

# *Εξωτερική Ταξινόμηση*

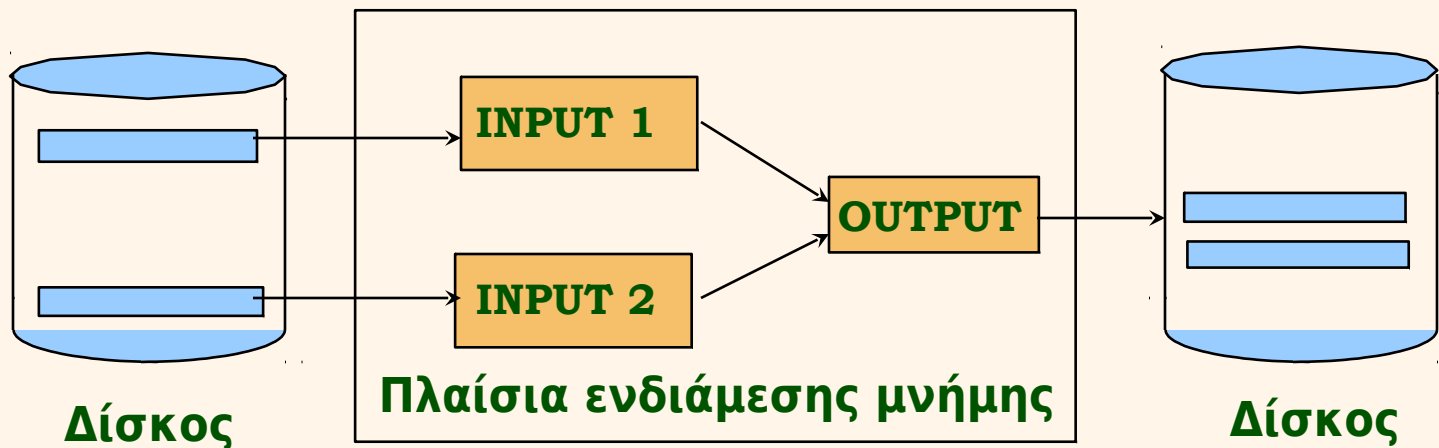
## Κεφάλαιο 13

# Τί χρειάζεται η Ταξινόμηση;

- ❖ Πρόκειται για κλασικό πρόβλημα της Επιστήμης Υπολογιστών!
- ❖ Τα δεδομένα πρέπει να είναι ταξινομημένα
  - π.χ., βρες τους φοιτητές με αλφαβητική σειρά
- ❖ Η ταξινόμηση είναι το πρώτο βήμα στη *μαζική φόρτωση* των B+tree.
- ❖ Με την ταξινόμηση μπορούμε να απαλλαγούμε από τις διπλότυπες εγγραφές (Γιατί;)
- ❖ Αλγόριθμος σύζευξης με *ταξινόμηση+συγχώνευση*.
- ❖ Πρόβλημα: ταξινόμησε 1Gb δεδομένων με 1Mb RAM
  - γιατί όχι με χρήση ιδεατής μνήμης;

