Δεντρικά Ευρετήρια

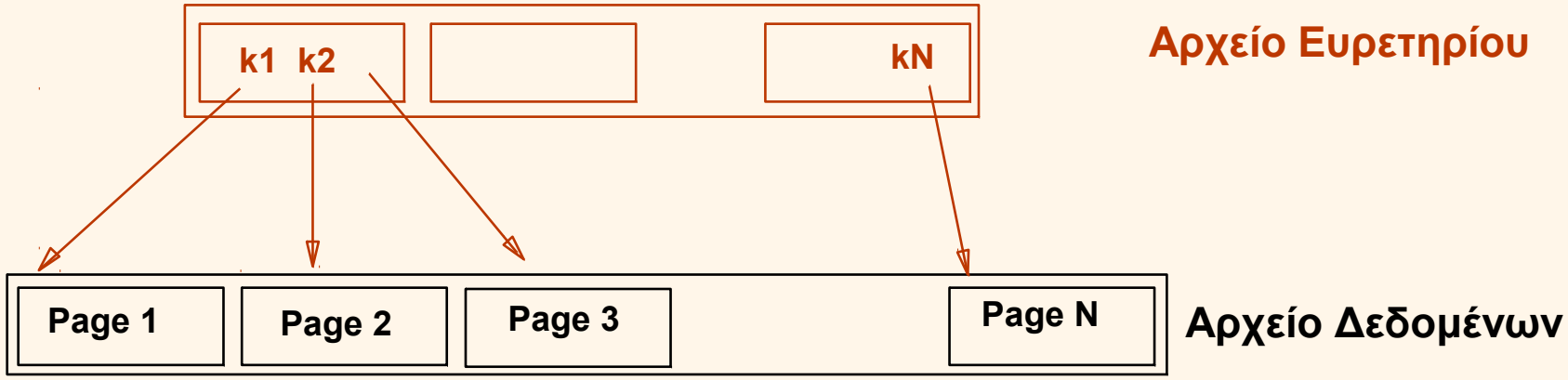
Κεφάλαιο 10

Εισαγωγή

- *Όπως σε κάθε ευρετήριο, υπάρχουν 3 εναλλακτικές για τις καταχωρίσεις δεδομένων* **k***:
  (1) Εγγραφή δεδομένων με τιμή κλειδιού αναζήτησης **k**
  (2) <**k**, rid εγγραφής με τιμή κλειδιού αναζήτησης k>
  (3) <k, λίστα από rids εγγραφών με τιμή κλειδιού αναζήτησης **k**>
- Η επιλογή εναλλακτικής είναι ανεξάρτητη από τη *τεχνική ευρετηριοποίησης* που χρησιμοποιείται για τον εντοπισμό των καταχωρίσεων δεδομένων **k***.
- Οι τεχνικές δεντρικής ευρετηριοποίησης υποστηρίζουν *αναζητήσεις με διάστημα* και *αναζητήσεις με ισότητα*.
- *ISAM*: στατική δομή, *B+tree*: δυναμική, αναδιοργανώνεται αυτόματα κατά τις εισαγωγές και τις διαγραφές.

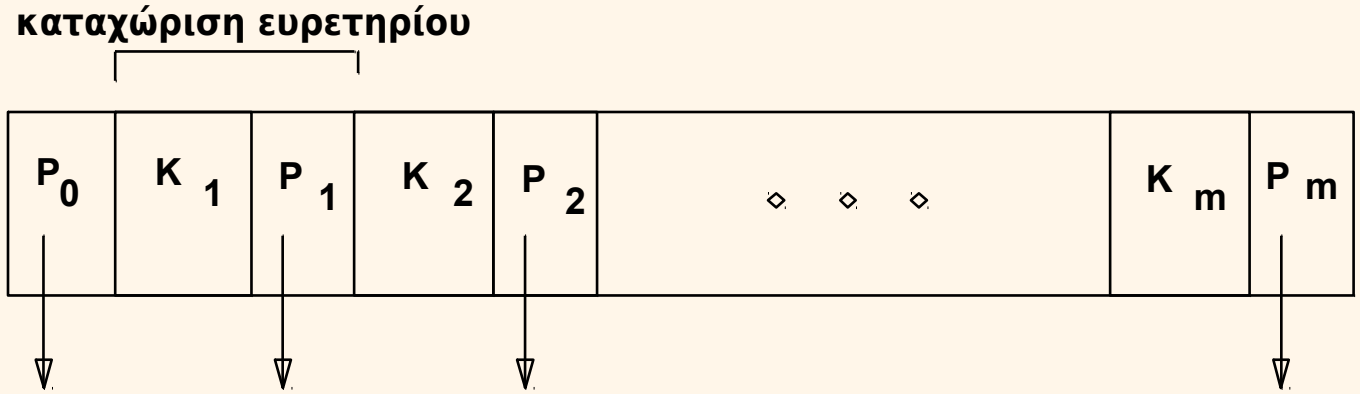
Αναζήτηση με διάστημα

- *'Βρες όλους τους φοιτητές με βαθμό>8'*
  - Αν τα δεδομένα είναι σε ταξινομημένο αρχείο, κάνε δυαδική αναζήτηση για να βρεις τον πρώτο τέτοιο φοιτητή, και μετά σάρωσε το αρχείο για να βρεις τους υπόλοιπους.
  - Το κόστος της δυαδικής αναζήτησης μπορεί να είναι αρκετά υψηλό.
- Απλή ιδέα: Δημιούργησε ένα αρχείο 'ευρετηρίου'.

