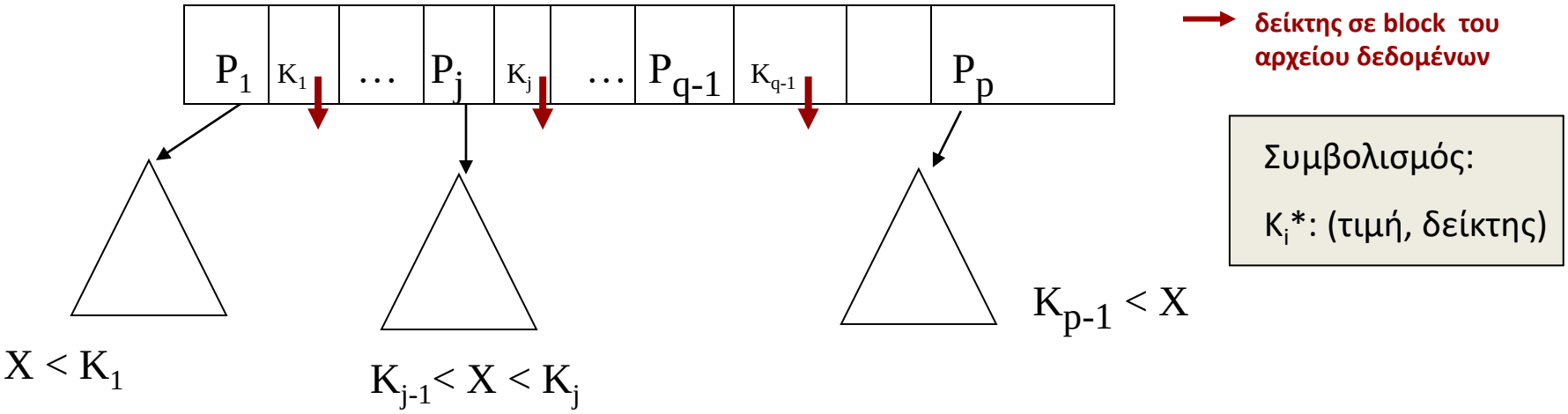
B-δέντρα, B^+-δέντρα

Δέντρα Αναζήτησης

Ένα δέντρο αναζήτησης (search tree) τάξεως ***p*** είναι ένα δέντρο τέτοιο ώστε κάθε κόμβος του περιέχει το πολύ ***p - 1*** τιμές αναζήτησης και ***p*** δείκτες ως εξής



$K_1 < K_2 < \ldots K_{q-1}$ και για όλες τις τιμές X στα υποδέντρα ισχύει $K_{j-1} < X < K_j$ για $1 < j < p$, $X < K_j$ για $j = 1$, και $K_{j-1} < X$ για $j = p$

Σημείωση: Γενικά στα ευρετήρια, ζεύγη <τιμή, προσδιοριστής εγγραφής>

Δέντρα Αναζήτησης

Κάθε κόμβος του δέντρου έχει μέγεθος ίσο με ένα block (σελίδα)

Ισοζυγισμένο: όλοι οι κόμβοι-φύλλα στο ίδιο επίπεδο

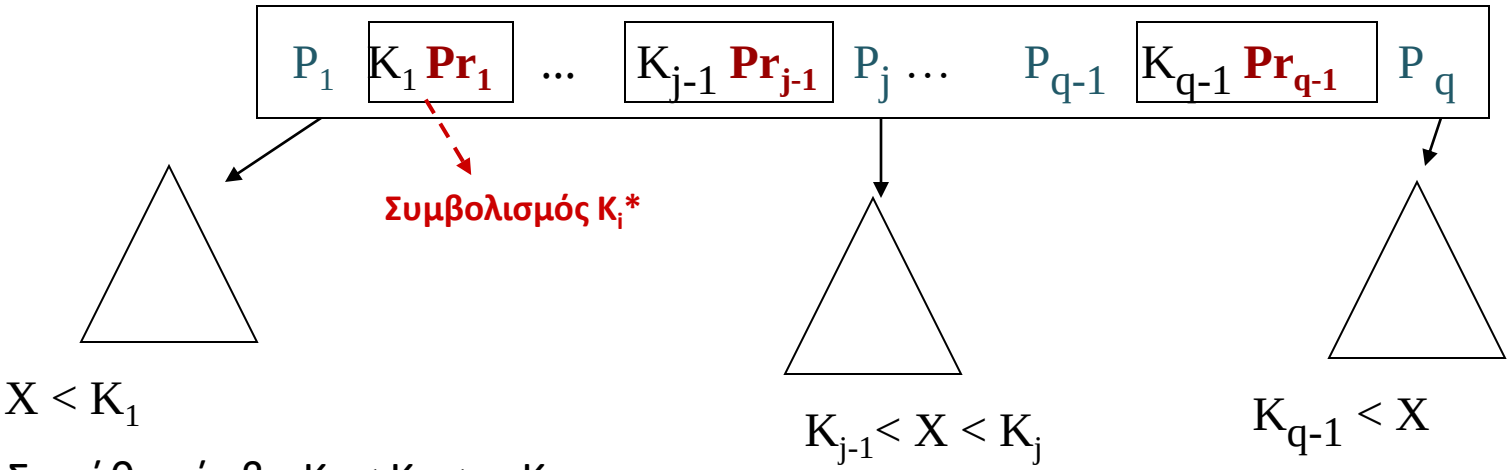
Β-δέντρο: ένα δέντρο αναζήτησης που παραμένει ισοζυγισμένο και χωρίς «πολύ αδειανούς» κόμβους

Β-δέντρα (ορισμός)

Ένα Β-δέντρο τάξεως (order) p ορίζεται ως εξής:

1. Κάθε εσωτερικός κόμβος είναι της μορφής

<P_1, <K_1, Pr_1>, P_2, <K_2, Pr_2>, … <K_{q-1}, Pr_{q-1}>, P_q>, q < p, όπου P_i δείκτης δέντρου, K_i τιμή αναζήτησης, Pr_i δείκτης δεδομένων



2. Σε κάθε κόμβο $K_1 < K_2 < … K_{q-1}$

3. Για όλες τις τιμές X στο υποδέντρο που δείχνει το P_j ισχύει $K_{j-1} < X < K_j$ για $1 < j < q$, $X < K_j$ για $j = 1$, και $K_{j-1} < X$ για $j = q$

B-δέντρα (ορισμός)

4. Κάθε κόμβος έχει το **πολύ** p δείκτες δέντρου

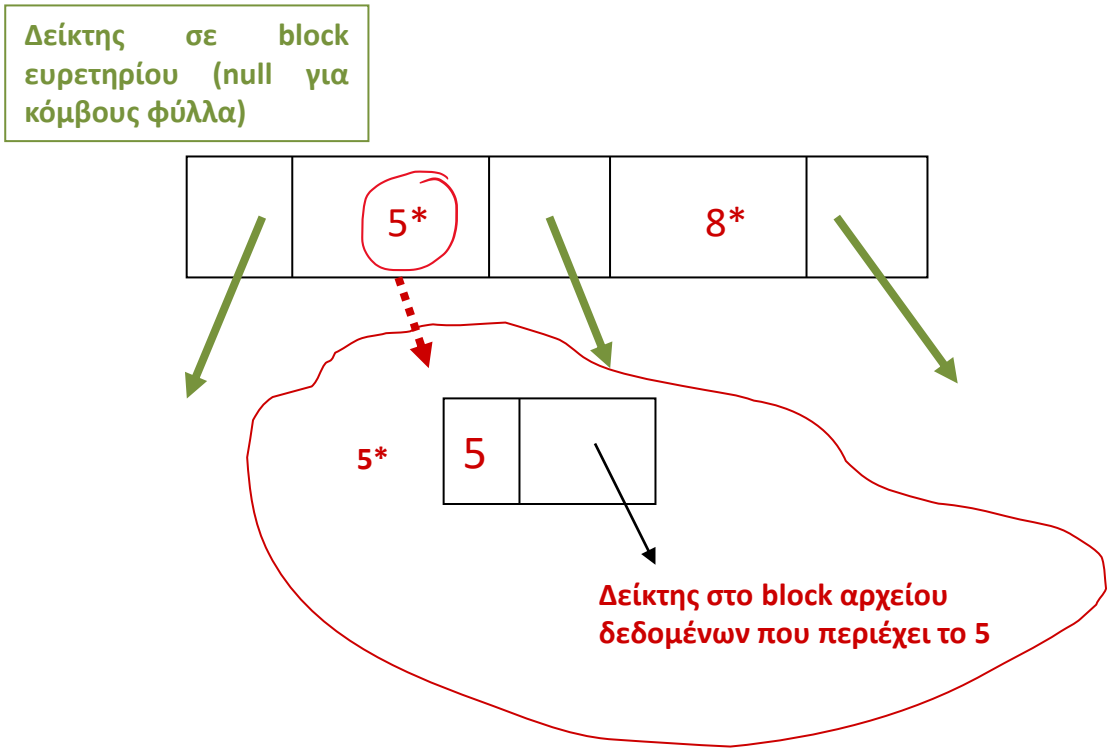
5. Κάθε κόμβος ***εκτός της ρίζας και των φύλλων*** έχει **τουλάχιστον** $\lceil (p/2) \rceil$ δείκτες. Η ρίζα έχει τουλάχιστον 2 εκτός αν είναι ο μόνος κόμβος του δέντρου.

6. Ένας κόμβος με q δείκτες δέντρου περιέχει q - 1 τιμές πεδίου αναζήτησης (και άρα και q - 1 δείκτες δεδομένων).

7. Όλα τα φύλλα βρίσκονται στο ίδιο επίπεδο. Τα φύλλα έχουν την ίδια δομή εκτός του ότι οι δείκτες δέντρου είναι null.

B-δέντρα (παράδειγμα)

τάξη ρ = 3 (2 τιμές ανά κόμβο, 3 δείκτες block ευρετηρίου)



B-δέντρα

Αναζήτηση

Διαβάζουμε το block της ρίζας

Αν η εγγραφή δεν υπάρχει στον κόμβο διαβάζουμε το αντίστοιχο block στο επόμενο επίπεδο

Εισαγωγή

Αναζήτηση του κατάλληλου φύλλου και εισαγωγή της τιμής σε αυτό

Τι γίνεται αν είναι «γεμάτος»; -> διάσπαση!

