση με βάση δευτερεύον κλειδί για το οποίο **δεν έχει δημιουργηθεί κατάλογος**, η αναζήτηση ταυτίζεται με την **αναζήτηση σε αρχείο σωρού**
  - στην περίπτωση που δημιουργείται κατάλογος για κάθε πεδίο, απαιτείται **πολλαπλάσιος χώρος** από τον κύριο χώρο των δεδομένων και το **κόστος ενημέρωσης** των καταλόγων είναι ιδιαίτερα **μεγάλο**.

# B<sup>+</sup>-δένδρα

- Μέγιστος-ελάχιστος αριθμός παιδιών για διαφορετικές τάξεις
  - Θεωρούμε ότι η τάξη  $p$  συμβολίζεται ως  $b$ .

Node Type	Children Type	Min Children	Max Children	Example $b = 7$	Example $b = 100$
Root Node (when it is the only node in the tree)	Keys	1	$b$	1 - 7	1 - 100
Root Node	Internal Nodes or Leaf Nodes	2	$b$	2 - 7	2 - 100
Internal Node	Internal Nodes or Leaf Nodes	$\lceil b/2 \rceil$	$b$	4 - 7	50 - 100
Leaf Node	Keys	$\lceil b/2 \rceil$	$b - 1$	3 - 6	50 - 99

# B<sup>+</sup>-δένδρα

Υπολογισμός της τάξης  $p$  του B<sup>+</sup>δένδρου

Μήκος κλειδιού  $V=9$  byte

Μέγεθος block  $B=512$ byte

Μήκος δείκτη υποδένδρου  $P=6$ byte

Μήκος δείκτη εγγραφής  $P_r=7$ byte

Για την τάξη  $p$  του δένδρου πρέπει να ισχύει

$$(p * P) + ((p - 1) * V) \leq B$$

Δηλαδή στην περίπτωση μας

$$(p * 6) + ((p - 1) * 9) \leq 512$$

$$15 * p \leq 521$$

$$P=34$$

# B<sup>+</sup>-δένδρα

Ο κόμβος φύλο μπορεί να είναι άλλης τάξης

Θα πρέπει να ισχύει:

$$(p_{\text{leaf}} * (P_r + V)) + P \leq B$$

$$(p_{\text{leaf}} * (7 + 9)) + 6 \leq 512$$

$$(16p_{\text{leaf}}) \leq 506$$

Δηλαδή  $p_{\text{leaf}} = 31$

# B<sup>+</sup>-δένδρα Παράδειγμα

- Υπολογισμός των εγγραφών ενός B<sup>+</sup>-δένδρου
  - Θεωρούμε ότι κάθε κόμβος είναι 69% γεμάτος
  - Κάθε εσωτερικός κόμβος θα έχει  $34 * 0,69 = 23$  pointers και 22 τιμές-κλειδιά
  - Κάθε φύλλο θα έχει  $31 * 0,69 = 21$  data records pointers
- Can often hold top levels in buffer pool:

– Root:	1 node	22 entries	23 pointers
– Level 1:	23 nodes	506 entries	529 pointers
– Level 2:	529 nodes	11638 entries	12167 pointers
– Level 3:	12167 nodes	255507 record	pointers

# B<sup>+</sup>-δένδρα – Προσπέλαση Εγγραφής

- Για τη μεταφορά δεδομένων, ο όρος **κάδος** ταυτίζεται με τον όρο **κόμβος**.
- **Παράγοντας καδοποίησης (Bkfr)** ορίζεται ο αριθμός των εγγραφών στους εξωτερικούς κόμβους.
- Όλοι οι κόμβοι εκτός από τους κόμβους των τελευταίων 2 επιπέδων αποθηκεύονται στην απομονωτική μνήμη.
- Όταν το δένδρο χρησιμοποιείται ως πρωτεύων κατάλογος και ως δευτερεύων κατάλογος απαιτείται **μια προσπέλαση** για τον κόμβο που βρίσκεται επάνω από το κατάλληλο φύλλο και **μια δεύτερη** για το φύλλο.
  - dtt=χρόνος μεταφοράς δεδομένων ενός κάδου

$$T_{\text{access}} = (s + r + btt) + (s + r + dtt)$$

# B<sup>+</sup>-δένδρα – Εξαντλητική ανάγνωση

Κάθε κόμβος περιέχει το ελάχιστο και το μέγιστο  $d$  και  $2d$  κλειδιά αντιστοίχως, όμως έχει αποδειχθεί ότι κατά μέσο όρο περιέχει  $2d \times \ln 2$  κλειδιά. Συνεπώς, το πλήθος των κάρδων που απαιτούνται για την αποθήκευση του B<sup>+</sup>-δένδρου είναι:

$$\frac{n}{Bkfr \times \ln 2} = bk$$



# B<sup>+</sup>-δένδρα – Εξαντλητική ανάγνωση

- Ο χρόνος για την εξαντλητική ανάγνωση ενός B<sup>+</sup> δένδρου ως πρωτεύων κατάλογος είναι :

$$T_{\text{exhaustive}}(\text{primary}) = bk \times (s + r + dtt) = \frac{n \times (s + r + dtt)}{Bkfr \times \ln 2}$$

- n ο αριθμός εγγραφών,
- Bkfr ο παράγοντας ομαδοποίησης,
- s είναι ο μέσος χρόνος εντοπισμού,
- r είναι ο μέσος χρόνος περιστροφής, και
- dtt είναι ο χρόνος μεταφοράς δεδομένων ενός κάδου
- bk το πλήθος των κάδων.

# B<sup>+</sup>-δένδρα – Εξαντλητική ανάγνωση

- Ο χρόνος για την εξαντλητική ανάγνωση ενός B<sup>+</sup>-δένδρου ως δευτερεύων κατάλογος είναι:

$$T_{\text{exhaustive}}(\text{secondary}) = n \times (s + r + dtt) + n \times \frac{s + r + btt}{2d \times \ln 2}$$

- $n$  είναι το πλήθος των εγγραφών του αρχείου
- ο πρώτος όρος αναφέρεται **στο κόστος ανάγνωσης των εγγραφών μία προς μία**, ενώ
- ο δεύτερος αναφέρεται **στο κόστος ανάγνωσης του τελευταίου επιπέδου του B<sup>+</sup>-δένδρου**. Ο δεύτερος όρος είναι σημαντικά μικρότερος και μπορεί να παραλειφθεί.

# B<sup>+</sup>-δένδρα – Εισαγωγή Αλγόριθμος

1. Εκκινήστε από τη ρίζα, βρείτε το μπλοκ όπου θα αποθηκευτεί η εγγραφή.
2. Εάν το μπλοκ δεν είναι γεμάτο, προσθέστε την εγγραφή.
3. Διαφορετικά, διασπάστε το μπλοκ.
  - 3.1 Δεσμεύστε ένα νέο φύλο και μετακινήστε τα μισά στοιχεία του υπάρχοντος μπλοκ στο νέο μπλοκ.
  - 3.2 **Εισαγάγετε το μικρότερο κλειδί** του νέου κόμβου **στον πατρικό κόμβο**.
  - 3.3 Εάν ο πατρικός κόμβος είναι γεμάτος, διασπάστε τον επίσης.
    - Προσθέστε το **μεσαίο κλειδί** στον πατρικό κόμβο.
  - 3.4 Επαναλάβετε μέχρι να βρεθεί ένα πατρικός κόμβος που δεν χρειάζεται διάσπαση.
4. Εάν η ρίζα διεσπαστεί, δημιουργήστε μια νέα ρίζα που έχει ένα κλειδί και 2 δείκτες.

# $B^+$ -δένδρα – Εισαγωγή

---

- Κατά τη εισαγωγή ενός κλειδιού σε δένδρο, κάθε εξωτερικός κόμβος περιέχει από  $d$  έως  $2d$  κλειδιά.
- Στη περίπτωση υπερχείλισης (σε εξωτερικό κόμβο) γίνεται διάσπαση και στον παλιό κόμβο αποθηκεύονται  $d$  κλειδιά, ενώ στο νέο αποθηκεύονται  $d+1$  κλειδιά.