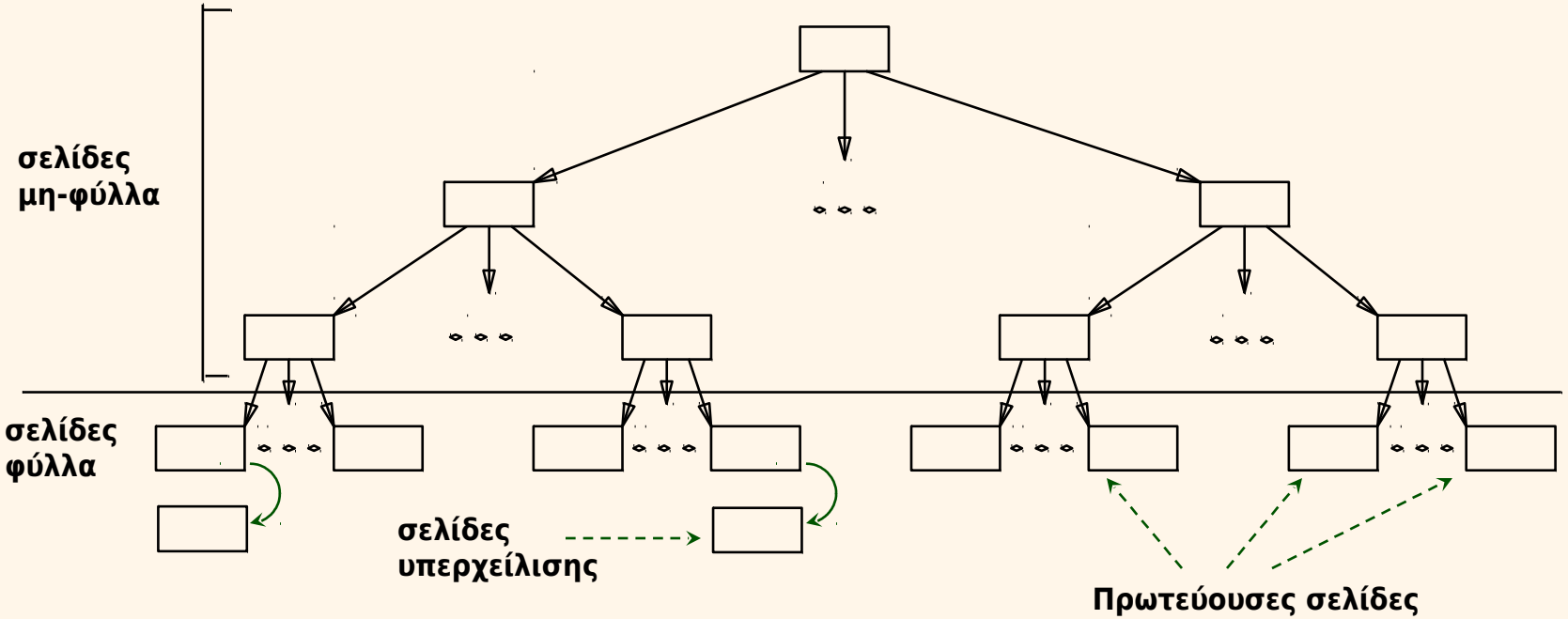


- *Μπορούμε να κάνουμε δυαδική αναζήτηση στο (μικρότερο) αρχείο ευρετηρίου!*

ISAM



- Το αρχείο ευρετηρίου μπορεί να είναι πάλι μεγάλο, αλλά η ιδέα μπορεί να επαναληφθεί!



- *Οι σελίδες φύλλα περιέχουν καταχωρίσεις δεδομένων.*

Σχόλια για το ISAM

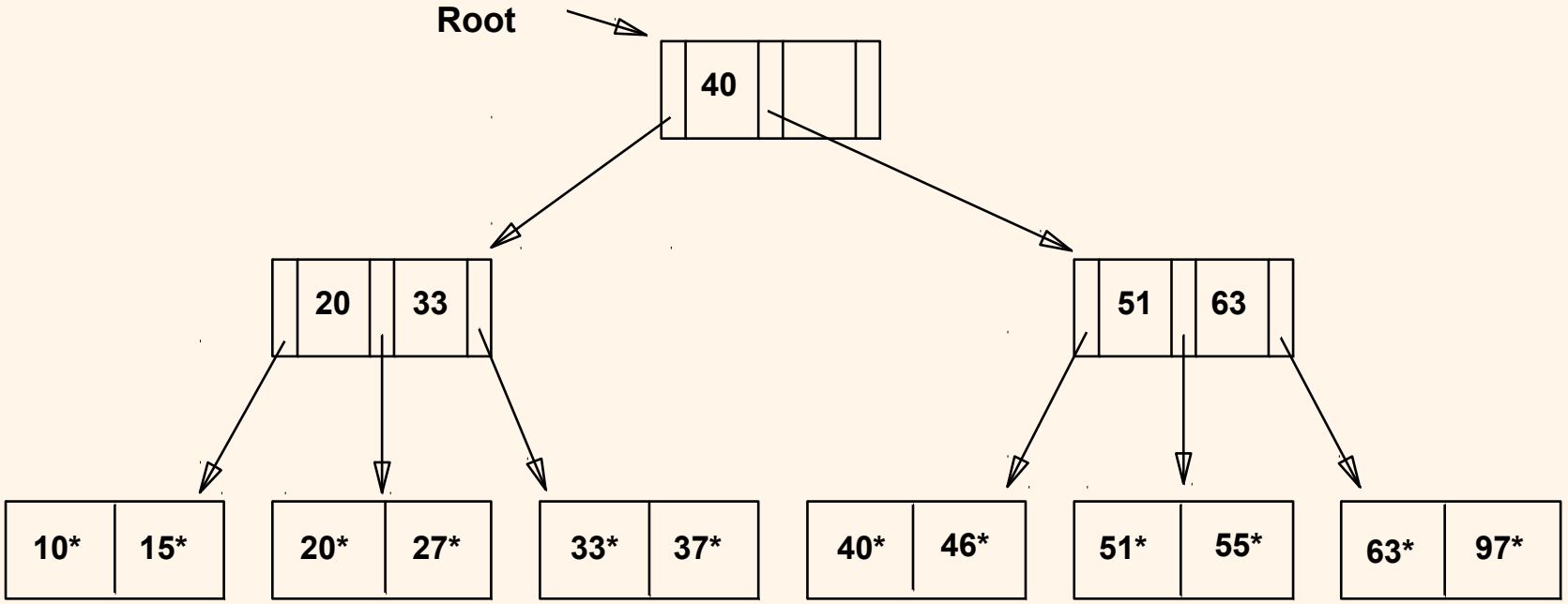
Σελίδες Δεδομένων
Σελίδες Ευρετηρίου
Σελίδες Υπερχείλισης

- *Δημιουργία αρχείου*: Οι σελίδες φύλλα (δεδομένων) δεσμεύονται σειριακά, ταξινομημένες ως προς το κλειδί αναζήτησης – μετά δεσμεύονται οι σελίδες ευρετηρίου και μετά υπάρχει χώρος για τις σελίδες υπερχείλισης.
- *Καταχωρίσεις ευρετηρίου*: <τιμή κλειδιού αναζήτησης, id σελίδας>. Κατευθύνουν την αναζήτηση καταχωρίσεων δεδομένων που βρίσκονται στις σελίδες φύλλα.
- *Αναζήτηση*: Ξεκίνα από τη ρίζα. Κάνε συγκρίσεις τιμών για να φτάσεις σε φύλλο. Κόστος ~ $\log_F N$, όπου
  F = # καταχωρίσεων / σελίδα ευρετηρίου, N = # φύλλων
- *Εισαγωγή*: Βρες το φύλλο όπου ανήκει η καταχώριση δεδομένων και τοποθέτησέ την εκεί.
- *Διαγραφή*: Βρες και διέγραψε καταχώριση δεδομένων από φύλλο. Σε περίπτωση άδειας σελίδας υπερχείλισης, ελευθέρωσε τη σελίδα.