Β-δέντρα (εισαγωγή)

Αρχικά ένας μόνο κόμβος (ρίζα) στο Επίπεδο 0

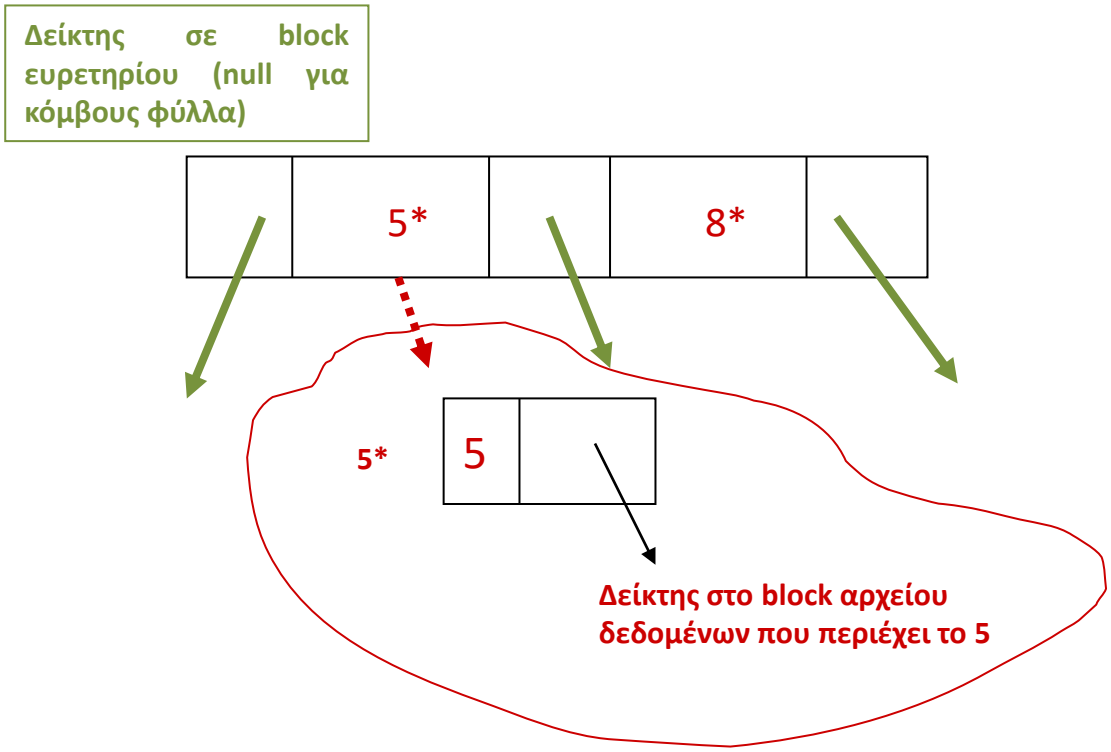
Όταν ο *κόμβος ρίζα* γεμίσει (p – 1 τιμές κλειδιού), μια νέα εισαγωγή οδηγεί στην διάσπαση του κόμβου σε δύο κόμβους στο Επίπεδο 1: η **μεσαία τιμή** πηγαίνει στη ρίζα, οι υπόλοιπες μοιράζονται εξίσου σε δύο κόμβους του Επιπέδου 1

Όταν *ένας κόμβος εκτός της ρίζας* γεμίσει, μια νέα εισαγωγή οδηγεί σε διάσπαση του κόμβου σε δύο κόμβους στο ίδιο επίπεδο και **μεταφορά της μεσαίας τιμής** στον γονέα του κόμβου

ΠΡΟΣΟΧΗ: η εισαγωγή της μεσαίας τιμής στο γονέα αν ο γονέας είναι γεμάτος μπορεί να οδηγήσει σε *διάσπαση του γονέα*. Η διάσπαση μπορεί να οδηγήσει ως τη ρίζα, οπότε δημιουργείται και νέο επίπεδο.

B-δέντρα (εισαγωγή)

τάξη ρ = 3 (2 τιμές ανά κόμβο, 3 δείκτες block ευρετηρίου) - Εισαγωγή 5, 8, 7, 14, 19, 6, 10

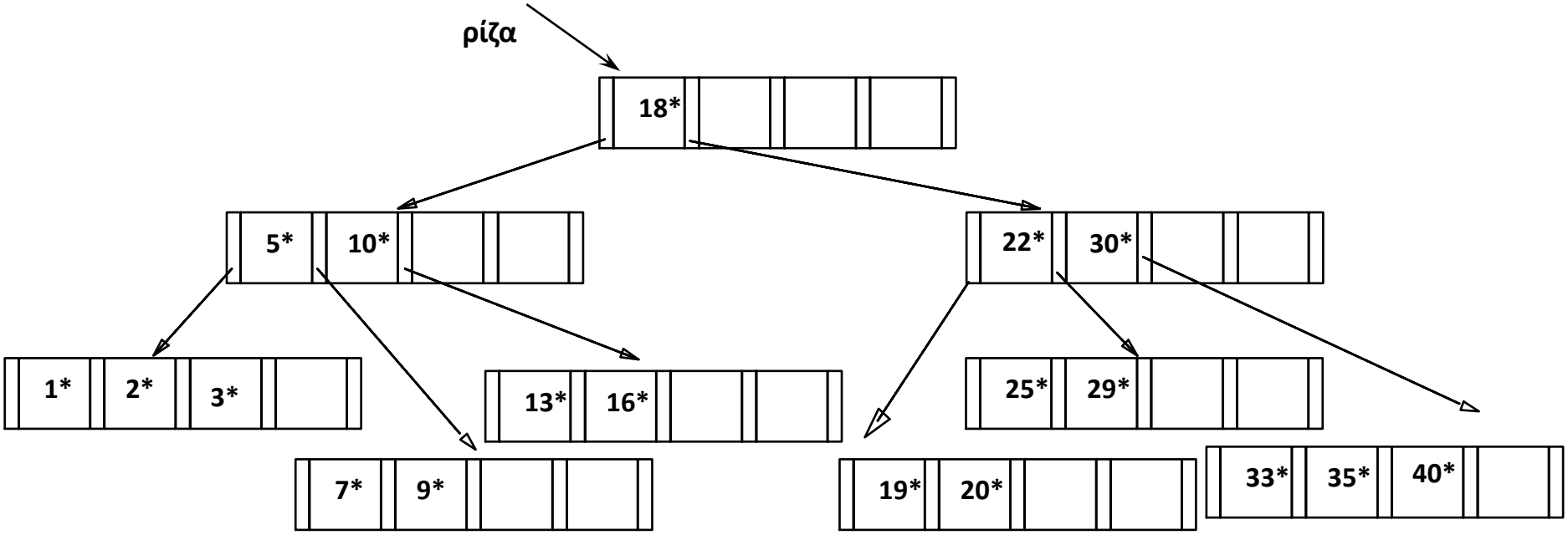


B-δέντρα (παράδειγμα)

Τάξης p = 5 -- το πολύ 4, τουλάχιστον 2 τιμές ανά κόμβο (εκτός της ρίζας)

5 10 3 18 16 25 7 22 30 2 9 33 40 29 19 20 13 1 35

B-δέντρα (παράδειγμα)



Β-δέντρα (διαγραφή)

Τιμή προς διαγραφή ανήκει σε *φύλλο* -> ok

Τιμή προς διαγραφή ανήκει σε *εσωτερικό κόμβο* ->

Αν σβήσουμε το K_i, τότε το μικρότερο κλειδί του υποδέντρου P_{i+1} πρέπει να το *αντικαταστήσει (δηλαδή το μικρότερο κλειδί του κόμβου στα δεξιά του κλειδιού που διαγράφεται)*

Τι γίνεται αν ο κόμβος «αδειάσει»;

Β-δέντρα (διαγραφή)

Αν υποχείλιση

αν είναι δυνατόν **ανακατανομή** με τον αριστερό αδελφό

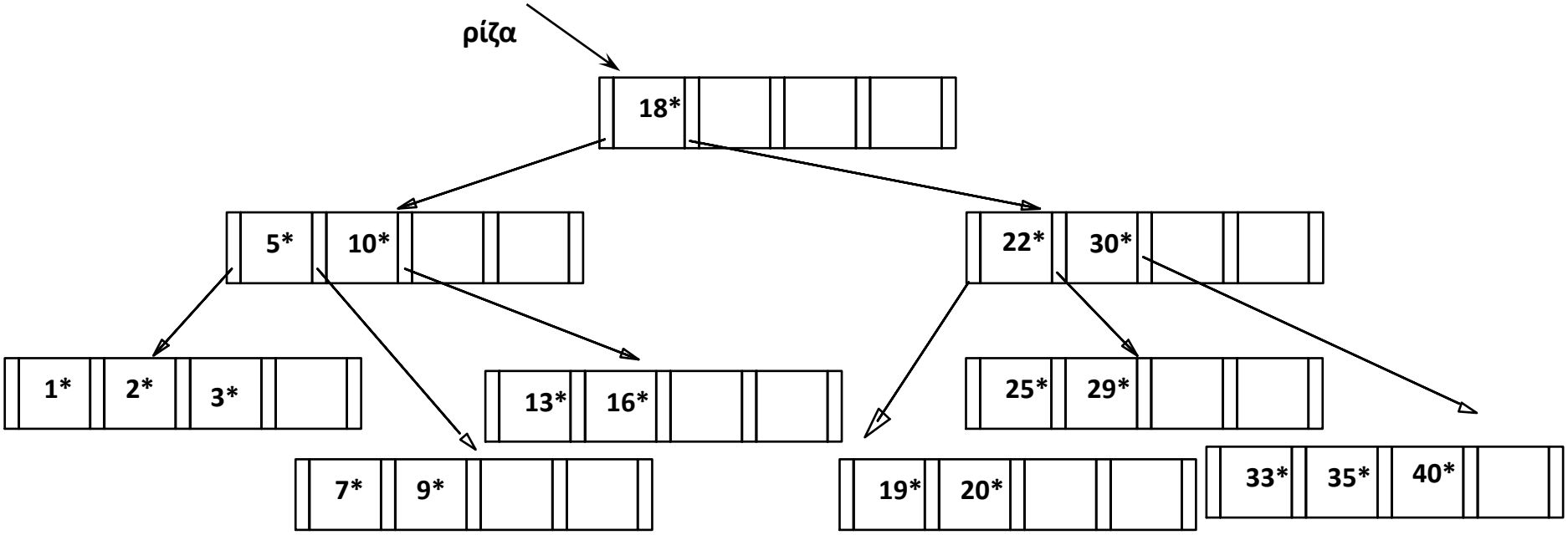
αν όχι, προσπάθεια ανακατανομής με το δεξιό αδελφό

αν όχι ανακατανομή, **συγχώνευση** των κόμβων

σε περίπτωση συγχώνευσης: *διαγράφουμε* και την αντίστοιχη εγγραφή στον γονέα (πιθανή υποχείλιση και στο γονέα)

- Σε κάθε περίπτωση (ανακατανομή και συγχώνευση) **κατεβάζουμε και την τιμή του γονέα** – και στο γονέα ανεβαίνει η νέα μεσαία τιμή

B-δέντρα (παράδειγμα)



B-δέντρα (παράδειγμα)

Διαγραφή τιμής σε φύλλο χωρίς υποχείλιση 1

Διαγραφή τιμής σε εσωτερικό κόμβο χωρίς υποχείλιση 30

Διαγραφή τιμής σε φύλλο με υποχείλιση και δανεισμό 7

Διαγραφή τιμής σε φύλλο με υποχείλιση και συγχώνευση 7 και μετά 5

Διαγραφή του 18

Β-δέντρα (υπολογισμός τάξης)

- Κάθε κόμβος του Β-δέντρου καταλαμβάνει μια σελίδα (block)

Υπολογισμός τάξης p (ώστε κάθε κόμβος να καταλαμβάνει ένα block)

Έστω *B* μέγεθος block, *V* μέγεθος πεδίου αναζήτησης (δηλαδή του πεδίου ευρετηριοποίησης), *Pr* μέγεθος δείκτη δεδομένων (εγγραφής) και *P* μέγεθος δείκτη δέντρου (block)

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

$p * (P + Pr + V) \leq B + V + Pr$

$p \leq (B + V + Pr) / (P + Pr+ V)$

Παράδειγμα, V = 9 bytes, B = 512 bytes, Pr = 7 bytes, P = 6 bytes,

τότε p = 23

B-δέντρα (υπολογισμός επιπέδων)

Έστω όπως πριν, p = 23. Έστω ότι κάθε κόμβος είναι γεμάτος κατά ~69%.