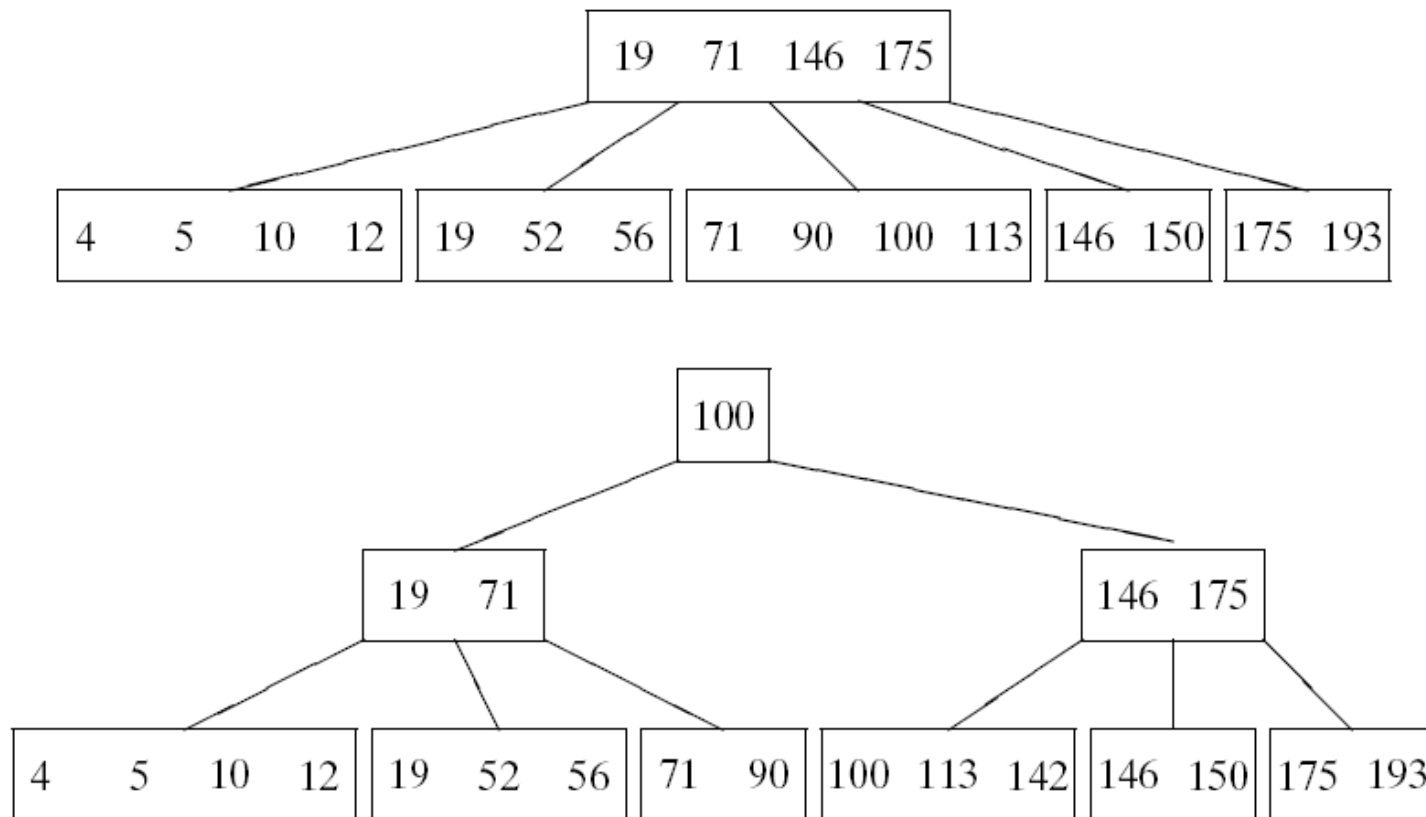
# B<sup>+</sup>-δένδρα – Εισαγωγή

---

- *Η τιμή του μικρότερου κλειδιού του νέου κόμβου (εξωτερικού κόμβου) καθώς και η διεύθυνση του νέου κόμβου ανεβαίνουν προς τον κόμβο πατέρα.*
- Αν αυτή η τιμή-διεύθυνση δεν χωρά, τότε γίνεται διάσπαση στον κόμβο πατέρα. Στον παλιό κόμβο αποθηκεύονται d κλειδιά, στο νέο d κλειδιά, ενώ στο ανώτερο επίπεδο ανέρχεται το **μεσαίο κλειδί**.

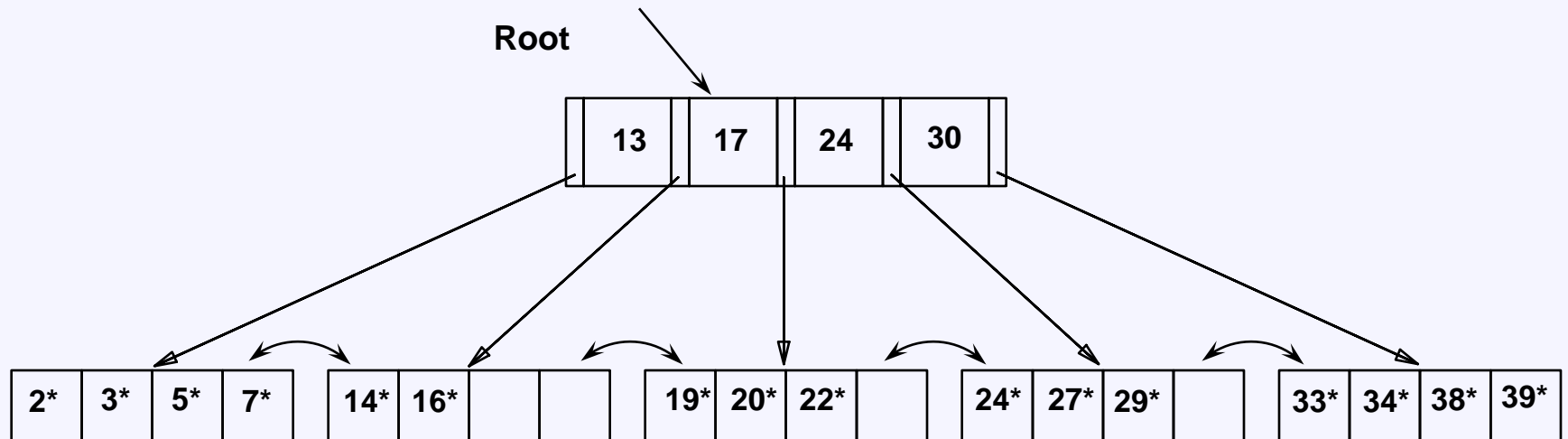
# B<sup>+</sup>-δένδρα - Εισαγωγή

- Εισαγωγή του κλειδιού 142

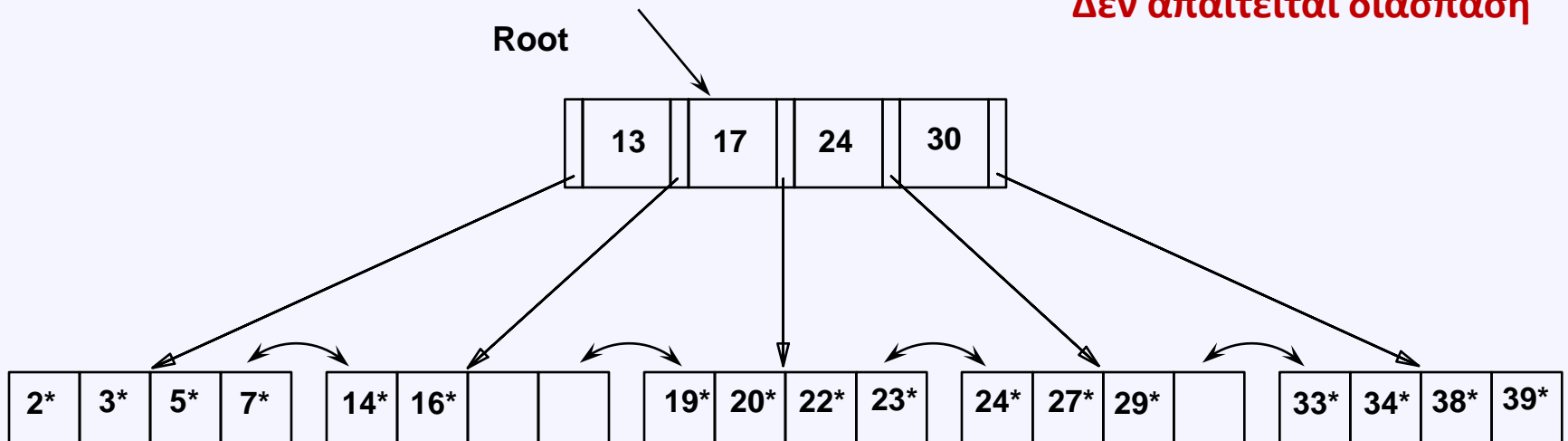


Σχήμα 12.14: Εισαγωγή σε B<sup>+</sup>-δένδρο με  $d=2$  και  $Bkfr=4$ .