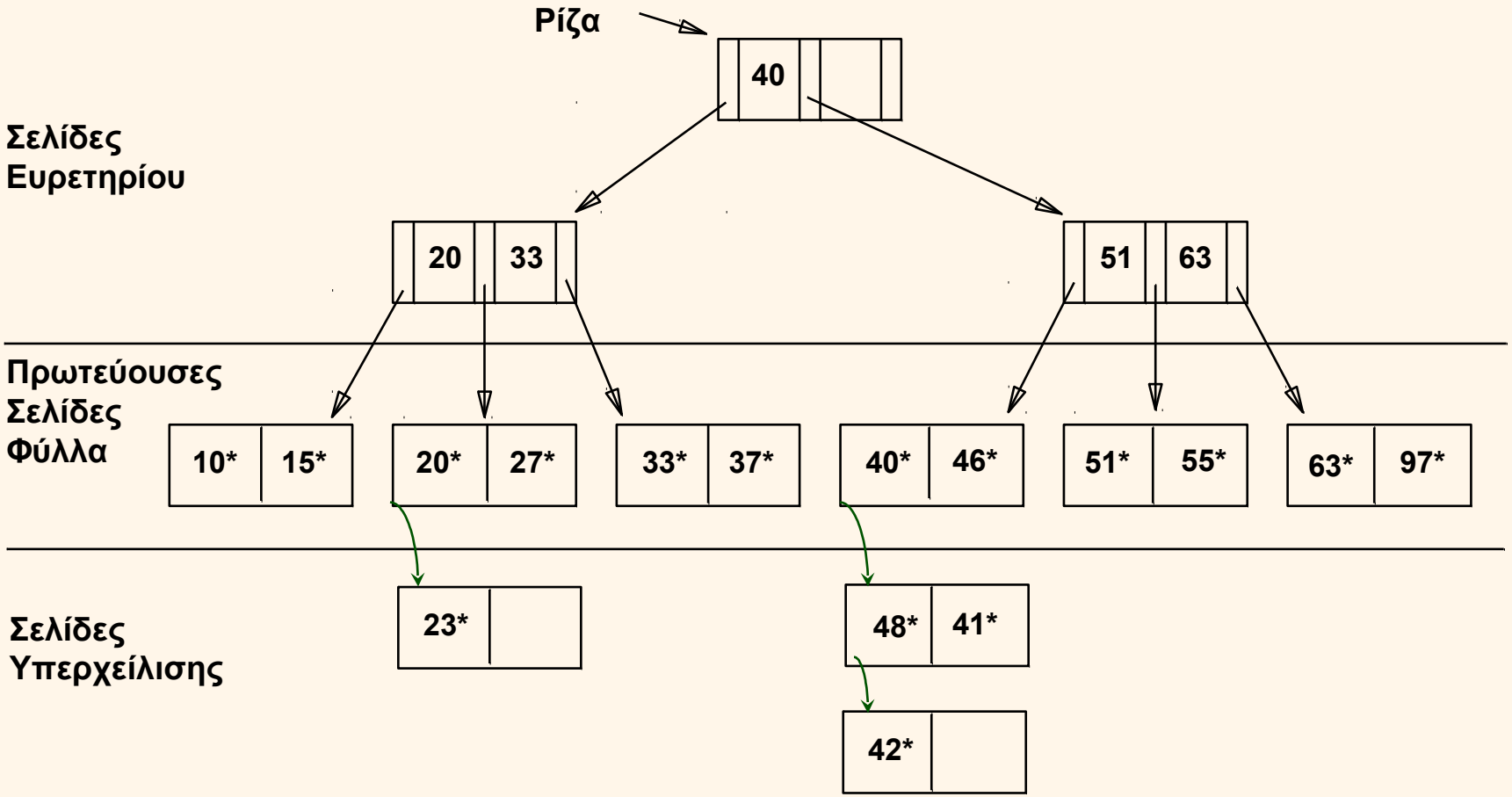
 Στατική δεντρική δομή: εισαγωγές/διαγραφές επηρεάζουν *μόνο τις σελίδες φύλλα*.

Παράδειγμα δέντρου ISAM

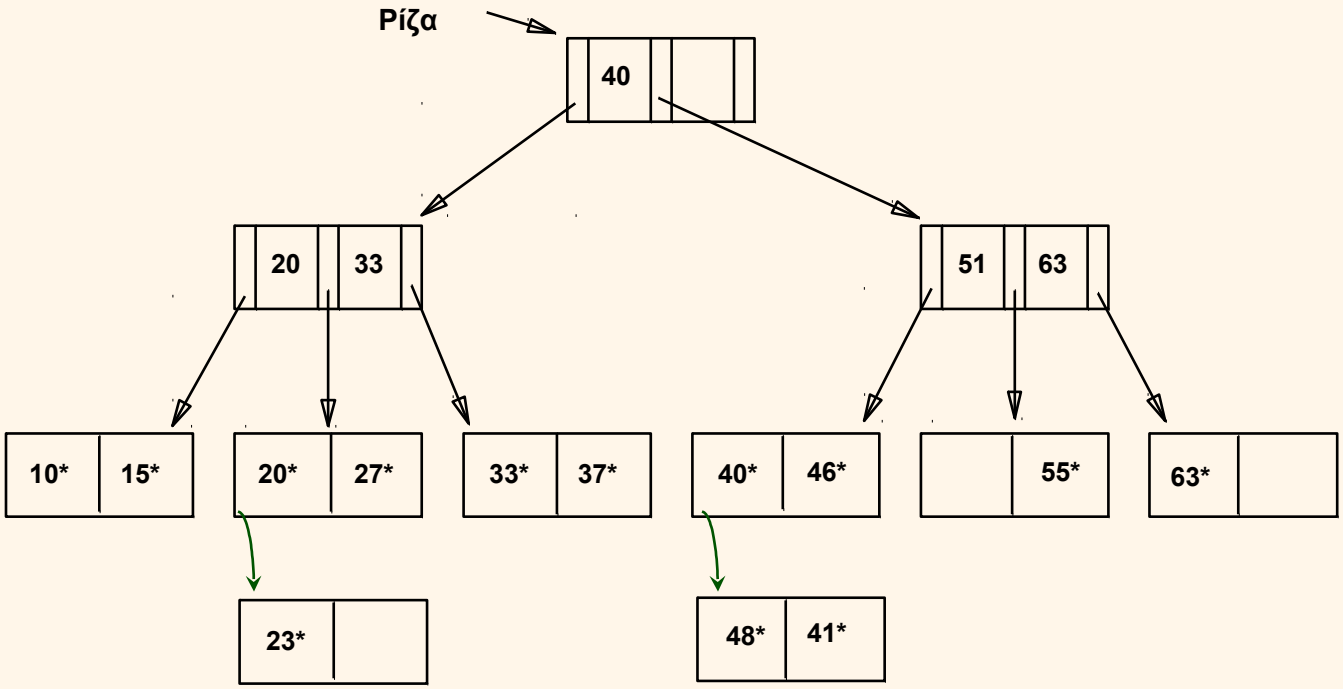
- Κάθε κόμβος χωρά 2 καταχωρίσεις. Δε χρειάζονται 'next-leaf-page' pointers. (Γιατί;)



Αφού εισάγουμε 23, 48*, 41*, 42* ...*



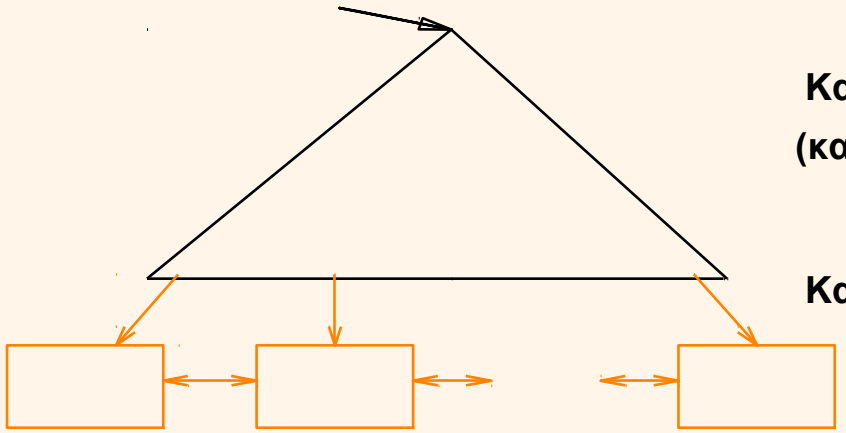
... και μετά διαγράψουμε 42*, 51*, 97*



- *Προσέξτε ότι το 51 εμφανίζεται στα επίπεδα των κόμβων ευρετηρίου, αλλά όχι σε φύλλο!*