Πόσα επίπεδα χρειαζόμαστε για να ευρετηριοποιήσουμε 65.000 τιμές;

(p - 1)*0,69 = 22*0,69 = 15 κλειδιά και 15 + 1 = 16 δείκτες ανά κόμβο

	#κόμβων	#τιμές	#δείκτες
Ρίζα	1 κόμβος	15 (22*0,69) καταχωρήσεις	16 δείκτες
Επίπεδο 1:	16 κόμβοι	240 (16*15) καταχωρήσεις	256 δείκτες
Επίπεδο 2:	256 κόμβοι	3.840 (256*15) καταχωρήσεις	4.096 δείκτες
Επίπεδο 3:	4.096 κόμβοι	61.440	

Σύνολο: 61.440 + 3.840 + 240 + 15 (65.535)

Β-δέντρα

- Ποιες τιμές του πεδίου ευρετηριοποίησης εισάγουμε στο Β-δέντρο;

  Όπως και στα ευρετήρια που είδαμε σε προηγούμενα μαθήματα αυτό εξαρτάται από το πεδίο δεικτοδότησης, δηλαδή αν είναι πεδίο διάταξης ή κλειδί

  Αν όχι πεδίο διάταξης, πυκνά ευρετήρια

- Αναζήτηση με βάση διάστημα τιμών;

B+-δέντρα

Διαφορά B^+ από B-δέντρο: Αποθηκεύουμε δείκτες δεδομένων (στο αρχείο δεδομένων) μόνο στα φύλλα

Δύο τύποι κόμβων:

- εσωτερικοί κόμβοι
- φύλλα

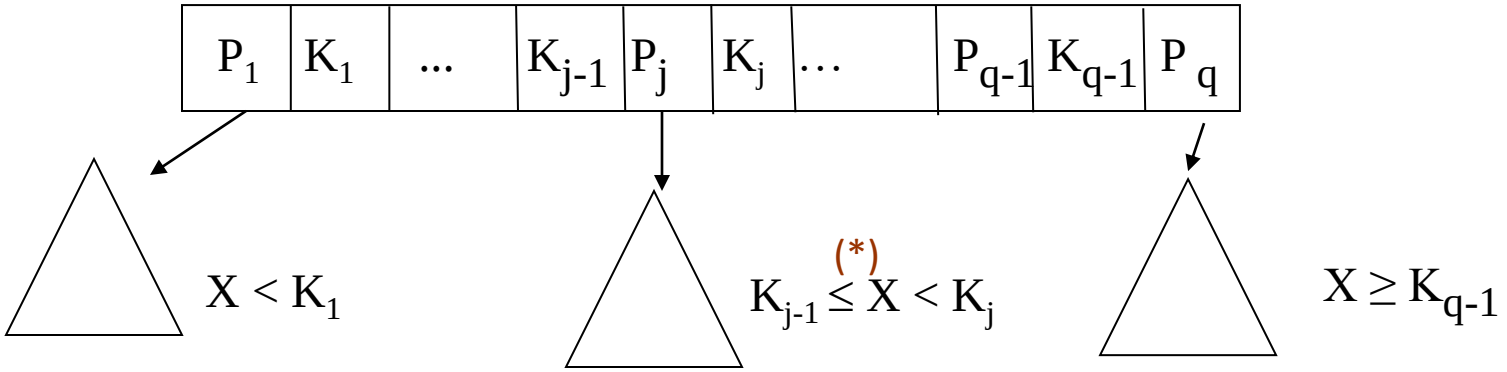
- Όλες οι τιμές του πεδίου ευρετηριοποίησης εμφανίζονται στα φύλλα.
- Οι τιμές που εμφανίζονται σε εσωτερικούς κόμβους παρέχουν πληροφορία μόνο για τη διάσχιση του δέντρου
- Κάποιες τιμές μπορεί να εμφανίζονται *παραπάνω από μια* φορά

B+-δέντρα (ορισμός)

Ένα B^+-δέντρο τάξεως (order) p για τους εσωτερικούς κόμβους και p_{leaf} για τα φύλλα ορίζεται ως εξής:

1. Κάθε εσωτερικός κόμβος είναι της μορφής

<P_1, K_1, P_2, K_2, , … K_{q-1}, P_{q-1}, P_q> $q \leq p$, όπου P_i δείκτης δέντρου, K_i τιμή αναζήτησης



2. Σε κάθε εσωτερικό κόμβο $K_1 < K_2 < \ldots K_{q-1}$

3. Για όλες τις τιμές X στο υποδέντρο που δείχνει το P_j ισχύει $K \leq X < K_j$ για $1 < j < q$, $X < K_j$ για $j = 1$, και $K_{j-1} \leq X$ για $j = q$

(*) κάνουμε τη σύμβαση ότι η τιμή πάει δεξιά, θα μπορούσε και $K_{j-1} < X \leq K_j$

B+-δέντρα (ορισμός)

4. Κάθε εσωτερικός κόμβος έχει το **πολύ p** δείκτες δέντρου

5. Κάθε εσωτερικός κόμβος ***εκτός της ρίζας*** έχει **τουλάχιστον ⌈(p/2)⌉** δείκτες. Η ρίζα έχει τουλάχιστον 2 εκτός αν είναι ο μόνος κόμβος του δέντρου.

6. Ένας κόμβος με q δείκτες δέντρου περιέχει q - 1 τιμές πεδίου αναζήτησης

B+-δέντρα (ορισμός)

1. Κάθε κόμβος-φύλλο είναι της μορφής

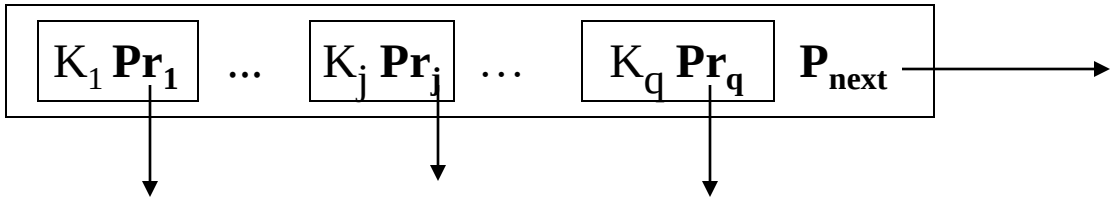
<<K_1, Pr_1>, <K_2, Pr_2>, … <K_q, Pr_q>, P_{next}>, $q \leq p_{leaf}$, όπου

p_{leaf} είναι η τάξη των κόμβων-φύλλων

K_i τιμή αναζήτησης,

Pr_i δείκτης δεδομένων που δείχνει στο block (ή στην εγγραφή) με τιμή στο πεδίο αναζήτησης K_i (ή σε ένα block ενδιάμεσου επιπέδου αν το πεδίο αναζήτησης δεν είναι κλειδί),

P_{next} δείχνει στο επόμενο φύλλο και χρησιμοποιείται για τη γρήγορη ανάγνωση του αρχείου σε διάταξη



2. Σε κάθε κόμβο-φύλλο $K_1 < K_2 < \ldots K_q$

B+-δέντρα (ορισμός)

3. Κάθε κόμβος-φύλλο έχει το **πολύ p_{leaf}** τιμές

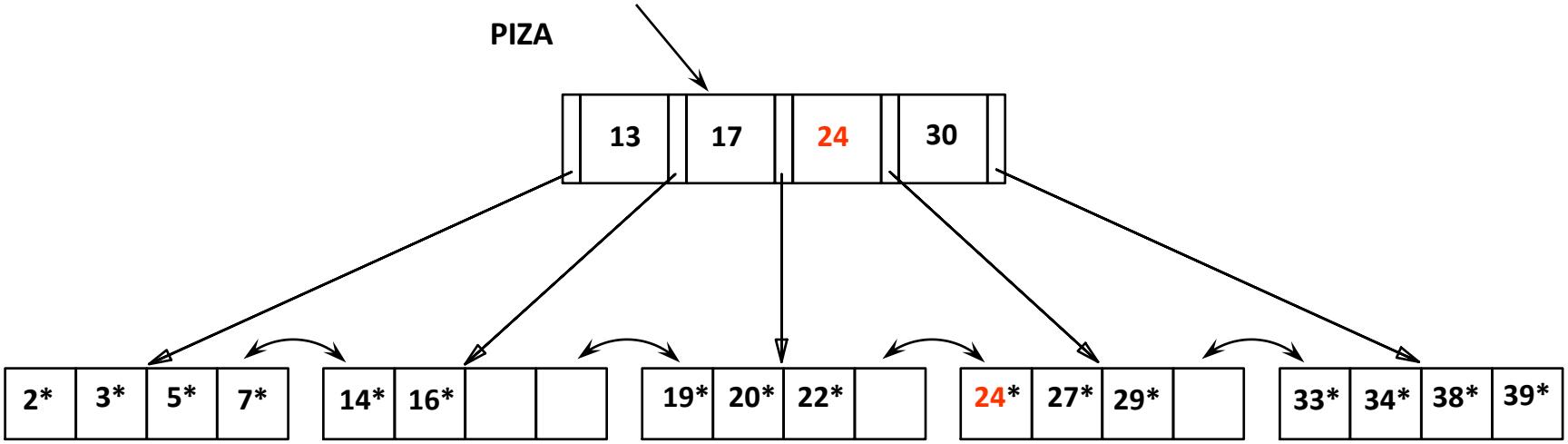
4. Κάθε κόμβος-φύλλο έχει **τουλάχιστον $\lceil (p_{leaf}/2) \rceil$** τιμές.

5. Όλοι οι κόμβοι-φύλλα βρίσκονται στο ίδιο επίπεδο.

B+-δέντρα

Η αναζήτηση ξεκινά από τη ρίζα, και οι συγκρίσεις των κλειδιών μας οδηγούν στα φύλλα

Αναζήτηση για τα 5, 15, 24, .. όλες οι καταχωρήσεις >= 17 ...