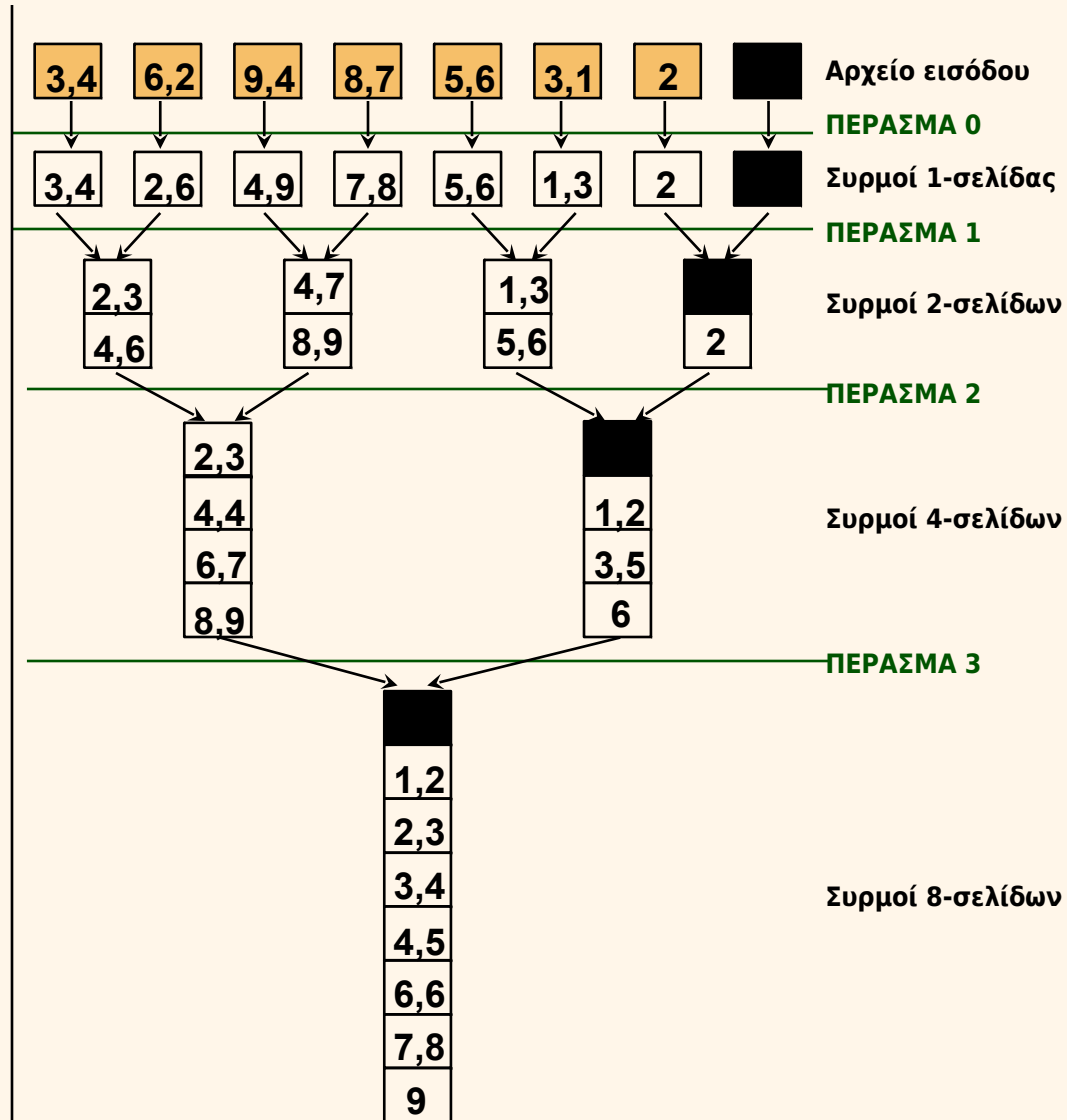
# Ταξ. 2-δρόμων: απαιτεί 3 πλαίσια

- ❖ Πέρασμα 1: Διάβασε μια σελίδα, ταξινόμησέ την, γράψ' την στο δίσκο.
  - χρειάζεται μόνο ένα πλαίσιο της ενδιάμεσης μνήμης
- ❖ Περάσματα 2, 3, ..., κλπ.:
  - χρειάζονται τρία πλαίσια