B+Tree: το πιο κοινό ευρετήριο

- Εισαγωγή/Διαγραφή με κόστος $\log_F N$, το δένδρο διατηρείται *ισοζυγισμένο*. (F = fanout, N= # σελίδων φύλλων)
- Ελάχιστη πληρότητα 50% (εκτός της ρίζας).
  Κάθε κόμβος περιέχει **d**<=*m*<=2**d** καταχωρίσεις.
  Η παράμετρος **d** είναι η *τάξη* του δένδρου.
- Υποστηρίζει αποτελεσματικές αναζητήσεις με ισότητα και διάστημα.

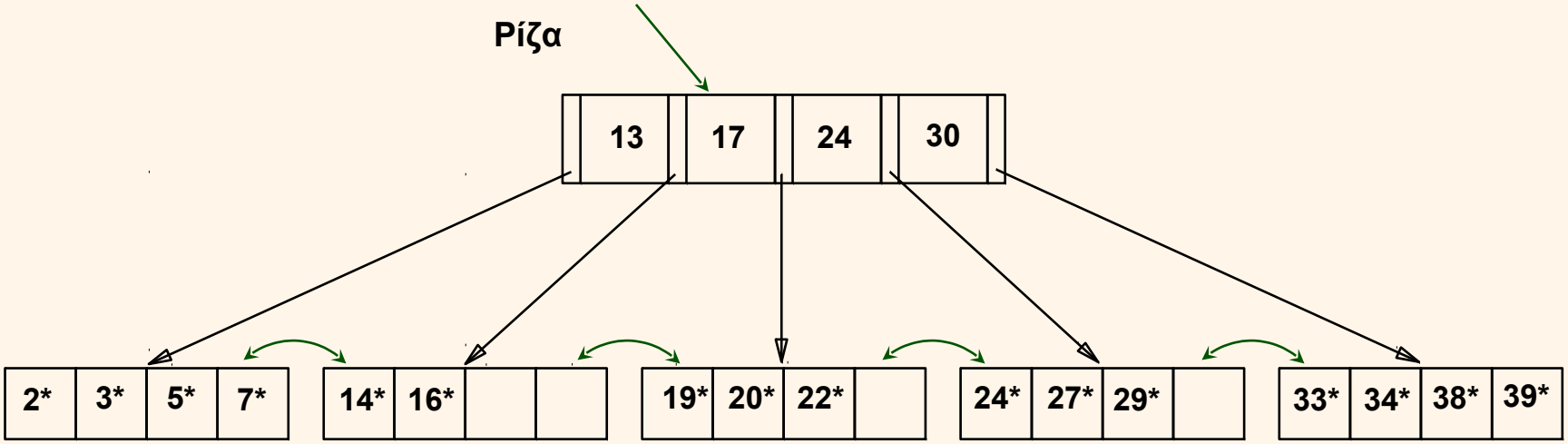


Καταχωρίσεις Ευρετηρίου
(κατευθύνουν την αναζήτηση)

Καταχωρίσεις Δεδομένων
("Ακολουθιακό σύνολο")

Παράδειγμα B+Tree

- Η αναζήτηση ξεκινά από τη ρίζα. Συγκρίσεις τιμών οδηγούν σε φύλλο (όπως στο ISAM).
- Αναζήτηση για 5*, 15*, όλες τις καταχωρίσεις δεδομένων >= 24* ...



- *Σύμφωνα με την αναζήτηση για το 15*, ξέρουμε ότι δεν υπάρχει στο δέντρο!*

B+Trees στην πράξη

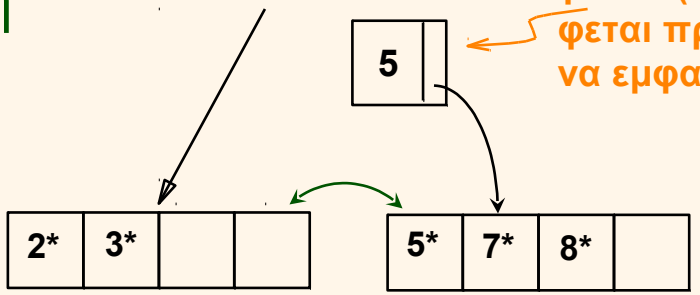
- Τυπική τάξη: 100. Τυπικός συντελεστής πληρότητας: 67%.
  - μέσο fanout = 133
- Τυπικές χωρητικότητες:
  - Ύψος 4: 133^4 = 312,900,700 εγγραφές
  - Ύψος 3: 133^3 = 2,352,637 εγγραφές
- Συχνά τα υψηλότερα επίπεδα μπορούν να βρίσκονται στην ενδιάμεση μνήμη:
  - Επίπεδο 1 = 1 σελίδα = 8 Kbytes
  - Επίπεδο 2 = 133 σελίδες = 1 Mbyte
  - Επίπεδο 3 = 17,689 σελίδες = 133 MBytes

Εισάγοντας μια καταχώριση δεδομένων σε ένα B+Tree

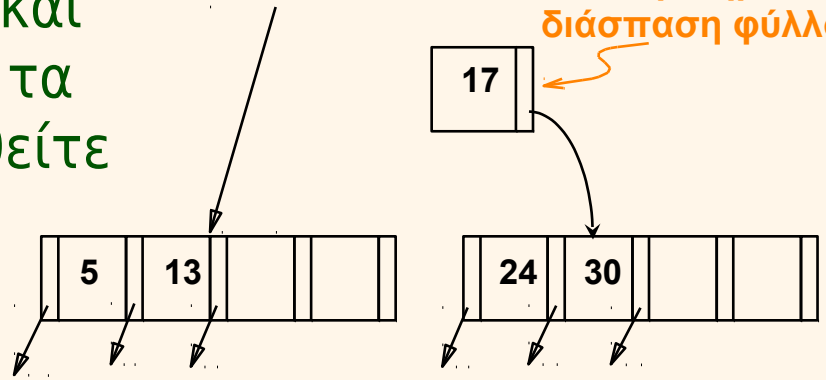
- Βρίσκουμε το σωστό φύλλο *L.*
- Βάζουμε την καταχώριση δεδομένων στο *L*.
  - Αν το *L έχει αρκετό χώρο, τελειώσαμε*!
  - Αλλιώς, πρέπει να *διασπάσουμε* το *L (στο L και σε ένα νέο κόμβο L2)*
    - Μοιράζουμε τις καταχωρίσεις στα δυο, **αντιγράφουμε προς τα επάνω** το μεσαίο κλειδί.
    - Εισάγουμε μια καταχώριση ευρετηρίου που να δείχνει στο *L2* στο γονιό του *L*.
- Αυτό μπορεί να συμβεί αναδρομικά
  - Για να διασπάσουμε κόμβο ευρετηρίου, μοιράζουμε τις καταχωρίσεις στα δυο, αλλά **μεταφέρουμε προς τα επάνω** το μεσαίο κλειδί. (συγκρίνετε τη διαφορά σε σχέση με τις διασπάσεις φύλλων.)
- Οι διασπάσεις “μεγαλώνουν” το δέντρο. Η διάσπαση της ρίζας αυξάνει το ύψος του δέντρου.
  - Μεγάλωμα δέντρου: *φαρδαίνει* ή *ψηλώνει κατά ένα επίπεδο στην κορυφή.*