B+-δέντρα (εισαγωγή)

1. Αναζήτηση του φύλλου για εισαγωγή: έστω φύλλο P

2. Εισαγωγή τιμής K στο κόμβο P

    Αν ο κόμβος-φύλλο δεν είναι γεμάτος

        εισαγωγή της τιμής

B+-δέντρα (εισαγωγή)

Αν ο κόμβος-φύλλο είναι γεμάτος (έχει p_{leaf} εγγραφές)

διάσπαση του κόμβου:

-- οι πρώτες $k = \lfloor (p_{leaf} + 1)/2 \rfloor$ παραμένουν στον κόμβο

-- οι υπόλοιπες σε καινούργιο κόμβο

-- εισαγωγή (αντιγραφή) της k+1-οστής τιμής (K_{k+1}) στο γονέα

Αν ένας εσωτερικός κόμβος είναι γεμάτος (έχει p εγγραφές)

διάσπαση του κόμβου: έστω $k = \lfloor ((p+1)/2) \rfloor$

-- οι εγγραφές μέχρι το P_k (μετά την εισαγωγή) παραμένουν στον κόμβο

-- η k+1-οστή K_{k+1} τιμή μεταφέρεται (δεν αντιγράφεται) στον πατέρα

-- οι υπόλοιπες σε καινούργιο κόμβο

B+-δέντρα (εισαγωγή)

Οι διασπάσεις κόμβων (εκτός ρίζας) “μεγαλώνουν” το δέντρο

Η διάσπαση της ρίζας “ υψώνει ” το δέντρο

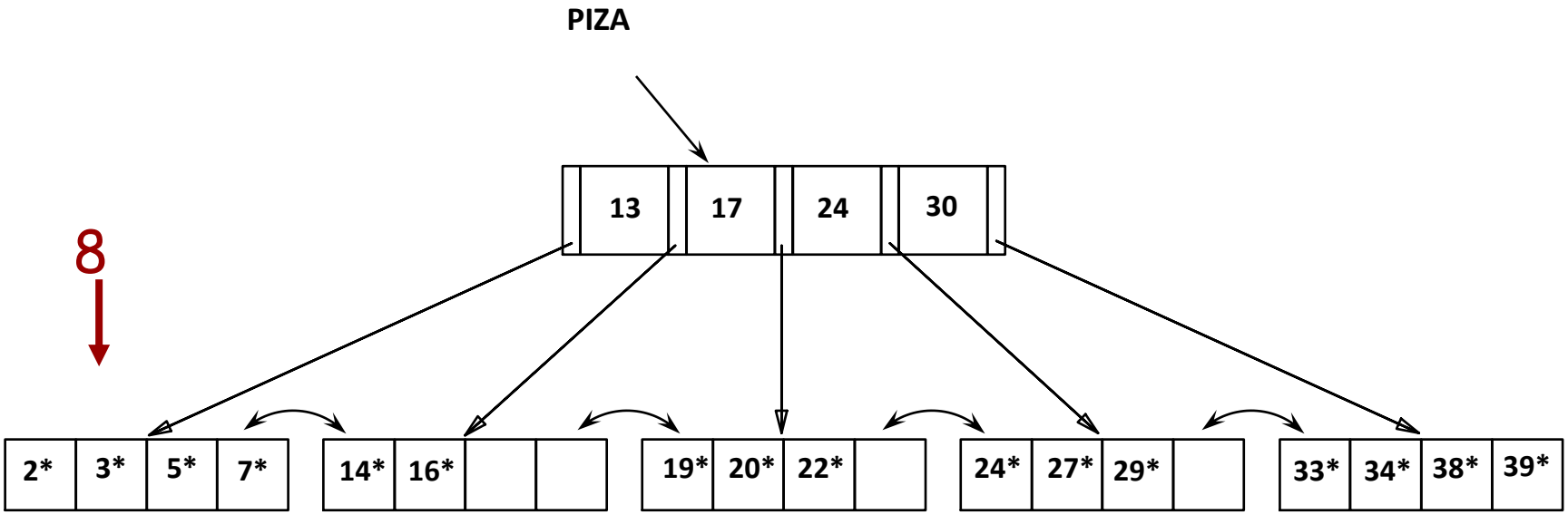
B+-δέντρα (παράδειγμα)

5, 8, 7, 14, 19, 6, 10

και τάξη ρ = 3 (2 τιμές ανά κόμβο, 3 δείκτες block ευρετηρίου) και p_{leaf} = 2

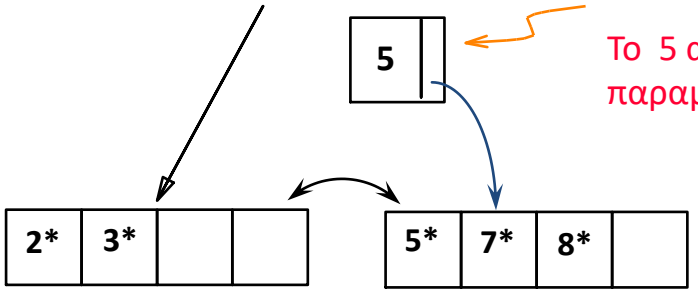
B+-δέντρα (παράδειγμα)

Εισαγωγή της καταχώρησης 8*



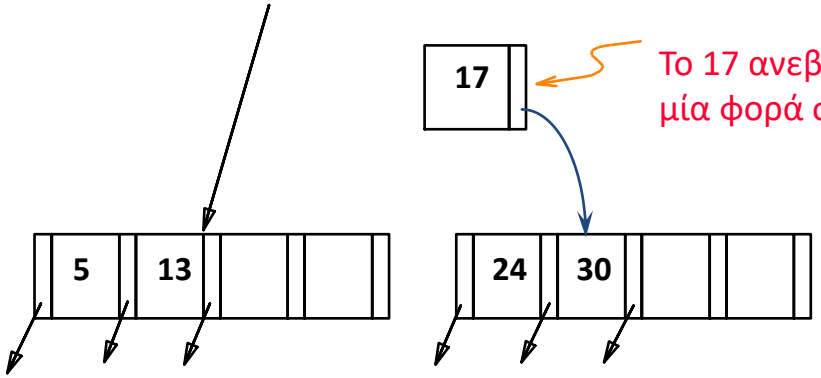
B+-δέντρα (παράδειγμα)

Καταχώρηση στον κόμβο γονέα (αντιγραφή)



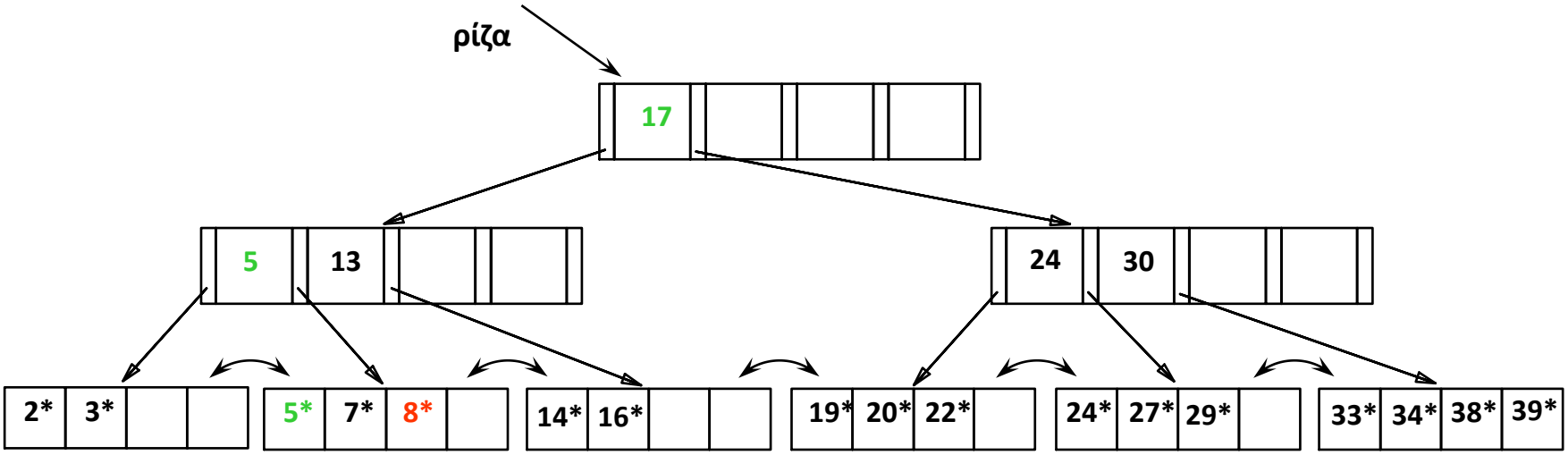
Το 5 ανεβαίνει επάνω, αλλά παραμένει και στο φύλλο

Καταχώρηση στον κόμβο γονέα (μεταφορά)



Το 17 ανεβαίνει επάνω και παρουσιάζεται μόνο μία φορά στο ευρετήριο (σε αντίθεση με τα φύλλα)

B+-δέντρα (παράδειγμα)



Η ρίζα διασπάστηκε οδηγώντας σε αύξηση του ύψους.

B+-δέντρα

Όλες οι τιμές εμφανίζονται στα φύλλα και *κάποιες επαναλαμβάνονται* και σε εσωτερικούς κόμβους (η τιμή Κ σε ένα εσωτερικό κόμβο μπορεί επίσης να εμφανίζεται ως η *πιο αριστερή τιμή* στο φύλλο του υποδέντρου με ρίζα το δείκτη στα δεξιά του Κ)

B+-δέντρα (διαγραφή)

Σε αντίθεση με τα B-δέντρα, οι τιμές είναι **μόνο** στα φύλλα

1. Αναζήτηση του φύλλου που περιέχει το K: έστω φύλλο P

2. Αν υποχείλιση

αν είναι δυνατόν ανακατανομή με τον αριστερό αδελφό ($> \lceil (n/2) \rceil$)

αν όχι, αν είναι δυνατόν ανακατανομή με το δεξιό αδελφό

αν όχι, συγχώνευση και των τριών κόμβων σε δύο κόμβους

B+-δέντρα (διαγραφή)

2. Αν υποχείλιση (αναλυτικά)

<ανακατανομή εγγραφών>

Αν είναι δυνατόν ανακατανομή με τον αριστερό αδελφό (>$\lceil (n/2) \rceil$)

αν όχι, αν είναι δυνατόν ανακατανομή με το δεξιό αδελφό

ανακατανομή εγγραφών σε κάθε κόμβο

βρείτε την εγγραφή στο γονέα του δεξιού κόμβου N

αντικατάσταση της τιμής κλειδιού στο γονέα τους με τη μικρότερη τιμή του κόμβου N

<συγχώνευση κόμβων>

Αν δεν είναι δυνατή η ανακατανομή

συγχώνευση κόμβων

οδηγεί σε διαγραφή στο παραπάνω επίπεδο, σβήνεται η εγγραφή που δείχνει στον κόμβο (πιθανότητα νέας υποχείλισης)

B+-δέντρα (διαγραφή)