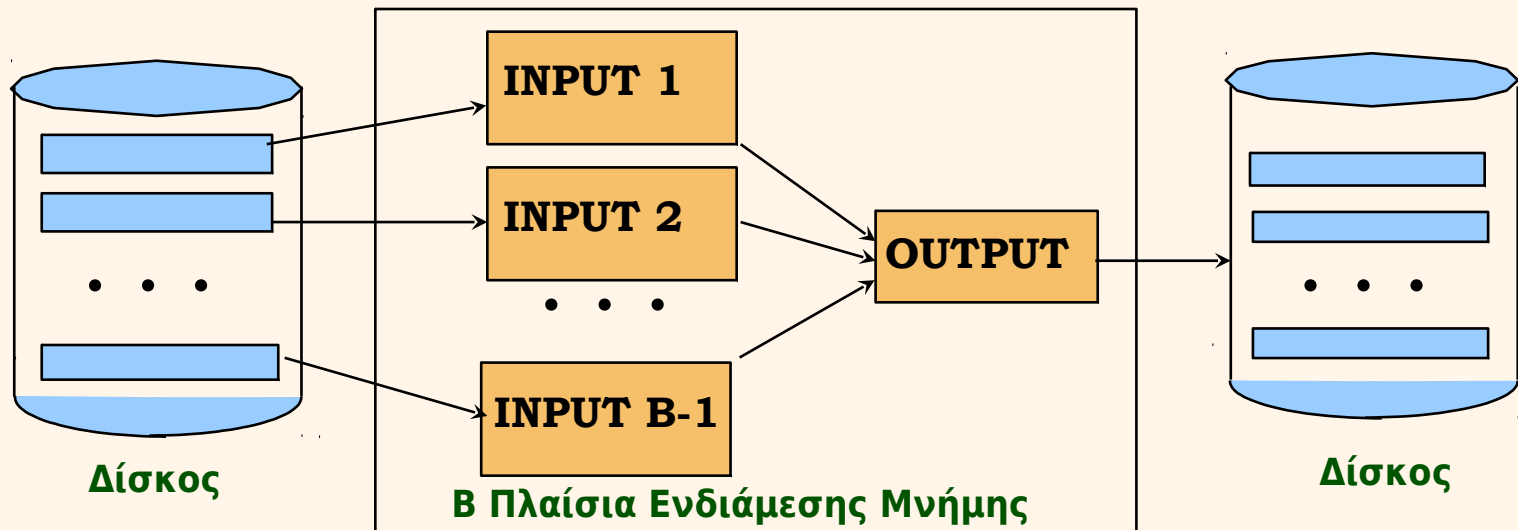
# Εξωτ. Ταξ. με Συγχώνευση 2-δρόμων

- ❖ Σε κάθε πέρασμα διαβάζουμε + γράφουμε καθεμιά σελίδα του αρχείου.
  - ❖  $N$  σελίδες στο αρχείο  $\Rightarrow$  πλήθος περασμάτων  $= \lceil \log_2 N \rceil + 1$
  - ❖ Άρα, συνολικό κόστος:  $2N(\lceil \log_2 N \rceil + 1)$
- Ιδέα: Διαίρει και Βασίλευε:** ταξινόμησε τα υπο-αρχεία και συγχώνευσε



# Γενική Εξωτερική Ταξ. με Συγχώνευση

- ❖ Περισσότερα των 3 πλαισίων. Πώς μπορούμε να τα χρησιμοποιήσουμε;
- ❖ Για να ταξινομήσουμε ένα αρχείο με  $N$  σελίδες χρησιμοποιώντας  $B$  πλαίσια ενδιάμεσης μνήμης:
  - Πέρασμα 0: χρησιμοποίησε  $B$  πλαίσια. Δημιούργησε  $\lceil N / B \rceil$  ταξινομημένους συρμούς των  $B$  σελίδων.
  - Πέρασμα 2, ..., κλπ.: συγχώνευσε  $B-1$  συρμούς.



# Κόστος Εξωτ. Ταξ. με Συγχώνευση

- ❖ Πλήθος περασμάτων:  $1 + \lceil \log_{B-1} \lceil N / B \rceil \rceil$
- ❖ Κόστος =  $2N * (\# \text{ περασμάτων})$
- ❖ Π.χ., με 5 πλαίσια, για να ταξινομηθεί ένα αρχείο 108 σελίδων:
  - Πέρασμα 0:  $\lceil 108 / 5 \rceil = 22$  ταξινομημένοι συρμοί των 5 σελίδων (ο τελευταίος έχει μόνο 3 σελίδες)
  - Πέρασμα 1:  $\lceil 22 / 4 \rceil = 6$  ταξινομημένοι συρμοί των 20 σελίδων (ο τελευταίος έχει μόνο 8 σελίδες)
  - Πέρασμα 2: 2 ταξινομημένοι συρμοί, 80 σελίδες και 28 σελίδες
  - Πέρασμα 3: Ταξινομημένο αρχείο 108 σελίδων

# Πλήθος Περασμάτων Εξ. Ταξινόμησης

N	B=3	B=5	B=9	B=17	B=129	B=257
100	7	4	3	2	1	1
1,000	10	5	4	3	2	2
10,000	13	7	5	4	2	2
100,000	17	9	6	5	3	3
1,000,000	20	10	7	5	3	3
10,000,000	23	12	8	6	4	3
100,000,000	26	14	9	7	4	4
1,000,000,000	30	15	10	8	5	4

# I/O Εξωτερικής Ταξ. με Συγχώνευση

- ❖ ... μεγαλύτεροι συρμοί, συνήθως, σημαίνει λιγότερα περάσματα!
- ❖ Θεωρήσαμε I/O με μια σελίδα τη φορά
- ❖ Πρακτικά, διάβασε ένα μπλοκ σελίδων σειριακά!
- ❖ Άρα, θα μπορούσε κάθε πλαίσιο (εισόδου/εξόδου) να είναι ένα μπλοκ σελίδων.
  - Αλλά, κάτι τέτοιο θα μείωνε το πλήθος των συρμών κατά τα περάσματα συγχώνευσης!
  - Παρόλα αυτά, στην πράξη, τα περισσότερα αρχεία ταξινομούνται με 2-3 περάσματα.