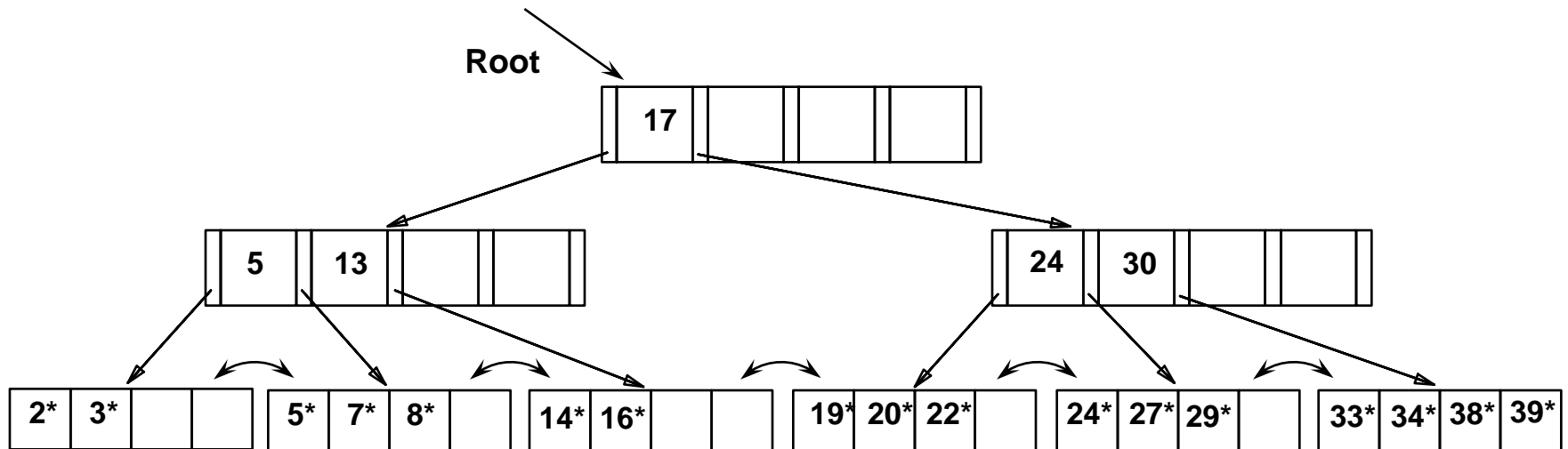
# Εισαγωγή 23\*



**Δεν απαιτείται διάσπαση**



# Εισαγωγή 8\*

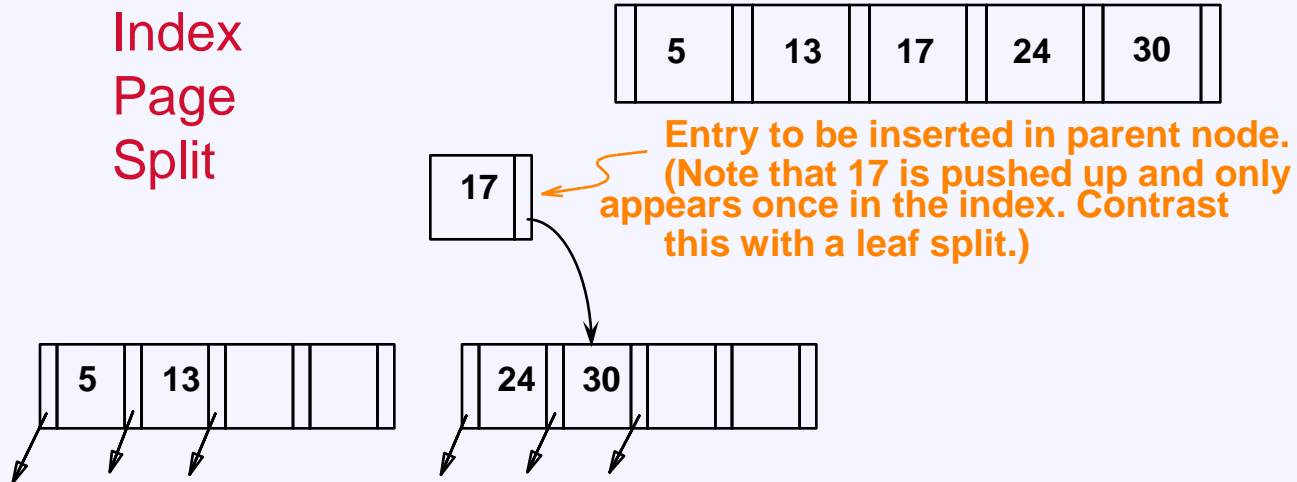
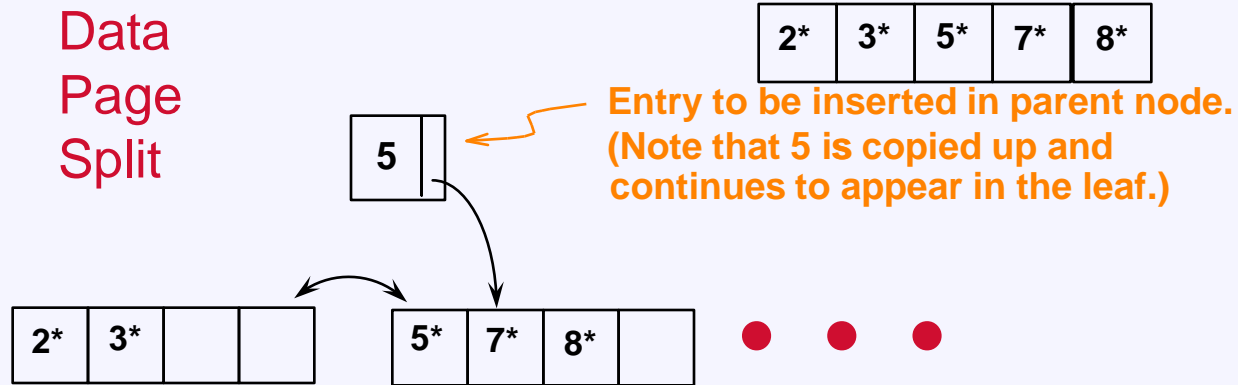


**Η ρίζα διασπάται και το δένδρο αυξάνει το ύψος του**



# Data vs. Index Page Split

(from previous example of inserting "8")



# B<sup>+</sup>-δένδρα – Διαγραφή Αλγόριθμος

1. Εκκινήστε από τη ρίζα, βρείτε το φύλλο L όπου ανήκει η εγγραφή.
2. Διαγράψτε την εγγραφή.
  - 2.1 Εάν το φύλλο L είναι τουλάχιστον μισό-γεμάτο, τέλος!
  - 2.2 Εάν το **φύλλο L έχει λιγότερες εγγραφές** από όσες πρέπει,
    - A.** Προσπαθήστε να κάνετε **ανακατανομή, δανειζόμενοι από έναν αδερφικό κόμβο** (με προτεραιότητα εξετάζεται ο αριστερός). Αν ο αδερφικός κόμβος έχει περισσότερες εγγραφές από το όριο τότε **οι εγγραφές μοιράζονται στους δύο κόμβους και ενημερώνεται ο πατέρας κόμβος.**
    - B.** Εάν η ανακατανομή αποτύχει, **συγχώνευσε το L και τον αδερφικό κόμβο.** Επίσης ενημερώνεται ο πατέρας και διαγράφεται ένα κλειδί και ο δείκτης.

# B<sup>+</sup>-δένδρα – Διαγραφή Αλγόριθμος

2.3 Εάν ο κόμβος είναι **εσωτερικός κόμβος** και μένει με λιγότερο από  $d$  κλειδιά,

**A.** Προσπαθήστε να κάνετε **ανακατανομή, δανειζόμενοι από έναν αδερφικό κόμβο** (με προτεραιότητα εξετάζεται ο αριστερός). Αν ο αδερφικός κόμβος έχει περισσότερα κλειδιά από το όριο τότε τα κλειδιά του αρχικού κόμβου, το κλειδί του πατρικού κόμβου τα κλειδιά του αδερφικού κόμβου αναδιανέμονται στους δύο αδερφικούς κόμβους. Το *μεσαίο κλειδί* από το σύνολο των κλειδιών ανεβαίνει στον πατρικό κόμβο.

**B.** Εάν η ανακατανομή αποτύχει, συγχώνευσε το κόμβο και τον αδερφικό κόμβο. Επίσης ενημερώνεται ο πατέρας και διαγράφεται ένα κλειδί και ο δείκτης.