Ειδικά για την ανακατανομή εσωτερικών κόμβων

Πάλι μέσω του γονέα τους

Δηλαδή θεωρούμε **και** την τιμή του γονέα στην ανακατανομή

Η τιμή αυτή αλλάζει στο γονέα

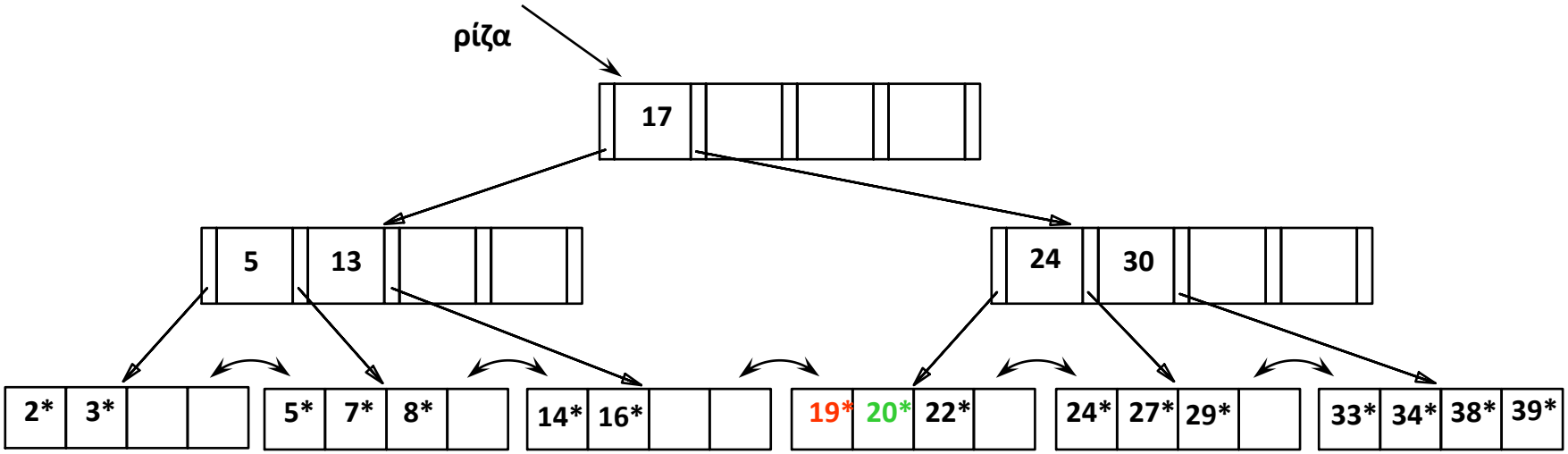
Εσωτερικοί κόμβοι

Ειδική περίπτωση στη συγχώνευση εσωτερικών κόμβων, όταν συγχωνεύεται ο ακραίος αριστερός δείκτης ενός εσωτερικού κόμβου (ο οποίος δεν έχει τιμή)

Τότε, πρέπει να συμβουλευτούμε τον γονέα των δύο κόμβων που συγχωνεύονται -> χρήση της τιμής του δείκτη που δείχνει σε αυτόν τον κόμβο

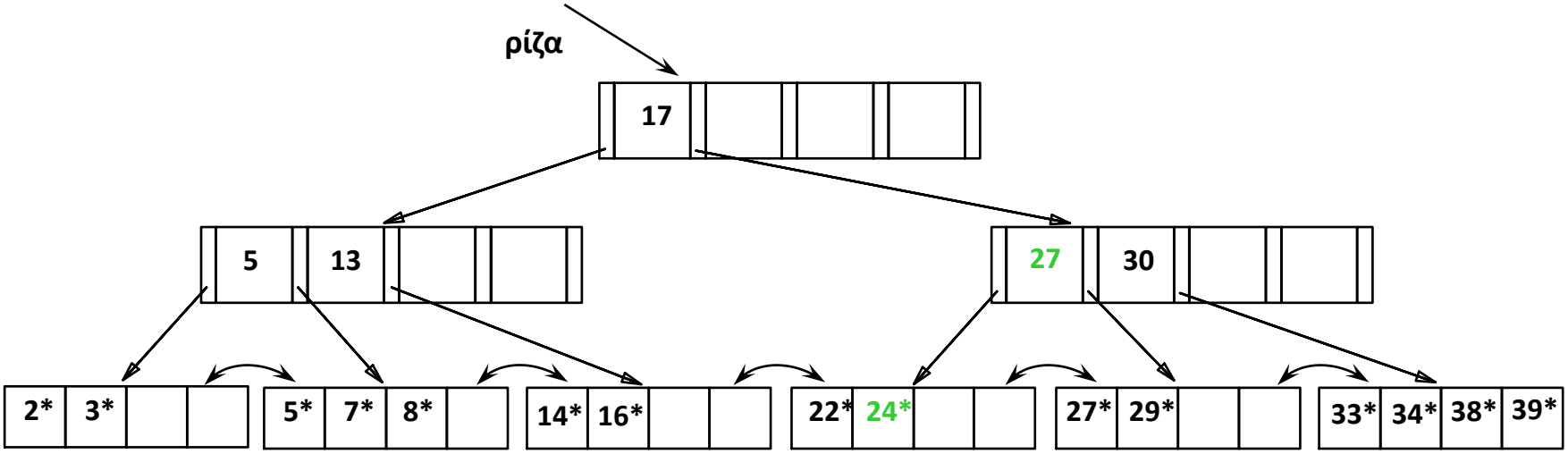
«Κατεβάζουμε» την τιμή από τον γονέα ως πιο αριστερή τιμή στον προς συγχώνευση κόμβο

B+-δέντρα (παράδειγμα)



Διαγραφή 19, 20

B+-δέντρα (παράδειγμα)

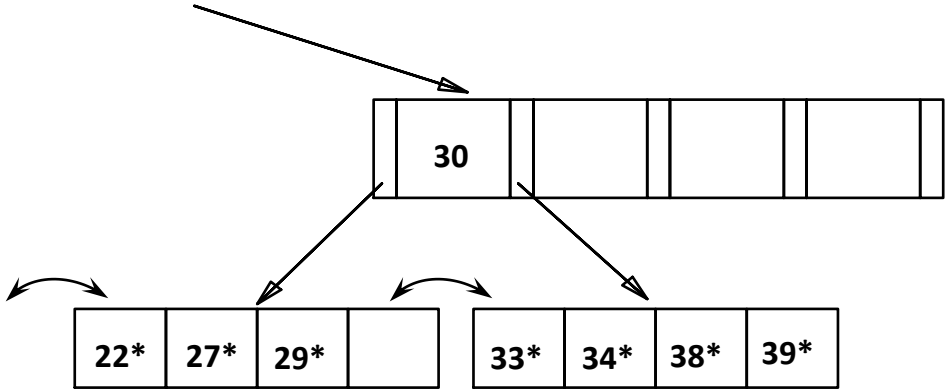


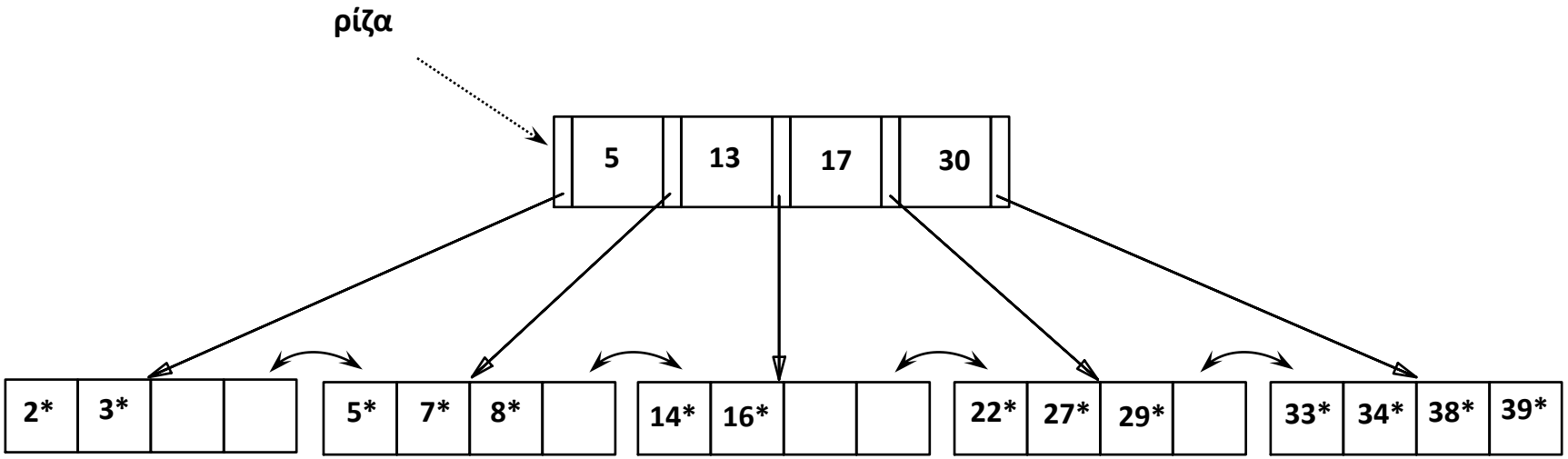
Το παράδειγμα μετά τη διαγραφή του 19* και του 20* (ανακατανομή με δεξί αδελφό και αντικατάσταση του 24 με 27)

Διαγραφή του 24 ->

B+-δέντρα (παράδειγμα)

Τέλος, η διαγραφή του 24*
(συγχώνευση)