Εισάγοντας το 8 στο B+Tree*

- Δείτε πώς ικανο-ποιείται η εγγύηση για την ελάχιστη πληρότητα στις διασπάσεις όλων των κόμβων.
- Προσέξτε τη διαφορά ανάμεσα στο *αντιγράφω* και *μεταφέρω* προς τα επάνω. Βεβαιωθείτε ότι καταλαβαί-νετε γιατί συμβαίνει αυτό.

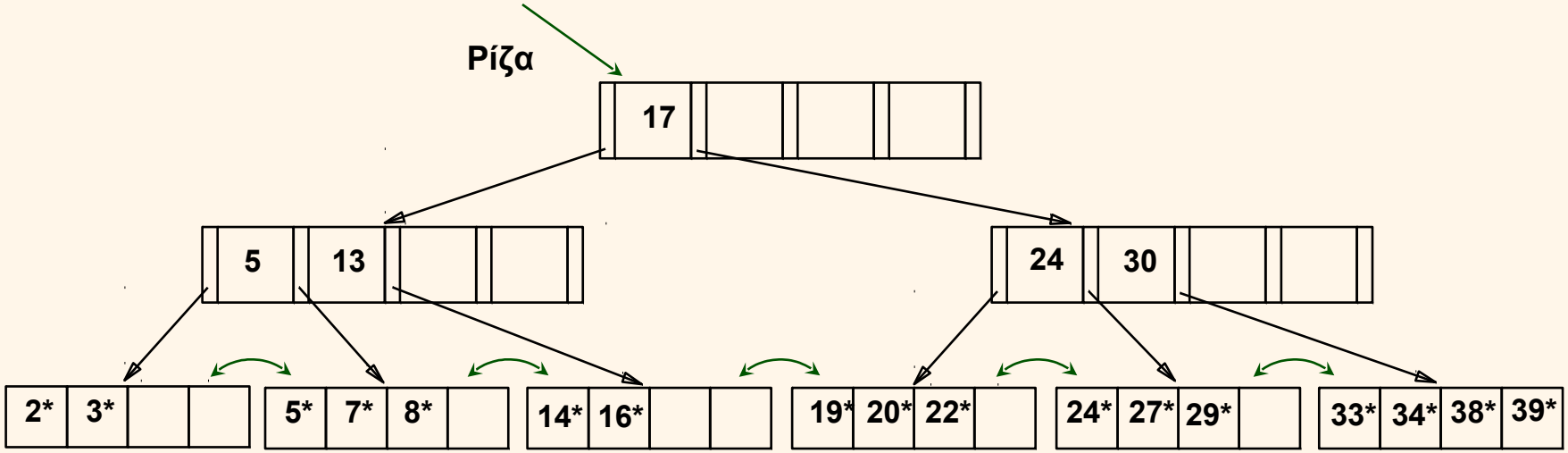


Καταχώριση προς εισαγωγή στο γονιό. (Προσέξετε ότι το 5 αντιγρά-φεται προς τα επάνω και εξακολουθεί να εμφανίζεται στο φύλλο.)



Καταχώριση προς εισαγωγή στο γονιό. (Προσέξτε ότι το 17 μεταφέρεται προς τα επάνω και εμφανίζεται μόνο μια φορά στο ευρετήριο. Συγκρίνετε με τη διάσπαση φύλλου.)

*Το B+Tree μετά την εισαγωγή του 8**

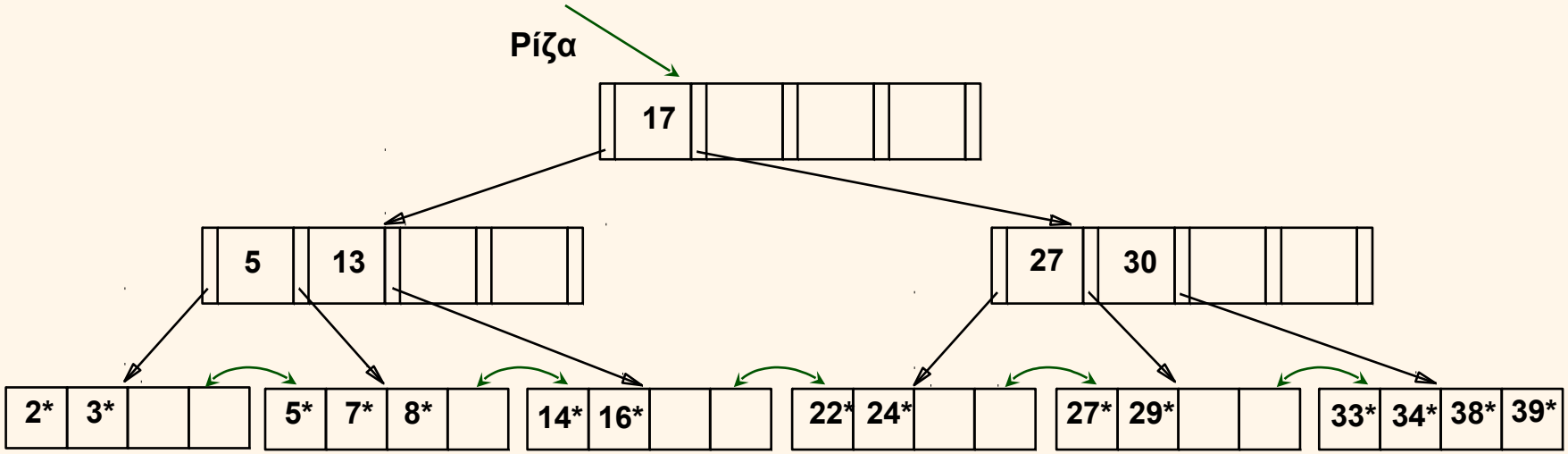


- Προσέξτε ότι έγινε διάσπαση της ρίζας και αυξήθηκε το ύψος του δέντρου.
- Εδώ, μπορούμε να αποφύγουμε τη διάσπαση με ανακατανομή των καταχωρίσεων. Στην πράξη, συνήθως αυτό δε συμβαίνει.