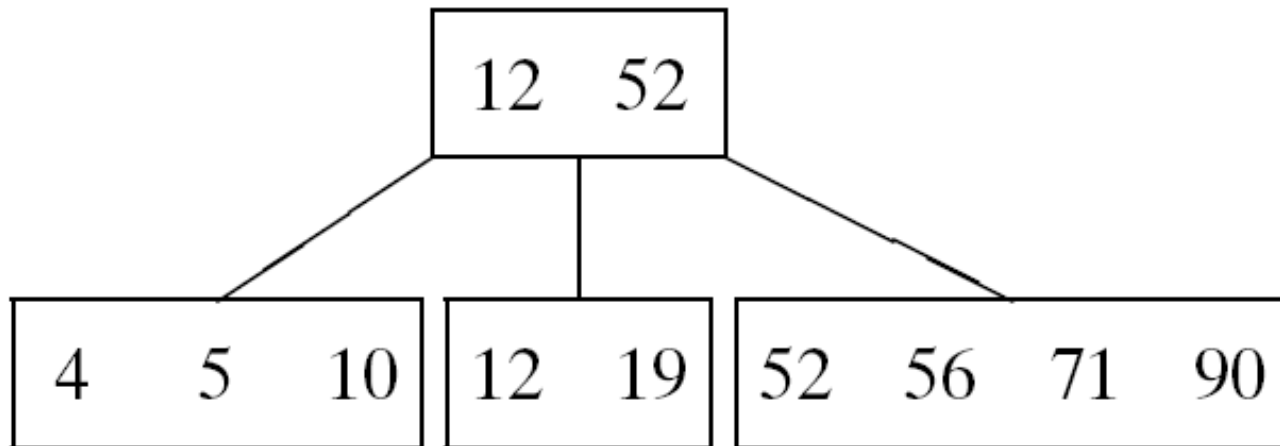
- Η συγχώνευση μπορεί να οδηγήσει μέχρι τη ρίζα και μειωθεί το ύψος.

# $B^+$ -δένδρα – Διαγραφή

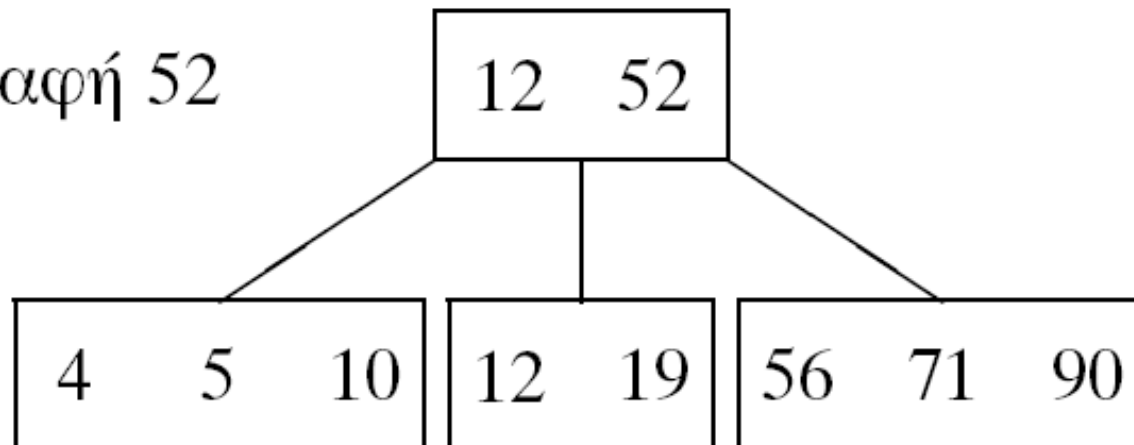
- *Η περιεκτικότητα των κόμβων δεν πρέπει να είναι κατώτερη του 50%.*
- Αν από τους τερματικούς κόμβους διαγραφεί κάποια εγγραφή με κλειδί που είναι αποθηκευμένο σε κάποιο ανώτερο επίπεδο, δεν δημιουργείται πρόβλημα ή αν ο εξωτερικός κόμβος έχει  $> d$  κλειδιά. (Σχήμα β και δ)
- Αν εξωτερικός κόμβος απομείνει με λιγότερες από  $d$  εγγραφές, τότε ελέγχεται αν οι γειτονικοί (ο αριστερός και μετά ο δεξιός) έχουν περισσότερες εγγραφές από αυτό το όριο.
  - Αν υπάρχει, οι εγγραφές μοιράζονται εξίσου μεταξύ τους και ενημερώνεται ο κόμβος πατέρας. (Σχήμα γ)

# B<sup>+</sup>-δένδρα - Διαγραφή

(α)