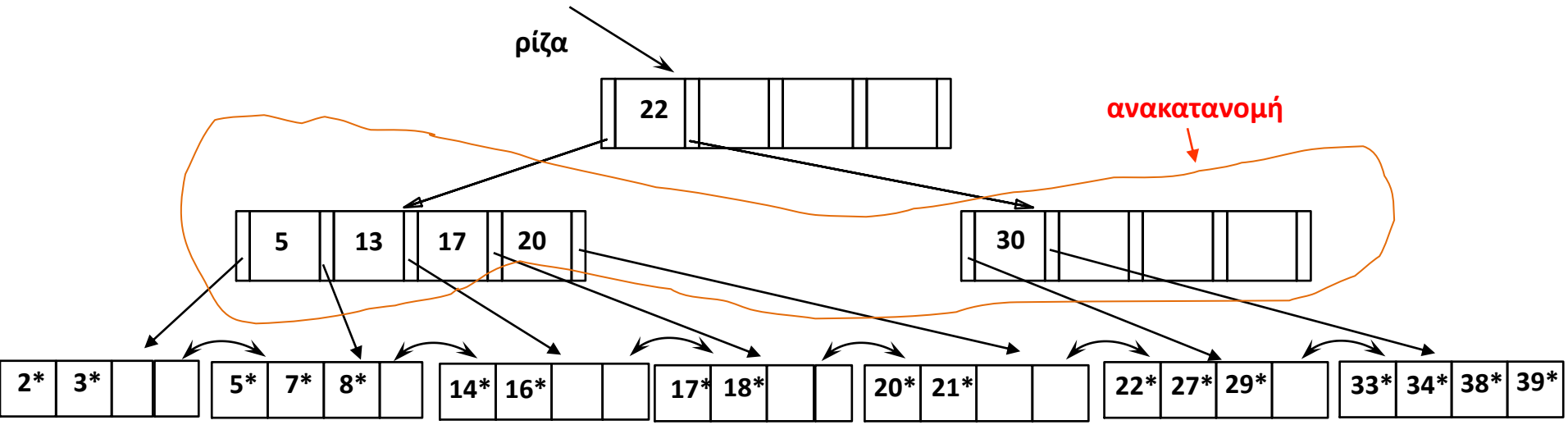




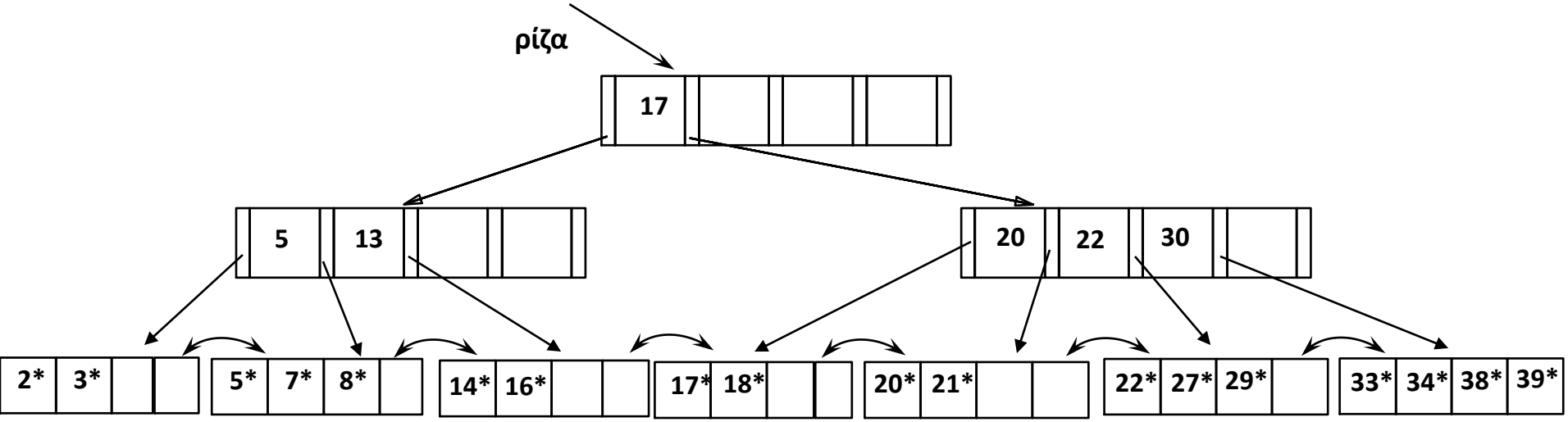
B+-δέντρα (παράδειγμα)

Παράδειγμα ανακατανομής

Έστω στο παρακάτω δέντρο μετά από συγχώνευση φύλλων

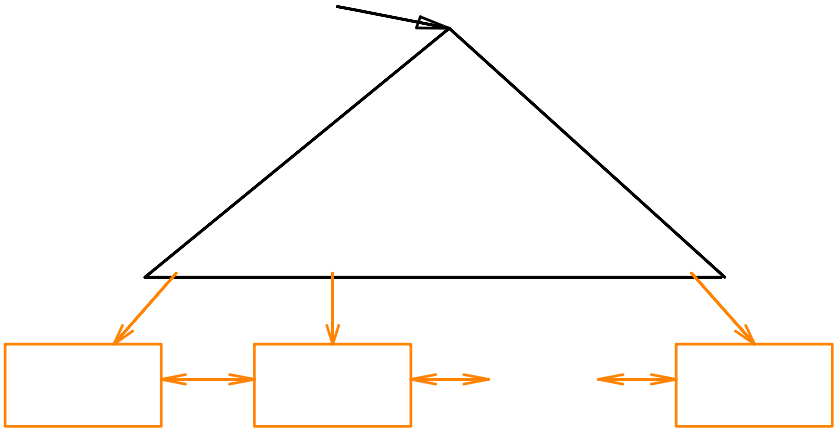


B+-δέντρα (παράδειγμα)



B+-δέντρα (συνοπτικά)

- Εισαγωγή/Διαγραφή με κόστος $\log_F N$ --- κρατούν το δέντρο σε ισορροπημένη μορφή. (F = διακλάδωση (τάξη), N = αριθμός των φύλλων)
- Ελάχιστη πληρότητα 50% (εκτός της ρίζας).
- Εξαιρετική δομή ΚΑΙ για ερωτήσεις ισότητας ΚΑΙ για ερωτήσεις διαστήματος (range queries).
- Το αρχείο δεδομένων μπορεί να είναι ή όχι ταξινομημένο



Καταχωρήσεις Ευρετηρίου

(Άμεση Αναζήτηση)

Καταχωρήσεις Δεδομένων
(«Σύνολο ακολουθίας»)

Τι αποθηκεύουμε στα φύλλα

Προσέγγιση 1
Δείκτη στη θέση της πλειάδας με την συγκεκριμένη τιμή στο πεδίο δεικτοδότησης

Προσέγγιση 2
Αν το πεδίο δεικτοδότησης είναι κλειδί, ορισμένα συστήματα μπορεί να αποθηκεύουν και την ίδια την πλειάδα (δηλαδή, το επίπεδο φύλλων είναι το αρχείο δεδομένων)

B+-δέντρα (υπολογισμός τάξης)

- Κάθε κόμβος του B+-δέντρου καταλαμβάνει μια σελίδα (block)

Τάξη p ώστε κάθε εσωτερικός-κόμβος να χωρά σε ένα block

Έστω B μέγεθος block, V μέγεθος πεδίου αναζήτησης, Pr μέγεθος δείκτη δεδομένων (εγγραφής) και P μέγεθος δείκτη δέντρου (block)

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

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

$p \leq (B + V) / (P + V)$

Παράδειγμα, V = 9 bytes, B = 512, Pr = 7 bytes, P = 6 bytes, τότε p = 34

Για B-δέντρο, p = 23

B+-δέντρα (υπολογισμός τάξης)

Τάξη p_{leaf} ώστε κάθε φύλλο να χωρά σε ένα block

Έστω B μέγεθος block, V μέγεθος πεδίου αναζήτησης, Pr μέγεθος δείκτη δεδομένων (εγγραφής) και P μέγεθος δείκτη δέντρου (block)

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

$p_{leaf} * (Pr + V) \leq B - P$

$p_{leaf} \leq (B - P) / (Pr + V)$

Παράδειγμα, V = 9 bytes, B = 512, Pr = 7 bytes, P = 6 bytes, τότε $p_{leaf} = 31$

B+-δέντρα (υπολογισμός επιπέδων)

Παράδειγμα, V = 9 bytes, B = 512, Pr = 7 bytes, P = 6 bytes, τότε p = 34. Έστω ότι κάθε κόμβος είναι *γεμάτος κατά 69%*. *Πόσες καταχωρήσεις (τιμές) χωρά αν έχει 4 επίπεδα*

Ρίζα	1 κόμβος	22 (33*0,69) καταχωρήσεις	23 δείκτες
Επίπεδο 2:	23 κόμβοι	506 (23*22) καταχωρήσεις	529 δείκτες
Επίπεδο 3:	529κόμβοι	11.638 (529*22) καταχωρήσεις	12.167 δείκτες
Επίπεδο φύλλων:	12.167 κόμβοι	255.507 (12.167 * 31 * 0.69) δείκτες δεδομένων	

Σε 3 επίπεδα 255.507 εγγραφές έναντι 65.535 για το B-δέντρο

Σημείωση: εγγραφές μόνο στα φύλλα

B+-δέντρα στην πράξη

- Βαθμός ή *μέση τιμή διακλάδωσης* (fan out) = 133
- Τυπική χωρητικότητα:
  - Ύψος 4: 133^4 = 312.900.700 εγγραφές
  - Ύψος 3: 133^3 = 2.352.637 εγγραφές
- Μπορεί να κρατά *τα υψηλότερα επίπεδα στη μνήμη* (buffer):
  - Επίπεδο 1 = 1 block = 8 Kbytes
  - Επίπεδο 2 = 133 blocks = 1 Mbyte
  - Επίπεδο 3 = 17.689 blocks = 133 MBytes

Ευρετήρια (ανακεφαλαίωση)

Είδη Ευρετηρίων

- Ευρετήριο ενός επιπέδου ένα διατεταγμένο αρχείο με εγγραφές (<τιμή, δείκτης>)
- Ευρετήριο πολλών επιπέδων
- Ευρετήρια δομής δέντρου (B-δέντρα, B+-δέντρα)
- Ευρετήρια κατακερματισμού /* θα τα δούμε στο επόμενο μάθημα

  το ευρετήριο είναι ένας πίνακας κατακερματισμού

  κάθε κάδος έχει εγγραφές ευρετηρίου,

  δηλαδή εγγραφές της μορφής (<τιμή, δείκτης>)

  h(τιμή)-> κάδος του ευρετηρίου

Ευρετήρια (ανακεφαλαίωση)

Ορισμοί

Πρωτεύον: όταν το πεδίο ευρετηριοποίησης είναι πρωτεύον κλειδί και πεδίο διάταξης του αρχείου

Δευτερεύον: αλλιώς

Συστάδων (clustered index) αν η διάταξη των εγγραφών στο ευρετήριο όμοια ή παρόμοια αυτής των εγγραφών στο αρχείο δεδομένων (συμβαίνει, πχ όταν το ευρετήριο κτίζεται στο πεδίο ταξινόμησης του αρχείου δεδομένων)

Ευρετήρια (ανακεφαλαίωση)

Το πολύ ένα ευρετήριο συστάδων – δηλαδή ένα ευρετήριο στο πεδίο διάταξης του αρχείου