Διαγράφοντας μια καταχώριση δεδομένων από ένα B+Tree

- Ξεκινάμε από τη ρίζα, βρίσκουμε το φύλλο *L* όπου ανήκει η καταχώριση.
- Διαγράφουμε την καταχώριση.
  - Αν το L είναι τουλάχιστον 50% γεμάτο, *τελειώσαμε!*
  - Αν το L έχει μόνο (ακριβώς) **d-1** καταχωρίσεις,
    - Προσπαθούμε να ανακατανείμουμε, δανειζόμενοι από *αδέρφι (διπλανός κόμβος με γονιό ίδιο με το γονιό του L)*.
    - Αν αποτύχουμε, *συγχωνεύουμε* τον *L* με το αδέρφι του.
- Αν γίνει συγχώνευση, πρέπει να διαγράψουμε την καταχώριση (που δείχνει στον *L* ή στο αδέρφι) από το γονιό του *L*.
- Η συγχώνευση μπορεί να φτάσει ως τη ρίζα και να μειωθεί το ύψος του δέντρου.

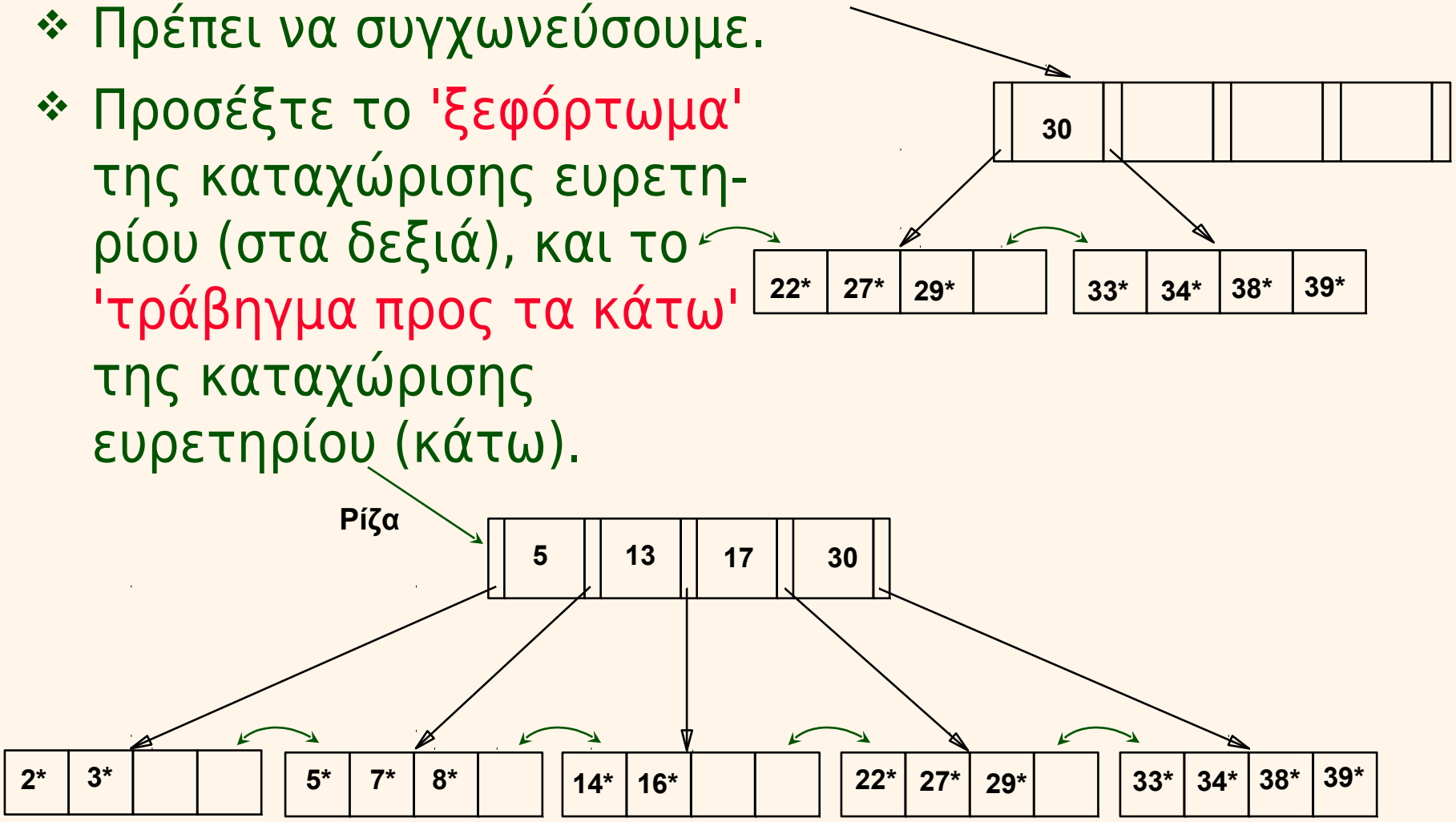
Το δέντρο μετά (την εισαγωγή του 8, και) τη διαγραφή των 19* και 20*...*



- Η διαγραφή του 19* είναι εύκολη.
- Η διαγραφή του 20* γίνεται με ανακατανομή. Προσέξτε πώς το μεσαίο κλειδί *αντιγράφεται προς τα επάνω*.

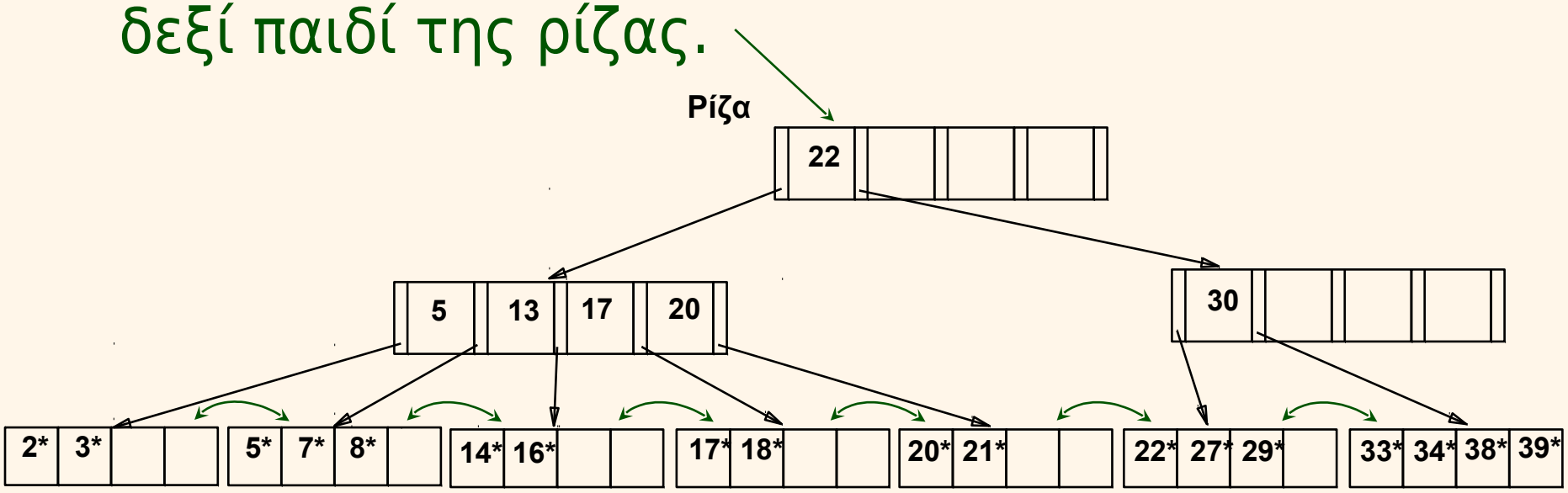
... και μετά τη διαγραφή του 24*

- Πρέπει να συγχωνεύσουμε.
- Προσέξτε το 'ξεφόρτωμα' της καταχώρισης ευρετηρίου (στα δεξιά), και το 'τράβηγμα προς τα κάτω' της καταχώρισης ευρετηρίου (κάτω).