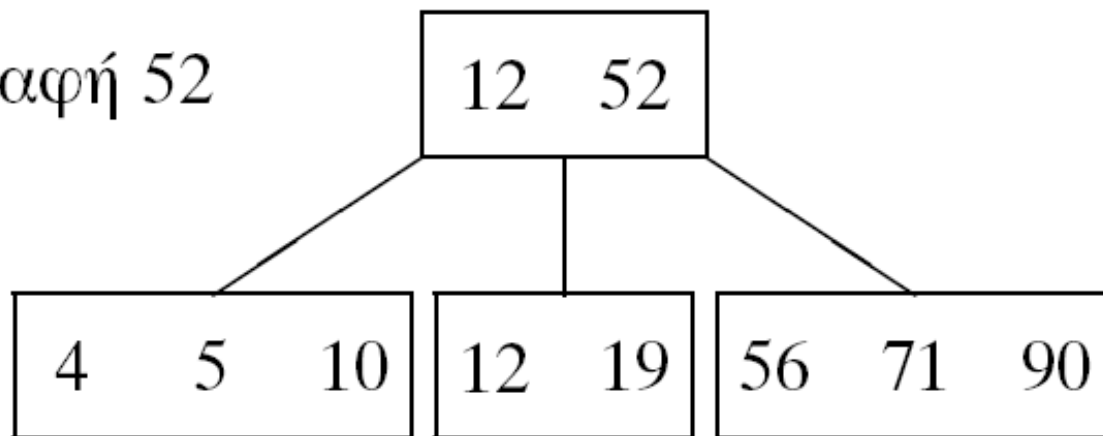


(β) διαγραφή 52

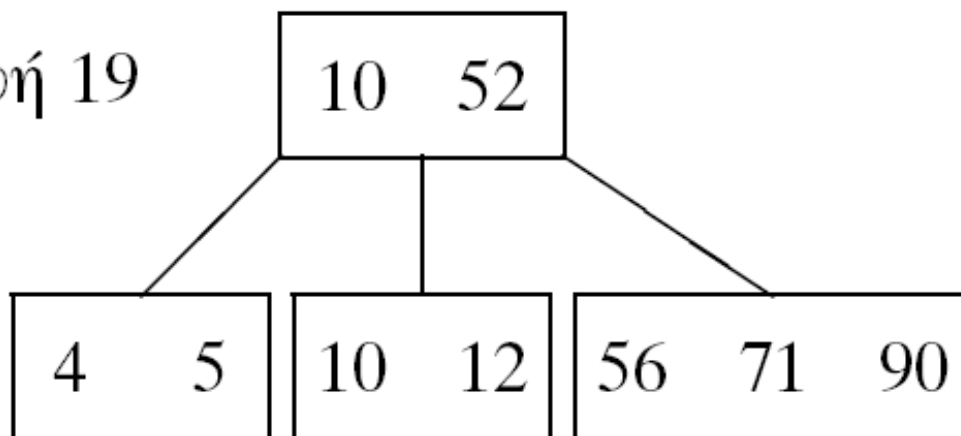


# B<sup>+</sup>-δένδρα - Διαγραφή

(β) διαγραφή 52

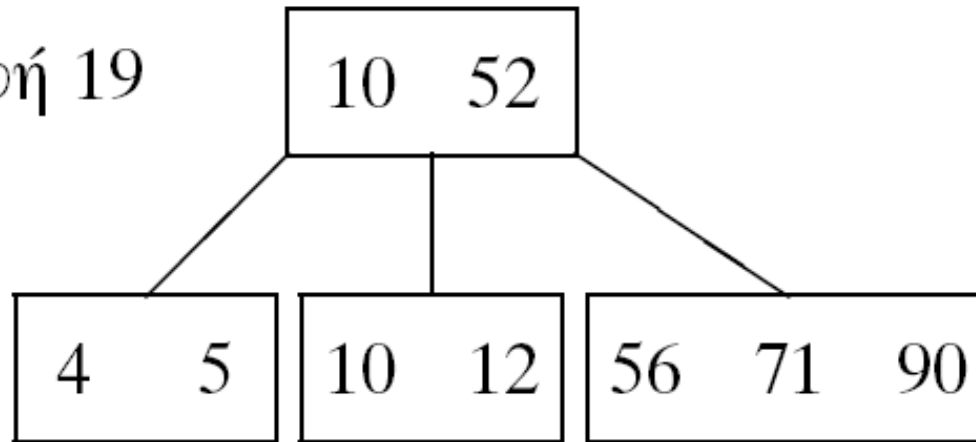


(γ) διαγραφή 19

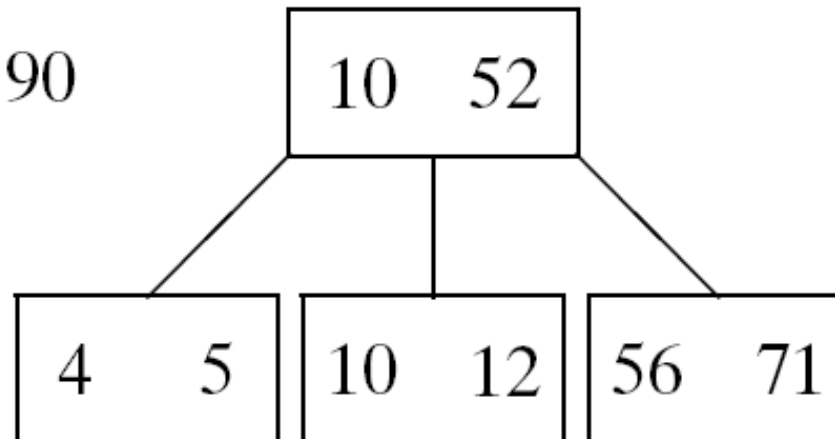


# B<sup>+</sup>-δένδρα - Διαγραφή

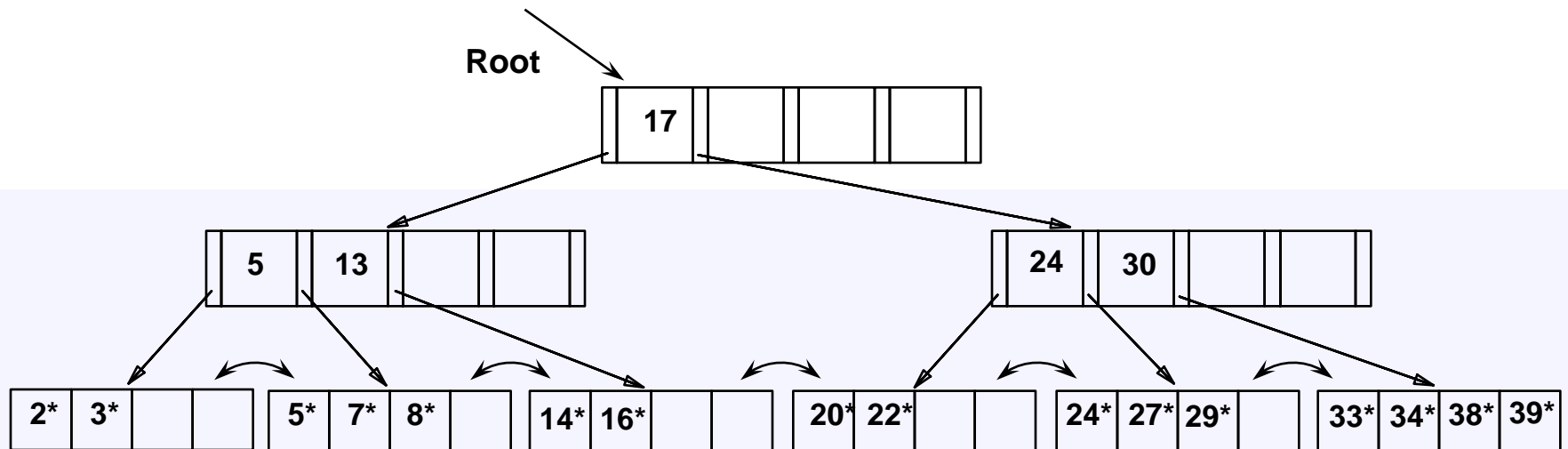
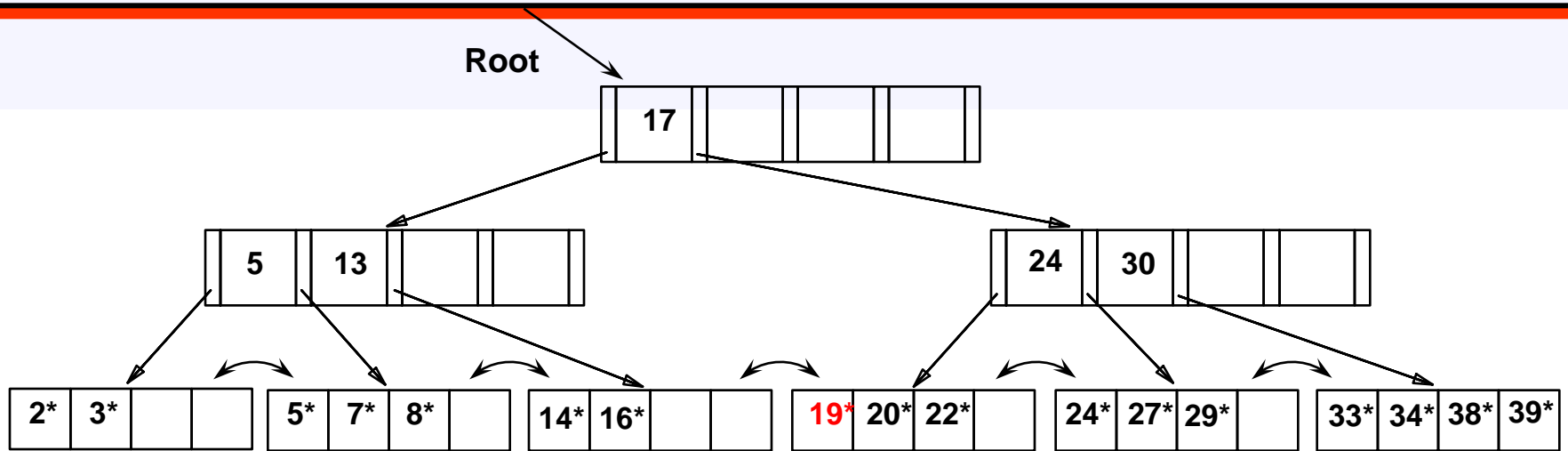
(γ) διαγραφή 19



(δ) διαγραφή 90

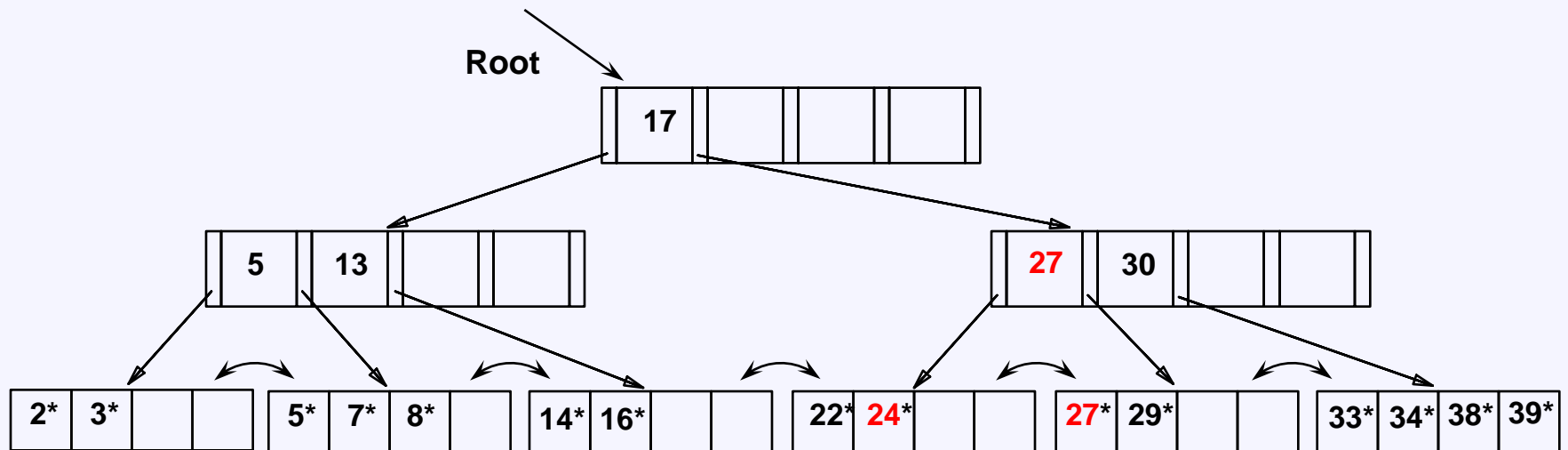
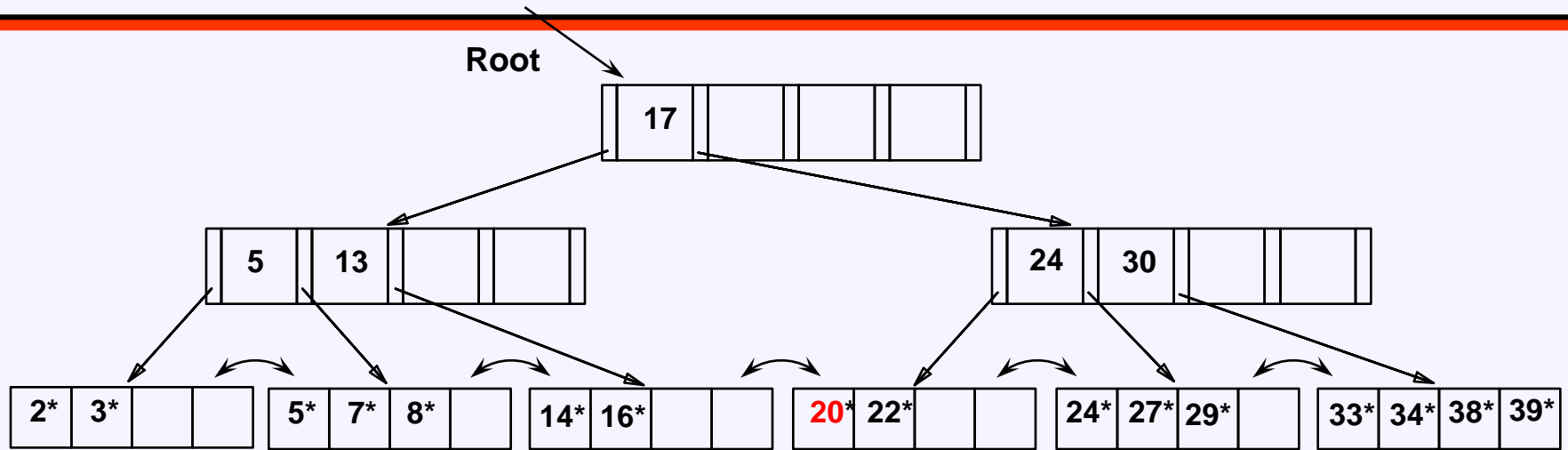