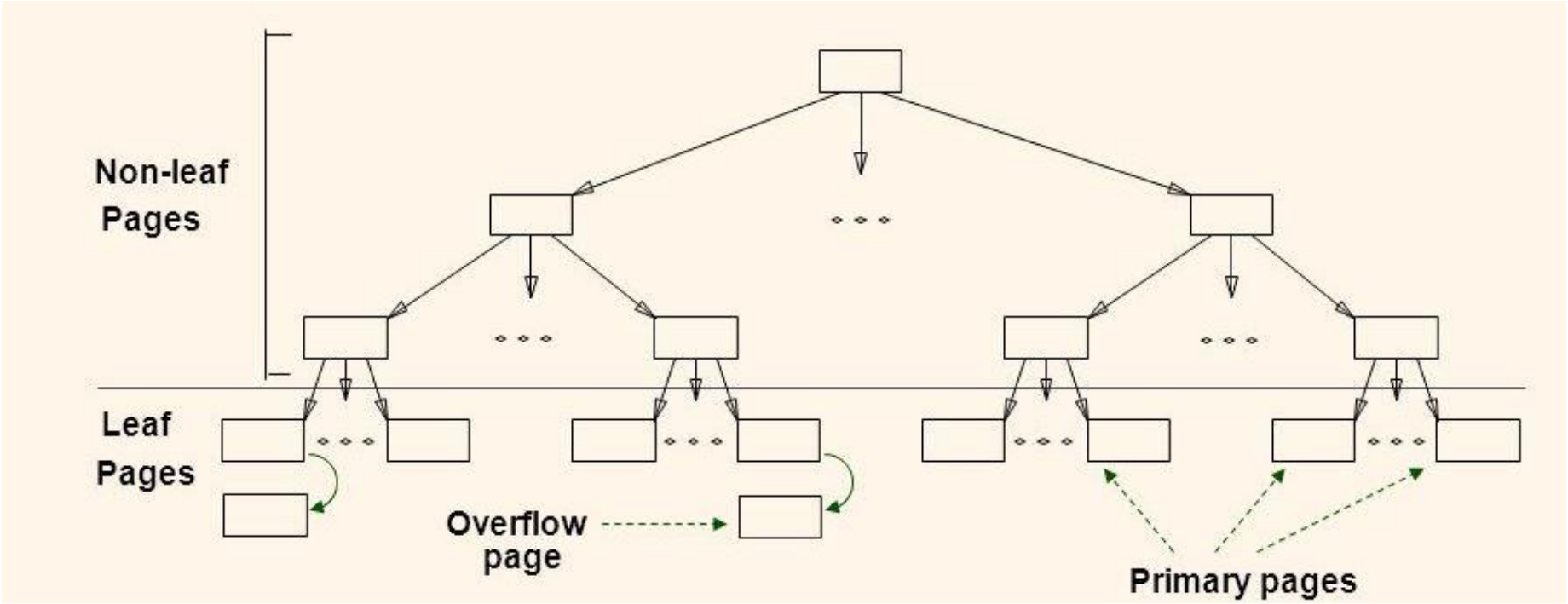
Range scan (αναζήτηση περιοχής ή διαστήματος τιμών)

- Συστάδων: #σελίδων στο αρχείο που ταιριάζουν
- Μη συστάδων: αριθμός εγγραφών στο ευρετήριο που ταιριάζουν – για κάθε τέτοια εγγραφή -> μια σελίδα αρχείου

ISAM

ISAM – Indexed Sequential Access Method
Στατικά ευρετήρια

Αλλάζουν μόνο οι δείκτες στα φύλλα
Όταν υπερχείλιση σε λίστες υπερχείλισης

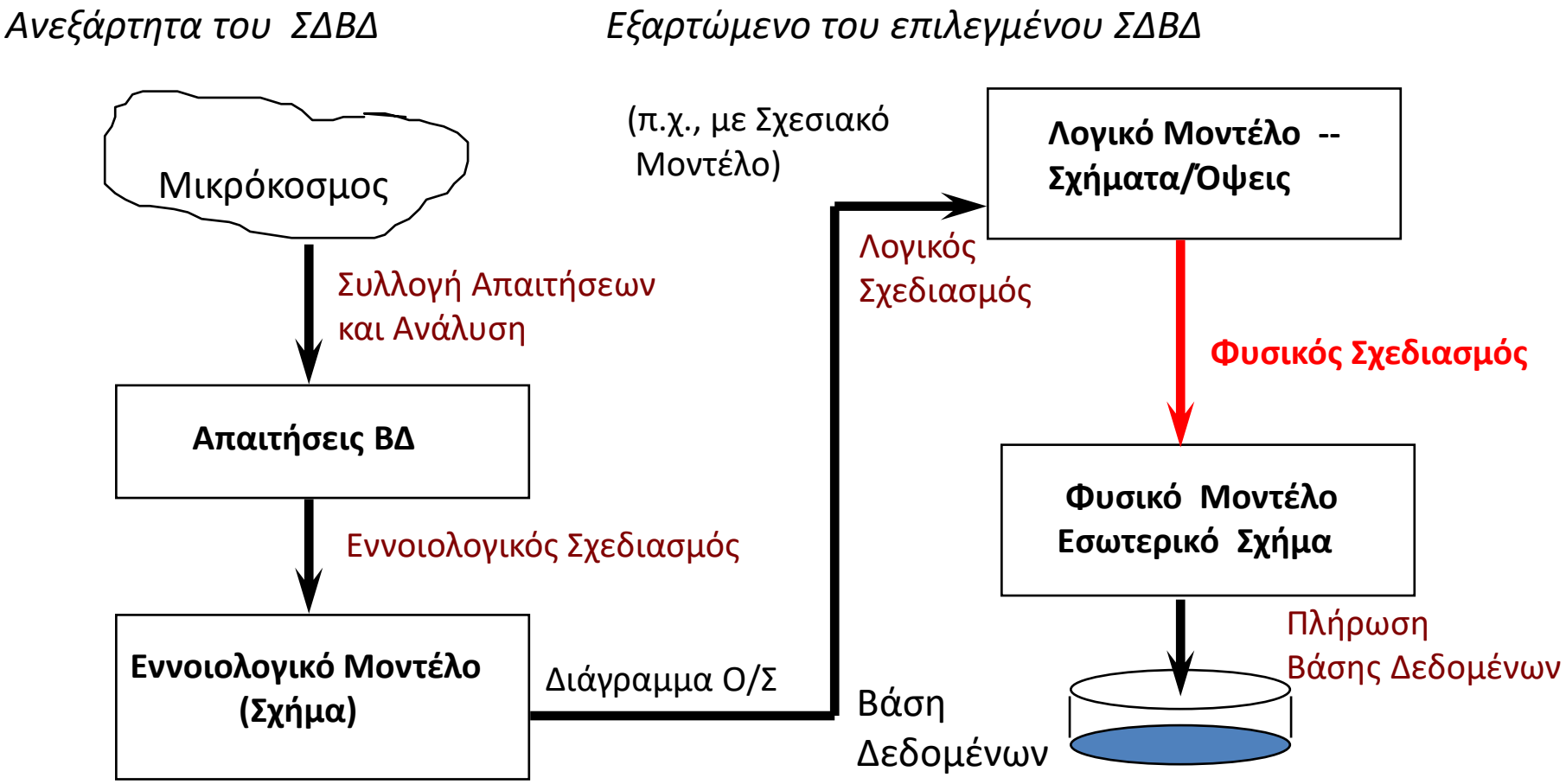


Φυσικός Σχεδιασμός

- Μετά τον σχεδιασμό Ο/Σ και το λογικό σχεδιασμό (σχεσιακό μοντέλο), έχουμε τα εννοιολογικά και λογικά (με τις όψεις) σχήματα για τη Βάση Δεδομένων.

- Το επόμενο βήμα είναι ο *φυσικός σχεδιασμός*, δηλαδή η επιλογή των δομών αποθήκευσης των σχέσεων, η επιλογή των ευρετηρίων, οι αποφάσεις για συστάδες - γενικά ότι είναι απαραίτητο για να επιτευχθούν οι προσδοκώμενες επιδόσεις χρήσης της ΒΔ.

Φυσικός Σχεδιασμός



Φυσικός Σχεδιασμός

Για να κάνουμε όσο το δυνατόν καλύτερο τον Φυσικό Σχεδιασμό πρέπει να κατανοήσουμε το φόρτο εργασίας (workload)

- Ποιες είναι οι σημαντικές ερωτήσεις και πόσο συχνά εμφανίζονται.
- Ποιες είναι οι πιο σημαντικές ενημερώσεις και πόσο συχνά εμφανίζονται.
- Ποια είναι η επιθυμητή επίδοση για την εκτέλεση αυτών των ερωτήσεων και ενημερώσεων.

Φυσικός Σχεδιασμός

Αποφάσεις που Απαιτούνται

Τι ευρετήρια πρέπει να δημιουργηθούν;

Ποιες σχέσεις πρέπει να έχουν ευρετήρια; Ποια γνωρίσματα χρησιμοποιούνται για αναζήτηση; Πρέπει να ορίσουμε πολλαπλά ευρετήρια;

Για κάθε ευρετήριο, τι είδους ευρετήριο πρέπει να είναι;

Συστάδες; Δέντρο/Κατακερματισμός; Δυναμικό/Στατικό; Πυκνό/Μη-πυκνό;

Χρειάζονται αλλαγές και στο εννοιολογικό/λογικό σχήμα;

Διαφορετικό κανονικοποιημένο σχήμα;

Denormalization (μήπως χρειάζεται από-κανονικοποίηση;)

Όψεις, Επανάληψη Δεδομένων (replication) …

Φυσικός Σχεδιασμός

Πριν δημιουργήσουμε ένα ευρετήριο, πρέπει να συνυπολογίσουμε και την επίδρασή του στις ενημερώσεις του φορτίου εργασίας!

Ένα ευρετήριο κάνει τις ερωτήσεις ΠΙΟ ΓΡΗΓΟΡΕΣ και τις ενημερώσεις ΠΙΟ ΑΡΓΕΣ

Επιπλέον, απαιτεί και χώρο στον δίσκο

Φυσικός Σχεδιασμός

Για *κάθε ερώτηση* (query) το φόρτο εργασίας:

- Σε ποιες σχέσεις έχει πρόσβαση?
- Ποια γνωρίσματα ανακαλεί?
- Ποια γνωρίσματα υπεισέρχονται στις συνθήκες για selection/join? Πόσο επιλεκτικές είναι αυτές οι συνθήκες?

Για *κάθε ενημέρωση* (insert/delete/update):

- Ποια γνωρίσματα υπεισέρχονται στις συνθήκες για selection/join? Πόσο επιλεκτικές είναι αυτές οι συνθήκες?
- Ο τύπος της ενημέρωσης (INSERT/DELETE/UPDATE), και τα γνωρίσματα που θα επηρεασθούν

Ευρετήρια στην SQL

Η SQL-92 *δεν* περιλαμβάνει εντολές για τη δημιουργία ευρετηρίων. Τα περισσότερα εμπορικά ΣΔΒΔ το υποστηρίζουν

create [unique] index <index_name>

on <table_name> (<attr_list>);

- Η <attr_list> μπορεί να περιέχει παραπάνω από ένα γνωρίσματα.
- Προαιρετικό UNIQUE σημαίνει ότι το <attr_list> είναι κλειδί του <table_name>.

drop index <index_name>

Ευρετήρια στην SQL

- Η Oracle δημιουργεί αυτόματα ευρετήρια για κάθε UNIQUE ή PRIMARY KEY ορισμό.
- Η MySQL δημιουργεί αυτόματα ευρετήρια για κάθε PRIMARY KEY ορισμό.