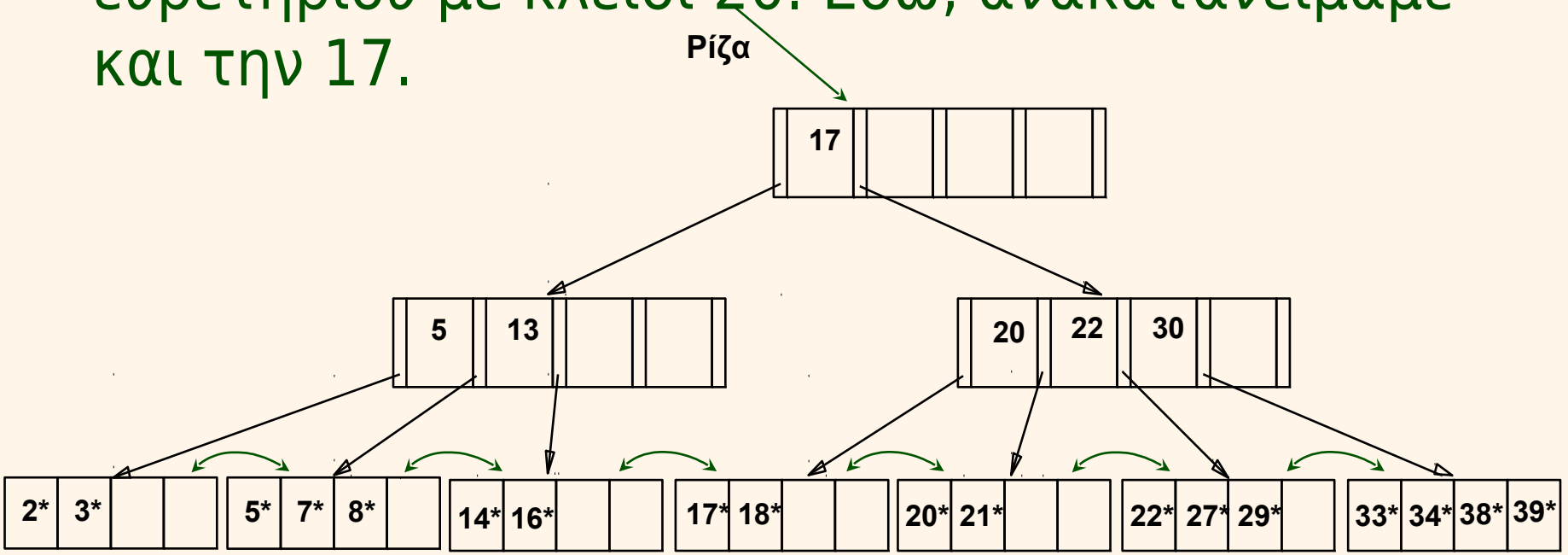
Παράδειγμα ανακατανομής σε μη-φύλλα

- Παρακάτω φαίνεται το δέντρο *κατά τη διαγραφή* του 24*. (Ποιό θα μπορούσε να είναι το αρχικό δέντρο;)
- Αντίθετα με το προηγούμενο παράδειγμα, είναι δυνατή η ανακατανομή από το αριστερό στο δεξί παιδί της ρίζας.



Μετά την ανακατανομή

- Διαισθητικά, οι καταχωρίσεις ανακατανέμονται δια της μεταφοράς της καταχώρισης διάσπασης στο γονιό.
- Αρκεί να ανακατανείμουμε την καταχώριση ευρετηρίου με κλειδί 20. Εδώ, ανακατανείμαμε και την 17.

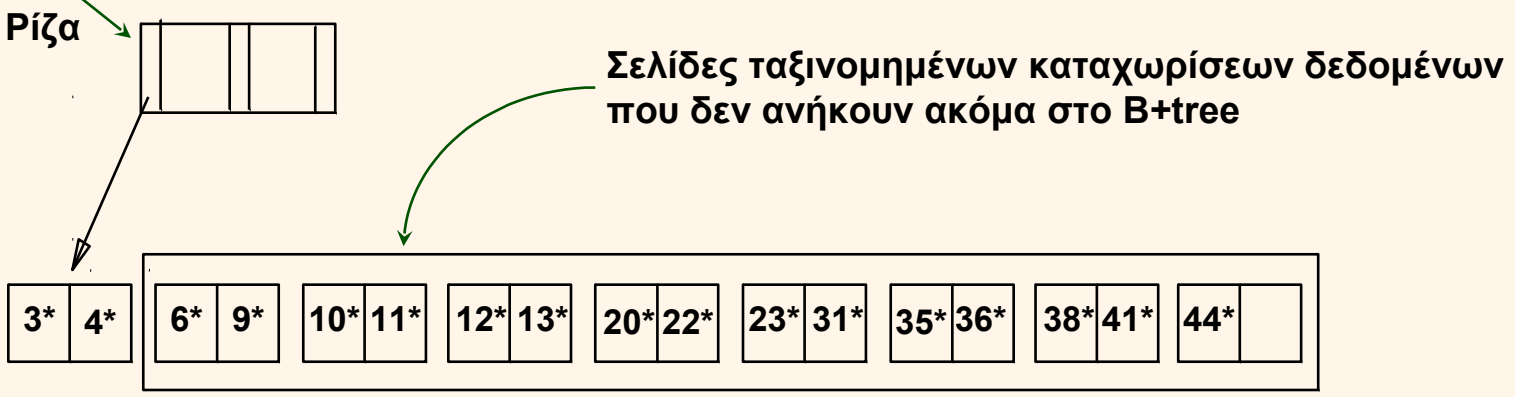


Συμπίεση Προθέματος Κλειδιού

- Απαραίτητη για την αύξηση του fan-out. (Γιατί;)
- Οι τιμές των κλειδιών στις καταχωρίσεις ευρετηρίου απλά 'κατευθύνουν την κυκλοφορία', Συχνά μπορούμε να τις συμπιέσουμε.
  - Π.χ., Αν έχουμε διαδοχικές καταχωρίσεις ευρετηρίου με τιμές κλειδιού αναζήτησης *Dannon Yogurt*, *David Smith και Devarakonda Murthy*, μπορούμε να συντομεύσουμε το *David Smith* σε *Dav*. (Και τα υπόλοιπα κλειδιά μπορούν να συντομευτούν ...)
    - Σωστά; Όχι ακριβώς! Τί θα συμβεί αν υπάρχει η καταχώριση δεδομένων *Davey Jones;* (Μπορούμε μόνο να συμπιέσουμε το *David Smith* σε *Davi*)
    - Γενικά, κατά τη συμπίεση, πρέπει κάθε καταχώριση ευρετηρίου να παραμένει μεγαλύτερη από όλες τις τιμές κλειδιών (οποιοδήποτε υποδέντρου που βρίσκεται) στα αριστερά της.
- Εισαγωγή/Διαγραφή: πρέπει να τροποποιηθούν ανάλογα.