# Διαγραφή 19\*





# Διαγραφή 20\* ...



# Διαγραφή 19\* και 20\*

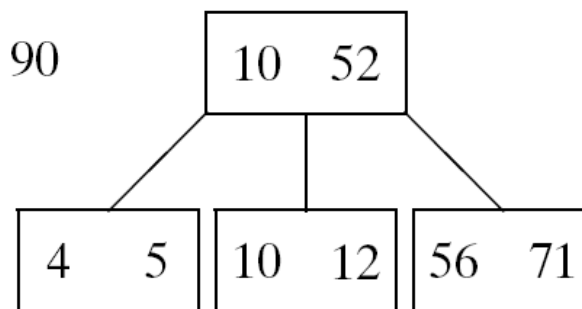
---

- Η διαγραφή του 19\* είναι απλή.
- Η διαγραφή του 20\* γίνεται με αναδιανομή.  
Παρατηρείστε πως **η τιμή του μικρότερου κλειδιού του δεξιού κόμβου αντιγράφεται πάνω.**

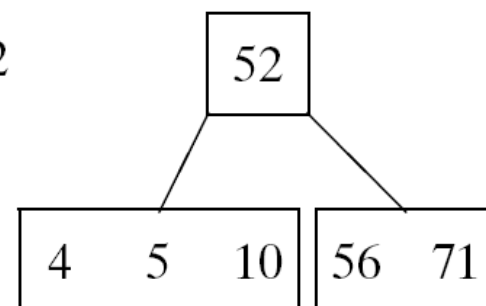
# B<sup>+</sup>-δένδρα - Διαγραφή

- Όταν οι γειτονικοί κόμβοι δεν έχουν περισσότερες από  $d$  ( $Bkfr/2$ ) εγγραφές γίνεται **συγχώνευση**
  - οι εγγραφές επανα-αποθηκεύονται στον ένα κόμβο, ενώ ο άλλος αποδίδεται στο σύστημα,
  - ενημερώνεται ο κόμβος πατέρας, όπου **διαγράφεται ένα κλειδί και ο αντίστοιχος δείκτης.**