Oracle

select <index_name> **from user_indexes**

user_indexes είναι ένας πίνακας του συστήματος

MySQL

show index from .. where

Καθώς και αντίστοιχοι πίνακες συστήματος (INFORMATION_SCHEMA)

Άσκηση

Έστω ένας πίνακας (σχέση) CITY(Name, Population, Country) ο οποίος έχει πληροφορία για 150.000 πόλεις που είναι ομοιόμορφα κατανεμημένες σε 1.000 χώρες (δηλαδή 100 πόλεις ανά χώρα). Το Name είναι *κλειδί*. Ο πίνακας είναι αποθηκευμένος σε *ένα διατεταγμένο αρχείο* ως προς το *γνώρισμα* Name. Τα γνωρίσματα Name και Country έχουν μέγεθος 16 bytes, το γνώρισμα Population 32 bytes και ένα block (σελίδα) 2048 bytes. Υποθέστε ότι όλοι οι δείκτες έχουν μέγεθος 32 bytes.

Έστω ένα B+ δέντρο στο Population (θεωρείστε διακριτές τιμές για το population)

- Ποιο είναι το **μικρότερο** και το **μεγαλύτερο** ύψος
- Κόστος αναζήτησης με και χωρίς το B+-δέντρο

Τι αλλάζει αν το B+δέντρο είναι στο γνώρισμα Name;
Τι αλλάζει αν το B+δέντρο είναι στο γνώρισμα Country;

Αρχεία και ευρετήρια κατακερματισμού

Αρχεία Κατακερματισμού

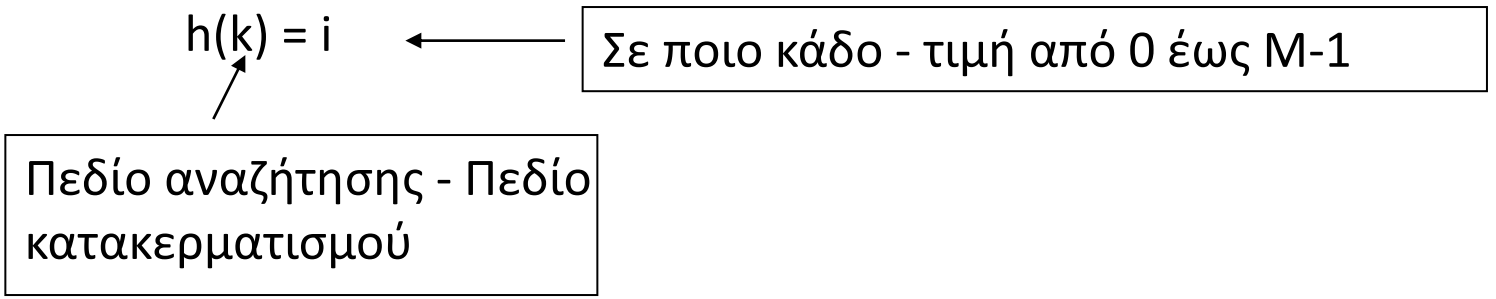
Βασική ιδέα: η τοποθέτηση των εγγραφών στα blocks του αρχείου γίνεται εφαρμόζοντας μια συνάρτηση κατακερματισμού σε κάποιο από τα πεδία της εγγραφής

Εσωτερικός Κατακερματισμός

Εσωτερικός Κατακερματισμός (τα δεδομένα είναι στη μνήμη, όπως στις δομές δεδομένων)

Πίνακας κατακερματισμού με Μ θέσεις - κάδους (buckets)

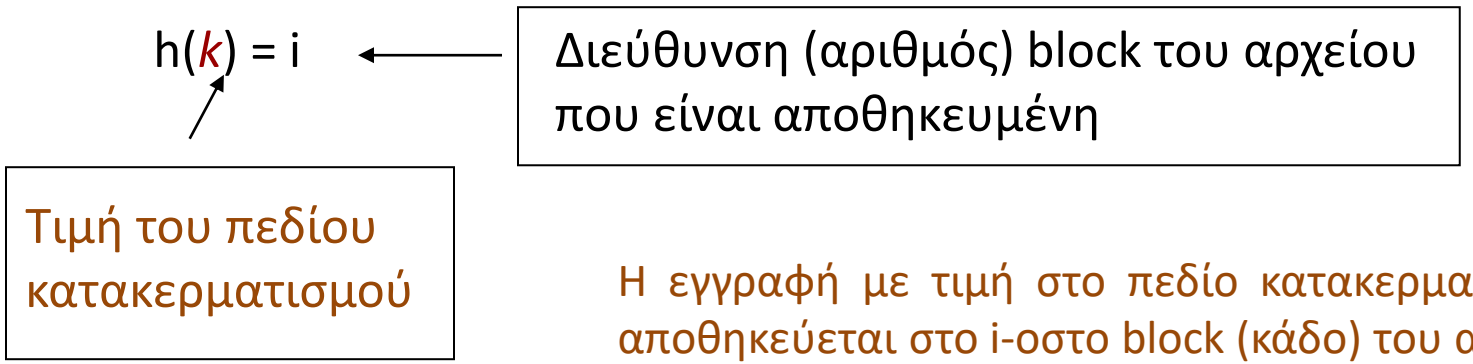
h: *συνάρτηση κατακερματισμού*



Αρχεία Κατακερματισμού

Εξωτερικός Κατακερματισμός (εφαρμογή σε δεδομένα αποθηκευμένα σε αρχεία)

Στόχος



Η εγγραφή με τιμή στο πεδίο κατακερματισμού *k* αποθηκεύεται στο i-οστο block (κάδο) του αρχείου

Κατακερματισμός

h: συνάρτηση κατακερματισμού

(Στόχος) Ομοιόμορφη κατανομή των κλειδιών στους κάδους (blocks)

- Συνηθισμένη συνάρτηση κατακερματισμού:

$$h(k) = k \bmod M$$

Κατακερματισμός

- Σύγκρουση (collision): όταν μια νέα εγγραφή κατακερματίζεται σε μία ήδη γεμάτη θέση

- Καλή συνάρτηση κατακερματισμού: κατανέμει τις εγγραφές ομοιόμορφα στο χώρο των διευθύνσεων (ελαχιστοποίηση συγκρούσεων και λίγες αχρησιμοποίητες θέσεις)

- Ευριστικοί:

  -- αν r εγγραφές, πρέπει να επιλέξουμε το M ώστε το r/M να είναι μεταξύ του 0.7 και 0.9

  -- όταν χρησιμοποιείται η mod τότε είναι καλύτερα το M να είναι πρώτος

Κατακερματισμός

Επίλυση Συγκρούσεων

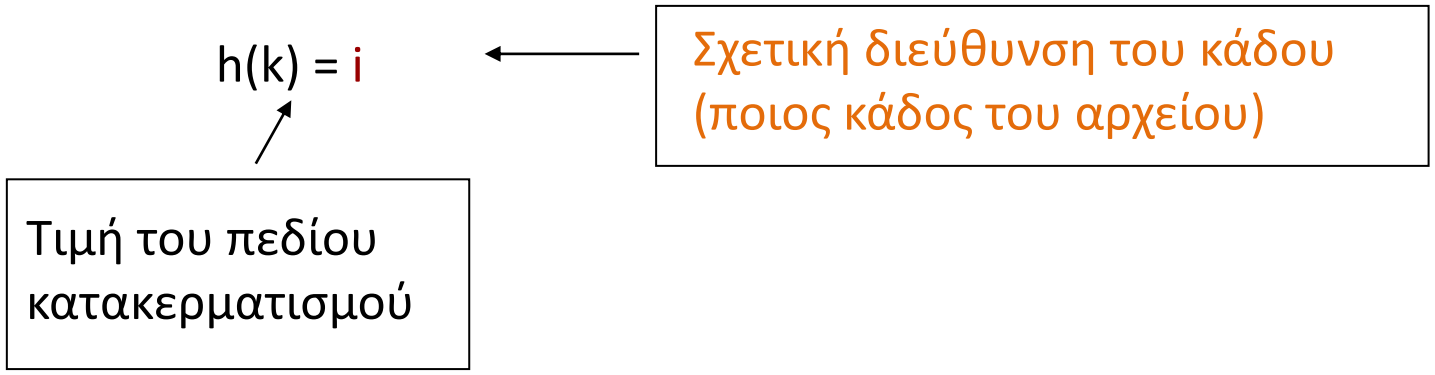
1. *Ανοιχτή Διευθυνσιοδότηση* (open addressing): χρησιμοποίησε την επόμενη κενή θέση

2. *Αλυσιδωτή Σύνδεση* (chaining): για κάθε θέση μια συνδεδεμένη λίστα με εγγραφές υπερχείλισης

3. *Πολλαπλός Κατακερματισμός* (multiple hashing): εφαρμογή μιας δεύτερης συνάρτησης κατακερματισμού

Εξωτερικός Κατακερματισμός

Κάδος: μια συστάδα από ένα ή περισσότερα συνεχόμενα blocks του αρχείου



Ο κατακερματισμός είναι πολύ αποδοτικός για *επιλογές (αναζητήσεις) ισότητας*

Εξωτερικός Κατακερματισμός

Ένας πίνακας που αποθηκεύεται στην επικεφαλίδα του αρχείου μετατρέπει τον αριθμό κάδου στην αντίστοιχη διεύθυνση block

0	διεύθυνση 1ου block του κάδου στο δίσκο
1	διεύθυνση 1ου block του κάδου στο δίσκο
2	διεύθυνση 1ου block του κάδου στο δίσκο
…	…
M-1	διεύθυνση 1ου block του κάδου στο δίσκο

Ευρετήρια Κατακερματισμού

Η ίδια ιδέα: αντί στους κάδους να έχουμε εγγραφές αρχείου (τις πλειάδες της σχέσης) έχουμε εγγραφές ευρετηρίου

Άσκηση

Έστω ένας πίνακας (σχέση) CITY(Name, Population, Country) ο οποίος έχει πληροφορία για 150.000 πόλεις που είναι ομοιόμορφα κατανεμημένες σε 1.000 χώρες (δηλαδή 100 πόλεις ανά χώρα). Το Name είναι κλειδί. Ο πίνακας είναι αποθηκευμένος σε ένα διατεταγμένο αρχείο ως προς το γνώρισμα Name. Τα γνωρίσματα Name και Country έχουν μέγεθος 16 bytes, το γνώρισμα Population 32 bytes και ένα block (σελίδα) 2048 bytes. Υποθέστε ότι όλοι οι δείκτες έχουν μέγεθος 32 bytes.

Έστω ένα **ευρετήριο κατακερματισμού** στο Population (θεωρείστε διακριτές τιμές για το population)
Θεωρείστε ότι κάθε κάδος έχει ένα block

Μέγεθος ευρετηρίου
Κόστος αναζήτησης με και χωρίς το ευρετήριο

Κατακερματισμός

Πρόβλημα στατικού κατακερματισμού:

Έστω M κάδους και r εγγραφές ανά κάδο - το πολύ M * r εγγραφές (αλλιώς μεγάλες αλυσίδες υπερχείλισης)

Δυναμικός κατακερματισμός

- Επεκτατός
- Γραμμικός

Δεν θα τον καλύψουμε φέτος

Ερωτήσεις;