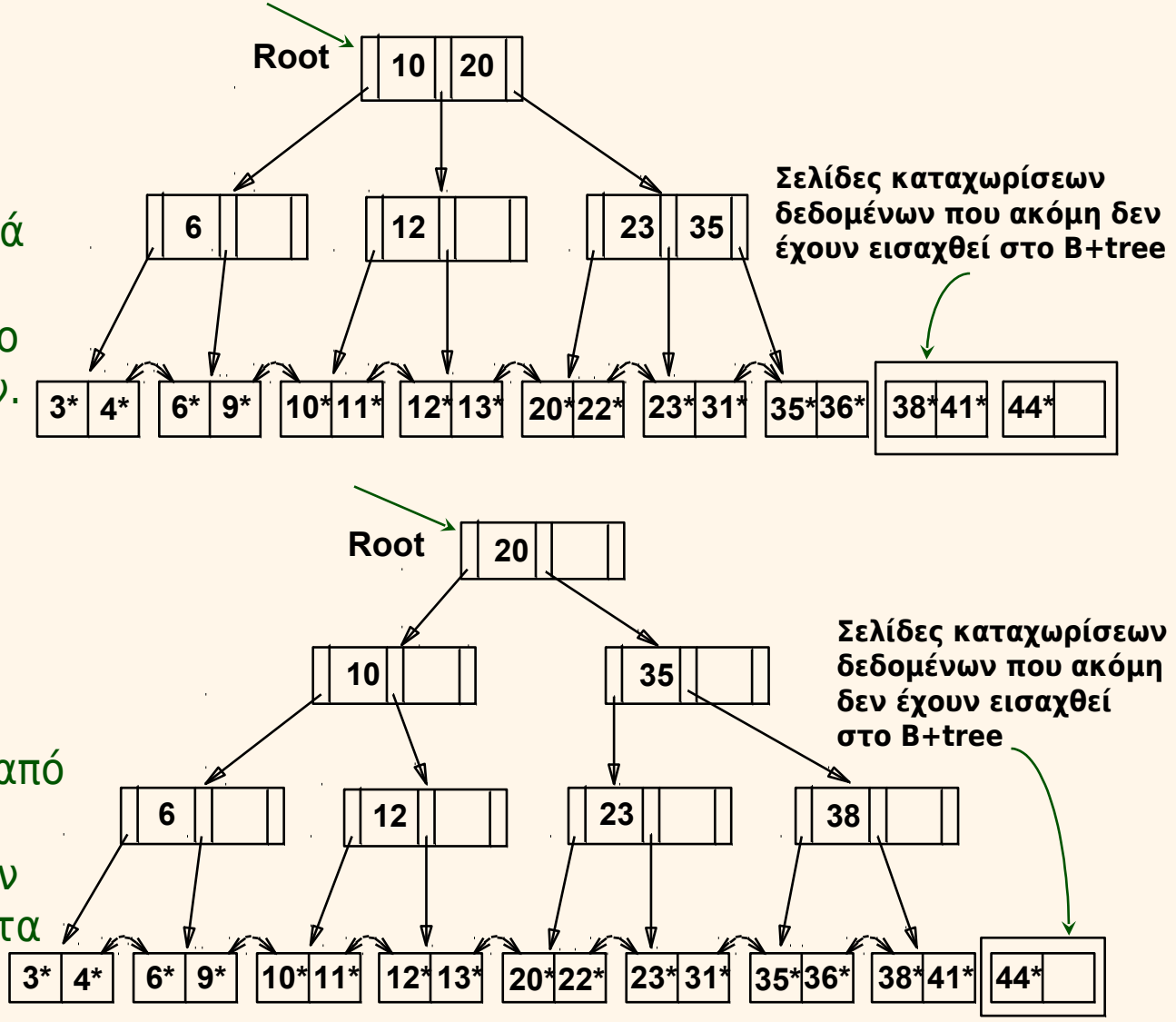
Μαζική Φόρτωση B+Tree

- Αν έχουμε μια μεγάλη συλλογή εγγραφών και θέλουμε να δημιουργήσουμε ένα B+tree πάνω σε κάποιο πεδίο, η διαδοχική εισαγωγή των εγγραφών είναι πολύ αργή.
- Αυτό μπορεί να γίνει πολύ πιο αποτελεσματικά με *Μαζική Φόρτωση*.
- *Αρχικοποίηση*: Ταξινόμησε όλες τις καταχωρίσεις δεδομένων, και βάλε έναν pointer στην πρώτη σελίδα (φύλλο) σε μια νέα σελίδα (ρίζα).



Μαζική Φόρτωση (συνέχεια)

- Οι καταχωρίσεις ευρετηρίου για τα φύλλα εισάγονται πάντα στην πιο δεξιά σελίδα ευρετηρίου αμέσως πάνω από το επίπεδο των φύλλων. Όταν αυτή γεμίζει, διασπάται. (Η διάσπαση μπορεί να μεταφερθεί κατά τη δεξιότερη διαδρομή ως τη ρίζα.)
- Πολύ γρηγορότερη από τις διαδοχικές εισαγωγές, ειδικά αν λάβουμε υπόψη και τα κλειδώματα!



Σύνοψη της Μαζικής Φόρτωσης

- Επιλογή 1: πολλαπλές εισαγωγές
  - Αργή.
  - Δεν καταλήγει σε δέσμευση συνεχόμενων σελίδων για τα φύλλα.
- Επιλογή 2: *Μαζική Φόρτωση*
  - Πλεονεκτεί ως προς τον έλεγχο ταυτοχρονισμού.
  - Λιγότερες I/O κατά την κατασκευή.
  - Τα φύλλα αποθηκεύονται σε συνεχόμενες σελίδες (και, βέβαια, είναι συνδεδεμένα).
  - Μπορεί να ελεγχθεί ο “συντελεστής πληρότητας” των σελίδων.

Ένα σημείωμα για την `Τάξη'

- Η έννοια της *Τάξης* (**d**), στην πράξη αντικαθίσταται από το κριτήριο του χώρου (σελίδα '*τουλάχιστον 50% γεμάτη*').
  - Οι σελίδες ευρετηρίου συνήθως χωράνε πολύ περισσότερες καταχωρίσεις από τις σελίδες φύλλα.
  - Το γεγονός ότι υπάρχουν εγγραφές και κλειδιά αναζήτησης μεταβλητού μήκους, σημαίνει ότι διαφορετικοί κόμβοι θα περιέχουν διαφορετικό πλήθος καταχωρίσεων.
  - Ακόμα και με πεδία σταθερού μήκους, η ύπαρξη πολλών εγγραφών με την ίδια τιμή κλειδιού αναζήτησης (*διπλο-εγγραφές*) οδηγούν σε καταχωρίσεις δεδομένων μεταβλητού μήκους (αν χρησιμοποιούμε την Εναλλακτική (3)).