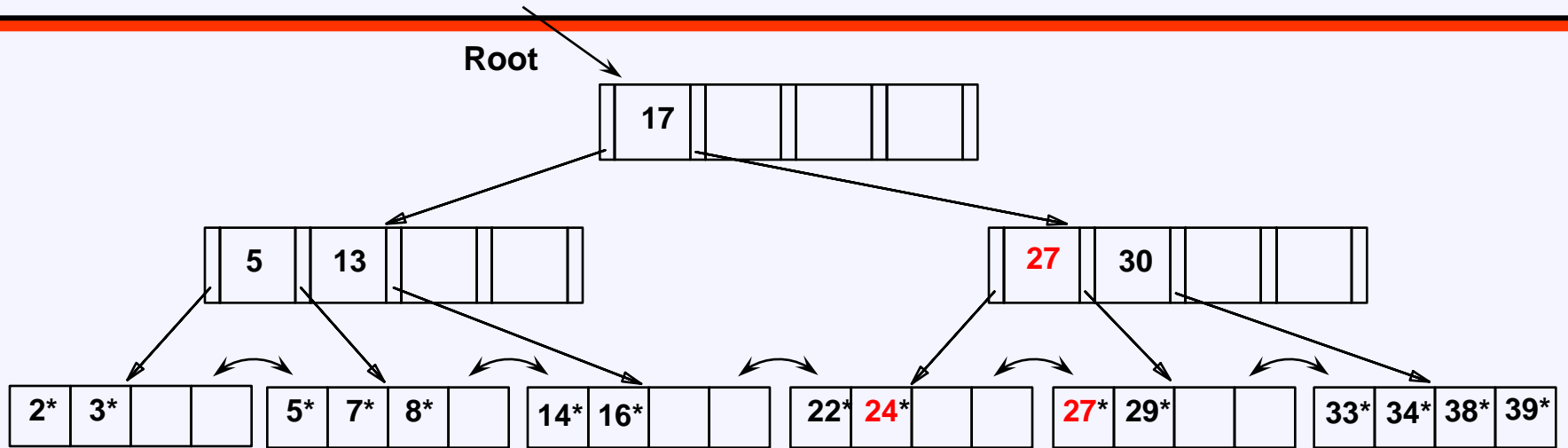
(δ) διαγραφή 90



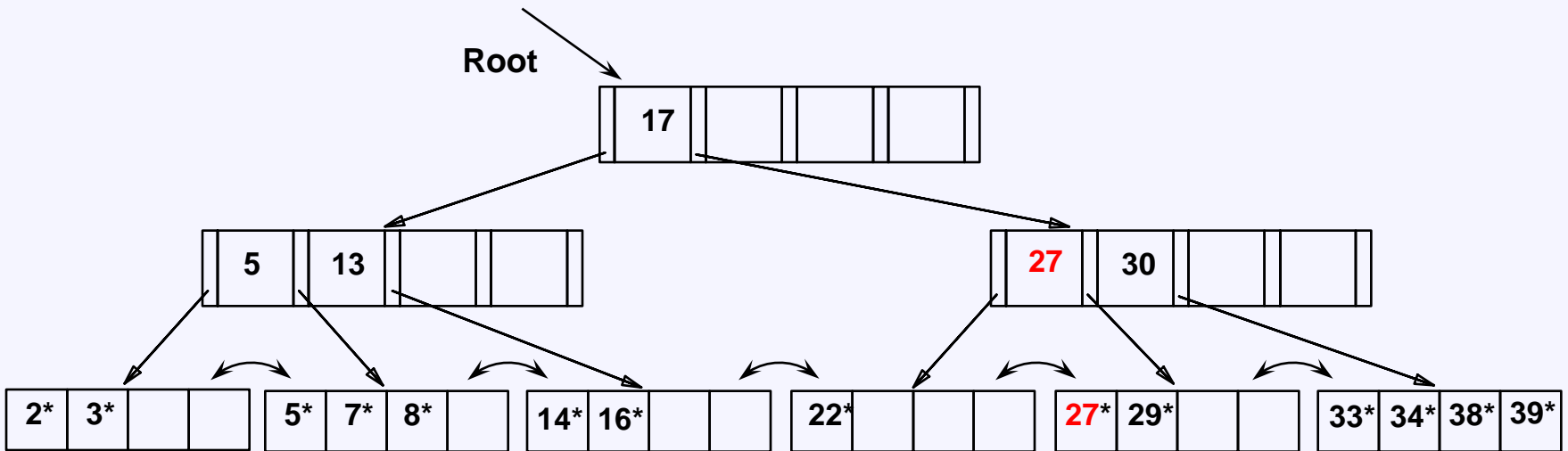
(ε) διαγραφή 12



# Διαγραφή 24\* ...

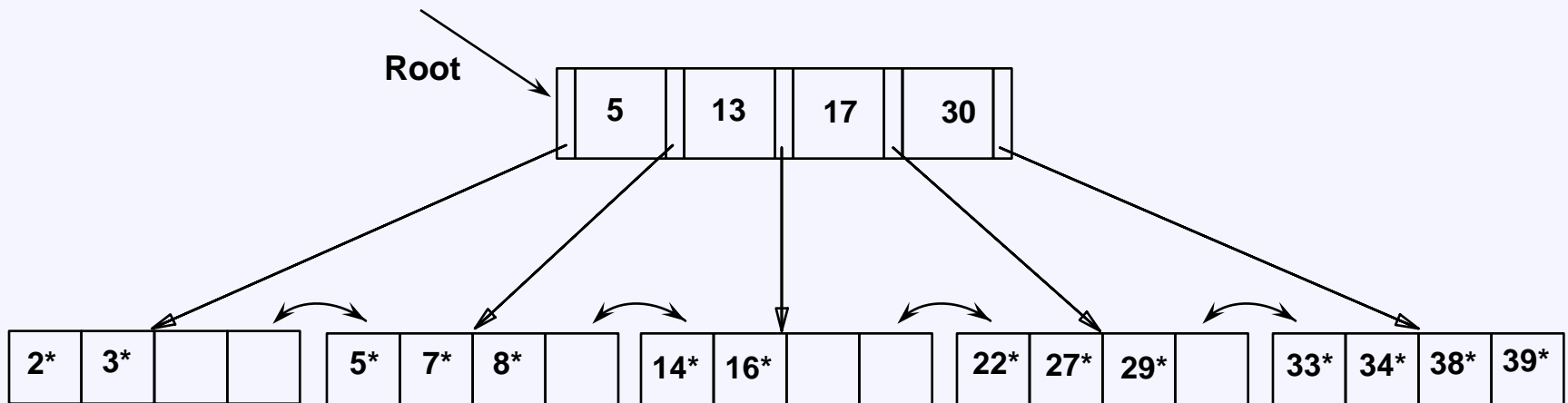
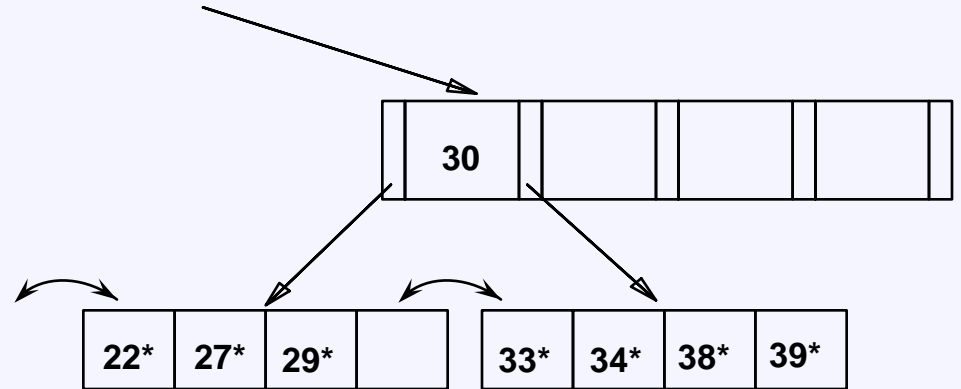


Δεν είναι πιθανή η αναδιανομή



# Διαγραφή 24\*

Πρέπει να γίνει συγχώνευση



# B<sup>+</sup>-δένδρα - Διαγραφή

---

- Έστω **εσωτερικός κόμβος** μένει με  $< d$  κλειδιά  
(Περίπτωση **Αναδιανομής**)
  1. ελέγχονται οι γειτονικοί κόμβοι αν περιέχουν περισσότερα κλειδιά από το όριο,
  2. αν υπάρχει τέτοιος κόμβος, τα κλειδιά του γειτονικού κόμβου, το κλειδί του πατέρα και τα κλειδιά του κόμβου αναδιανέμονται στους δύο κόμβους,
  3. το μεσαίο κλειδί ανέρχεται στον κόμβο πατέρα.

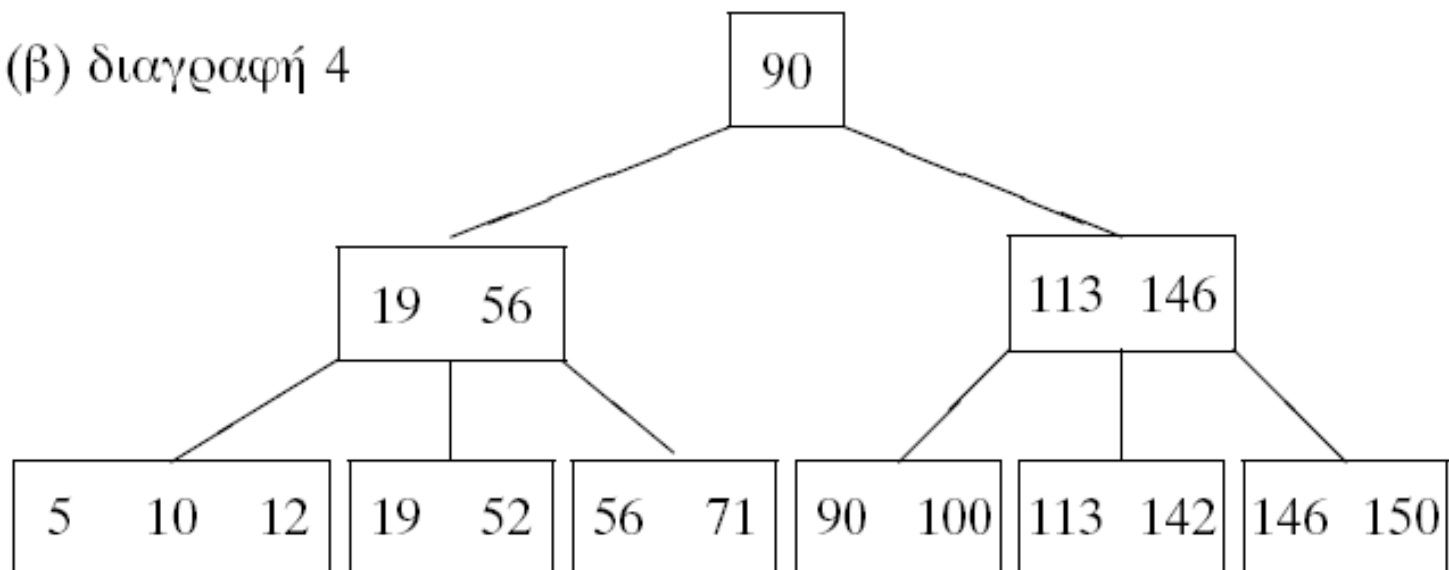
# B<sup>+</sup>-δένδρα – Διαγραφή

---

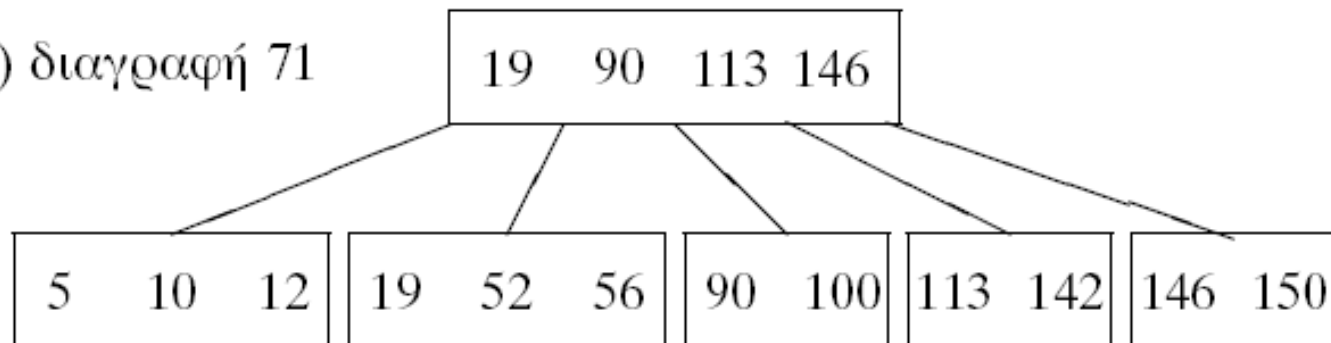
- Έστω **εσωτερικός κόμβος** μένει με  $< d$  κλειδιά και οι δύο γειτονικοί κόμβοι έχουν  $d$  κλειδιά (Περίπτωση **Συγχώνευσης**)
  1. τα κλειδιά επαναποθηκεύονται στον ένα κόμβο, ενώ ο άλλος αποδίδεται στο σύστημα,
  2. το αντίστοιχο κλειδί και ο σχετικός δείκτης διαγράφονται από τον κόμβο πατέρα,
  3. αν και ο πατέρας κόμβος μένει με λιγότερο από  $d$  κλειδιά, τότε το πρόβλημα μεταφέρεται ένα επίπεδο προς τα επάνω.

# B<sup>+</sup>-δένδρα - Διαγραφή

(β) διαγραφή 4



(γ) διαγραφή 71





# Παραλλαγές B-δένδρων

---

- Μεταξύ των άλλων στη βιβλιογραφία αναφέρονται οι εξής παραλλαγές:
  - **B-δένδρα με κλειδιά μεταβλητού μήκους** (B-trees with variable length entries),
  - **B-δένδρα μικρής τάξης,**
  - **Δυαδικά B-δένδρα** (Binary B-trees),
  - **Συμμετρικά Δυαδικά B-δένδρα** (Symmetric Binary B-trees), και
  - **Προθεματικά B<sup>+</sup>-δένδρα** (Prefix B<sup>+</sup>-trees).

# B-δένδρα με κλειδιά μεταβλητού μήκους

- Είναι δυνατόν να κατασκευασθεί ένα B-δένδρο για κλειδιά μεταβλητού μήκους (συμβολοσειρές) με περιεκτικότητα κόμβων τουλάχιστον 50%.
- Στο δένδρο αυτό **δεν έχει σημασία ο αριθμός των κλειδιών που περιέχει, αλλά το συνολικό μέγεθος των κλειδιών.**
- Έχουν προταθεί αρκετοί αλγόριθμοι με σκοπό τη βελτιστοποίηση του δένδρου αυτού (ως προς το ύψος και το πλήθος του συνολικού αριθμού κόμβων) και συνεπώς την ελαχιστοποίηση του χρόνου αναζήτησης.
- Για το σκοπό αυτό, **θα πρέπει τα μικρού μεγέθους κλειδιά να αποθηκευθούν προς τη ρίζα.**
  - *Η δομή αυτή εφαρμόζεται μόνο σε στατικά και γνωστά από την αρχή δεδομένα.*

# B-δένδρα μικρής τάξης

---

- Υπάρχουν πολλές παραλλαγές:
  - **2-3 δένδρα**, δηλαδή B-δένδρα τάξης  $m=3$  που περιέχουν 1 έως 2 εγγραφές
  - **1-2 δένδρα**, δηλαδή B-δένδρα τάξης  $m=2$  που περιέχουν 0 έως 1 εγγραφές
  - **2-3 δένδρα αδελφών** (2-3 brother trees), με τον περιορισμό ότι ένας κόμβος με 1 εγγραφή πρέπει να έχει οπωσδήποτε αδελφό κόμβο με 2 εγγραφές

# B-δένδρα μικρής τάξης

---

- Υπάρχουν πολλές παραλλαγές:
  - **2-3 δένδρα υιών** (2-3 son trees) με τον περιορισμό ότι **δεν** μπορεί να υπάρξει ένα ζεύγος κόμβων με σχέση πατέρα-παιδιού, όπου και οι δύο κόμβοι να έχουν μια μόνο εγγραφή,
  - **1-2 δένδρα υιών** με παρόμοιο περιορισμό.

# B-δένδρα μικρής τάξης

---

- Είναι ευκολονόητο ότι οι περιορισμοί αυτοί καθιστούν δύσκολο καθήκον την ανάπτυξη των σχετικών αλγορίθμων διαχείρισης των δομών.
- Όλες αυτές οι παραλλαγές αποτελούν περισσότερο θεωρητικές αναζητήσεις παρά οργανώσεις με πρακτική εφαρμογή σε ΣΔΒΔ.

# Προθεματικά B<sup>+</sup>-δένδρα

---

- Τα Προθεματικά B<sup>+</sup>-δένδρα έχουν παρόμοια δομή με τα B<sup>+</sup>-δένδρα και είναι μία άλλη παραλλαγή για υλοποίηση κυρίως σε δευτερεύουσα μνήμη.
- Βασική Ιδέα: Αν το κλειδί είναι συμβολοσειρά, τότε είναι προτιμότερο **στους εσωτερικούς κόμβους** του καταλόγου να **αποθηκεύεται μόνο εκείνο το πρόθεμα του κλειδιού**, το οποίο είναι **απαραίτητο για τη διαδικασία αναζήτησης**.

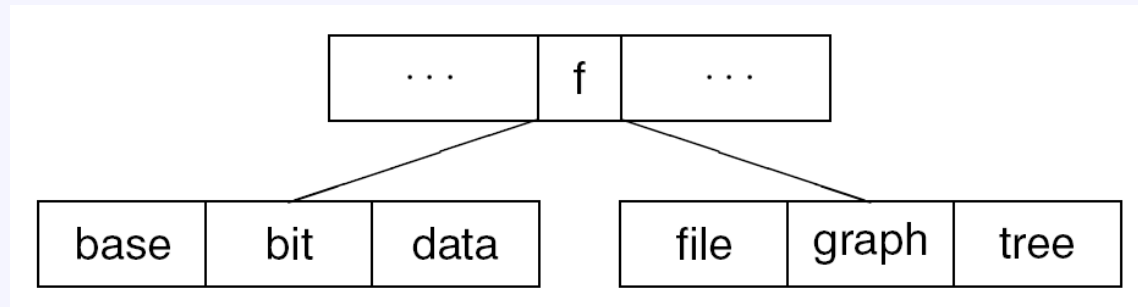
# Προθεματικά B+-δένδρα

---

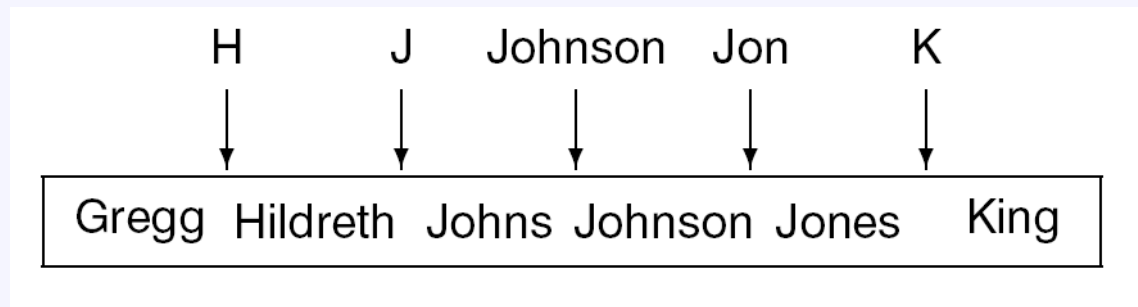
- Έτσι ο κατάλογος περιέχει τους λεγόμενους **διαχωριστές** (separators) που είναι **μεταβλητού μήκους**.
- Το μήκος των διαχωριστών ποικίλει από 1 byte μέχρι ένα πλήρες κλειδί και εξαρτάται από τις τιμές των γειτονικών κλειδιών των εγγραφών.

# Προθεματικά B+-δένδρα

- Παράδειγμα διαχωριστή.



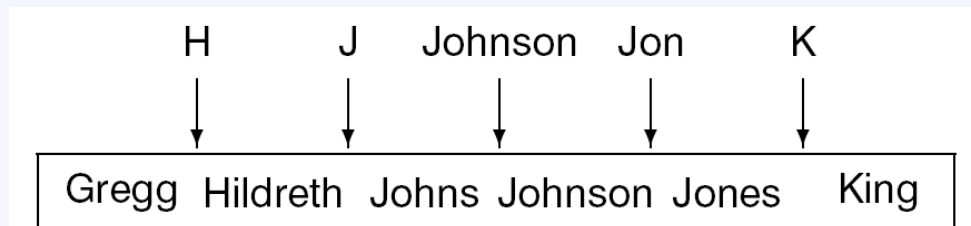
- Πλήρης κόμβος με 6 κλειδιά και 5 διαχωριστές μεταξύ των κλειδιών.





# Προθεματικά B+-δένδρα

- Στον κόμβο του παραδείγματος εισάγεται το κλειδί JUNG, οπότε ο κόμβος υπερχειλίζει και διασπάται.
- Αν τα κλειδιά αναδιανεμηθούν εξίσου, τότε το κλειδί JOHNSON πρέπει να αποθηκευθεί σε ένα νέο δεξιό κόμβο και ο διαχωριστής JOHNSON να ανέλθει κατά ένα επίπεδο.
- Αν τα κλειδιά δεν διανεμηθούν εξίσου στους δύο κόμβους, αλλά στον αριστερό κόμβο αποθηκευθούν τα GREGG, HILDRETH και στο δεξιό τα JOHNS, JOHNSON κλπ., τότε ο διαχωριστής θα είναι το J.



# Προθεματικά B+-δένδρα

- Η αναδιανομή των κλειδιών σε περίπτωση διάσπασης είναι θέμα βελτιστοποίησης.
- Ως **διάστημα διάσπασης** των τερματικών κόμβων,  $\sigma_{ex}$ , και των εσωτερικών κόμβων,  $\sigma_{in}$ , **ορίζεται ο αριθμός των bytes ή των κλειδιών προς οποιαδήποτε πλευρά του μέσου του κόμβου** που θα μπορούσε να θεωρηθεί ως σημείο διάσπασης.
- ***Το σημείο διάσπασης επιλέγεται εντός του διαστήματος διάσπασης, ώστε να ελαχιστοποιηθεί ο σχετικός διαχωριστής.***