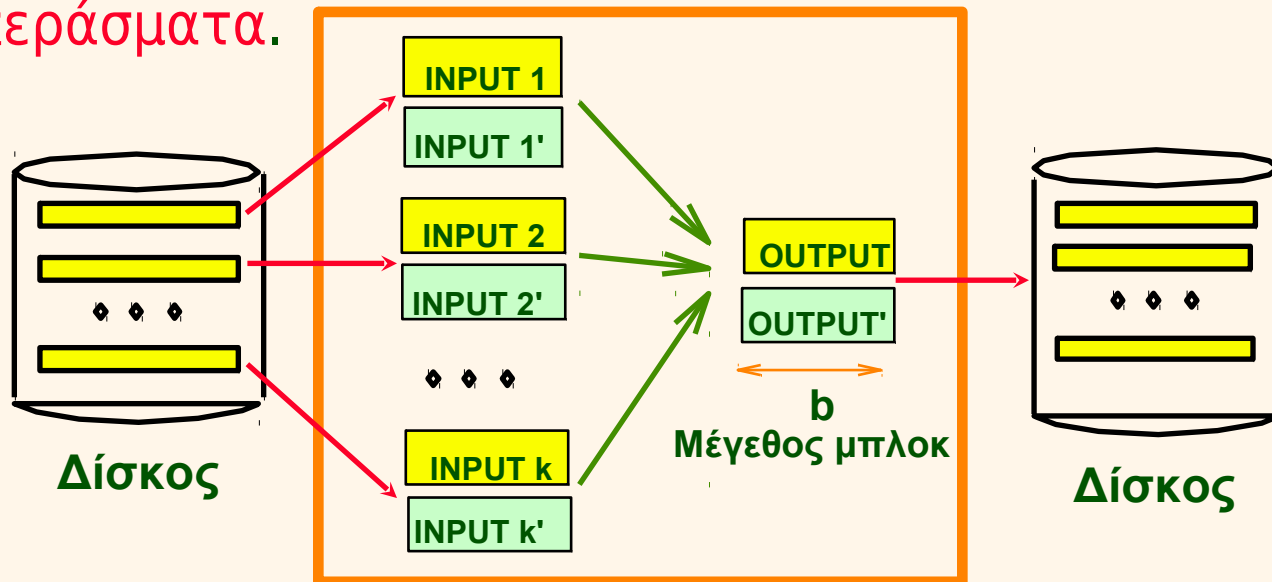
# Πλήθος Περασμάτων της Βελτιστοποιημένης Ταξινόμησης

N	B=1,000	B=5,000	B=10,000
100	1	1	1
1,000	1	1	1
10,000	2	2	1
100,000	3	2	2
1,000,000	3	2	2
10,000,000	4	3	3
100,000,000	5	3	3
1,000,000,000	5	4	3

☛ Μέγεθος μπλοκ = 32, αρχικό πέρασμα δημιουργεί συρμούς μεγέθους 2B.

# Διπλά Πλαίσια

- ❖ Για να μειώσουμε το χρόνο αναμονής για την ολοκλήρωση των αιτήσεων I/O, μπορούμε να **προφορτώνουμε** στο σκιώδες μπλοκ.
  - Θα έχουμε περισσότερα περάσματα, όμως στην πράξη, τα περισσότερα αρχεία **θα εξακολουθούν** να ταξινομούνται σε 2-3 περάσματα.



**B** πλαίσια ενδιάμεσης μνήμης, συγχώνευση k-δρόμων

# Ταξινομώντας εγγραφές!

- ❖ Η ταξινόμηση έχει γίνει αγώνας δρόμου!
  - Παράλληλη ταξινόμηση είναι η απάντηση ...
- ❖ Datamation: Ταξινόμησε 1M εγγραφές των 100 bytes
  - Συνηθισμένο DBMS: 15 λεπτά
  - Παγκόσμιο ρεκόρ (2004): 3.5 **δευτερόλεπτα**
    - 12-CPU SGI υπολογιστής, 96 δίσκοι, 2GB RAM
  - 2011: 40ms (cluster 16 nodes: i7, 12GB RAM, 6 Disks 1TB)
- ❖ Νέα τεστ αναφοράς:
  - Ταξινόμηση Λεπτού: Πόσες εγγραφές ταξινομούνται σε 1 λεπτό;
  - Ταξινόμηση Δολαρίου: Πόσες εγγραφές ταξινομούνται με \$1.00;

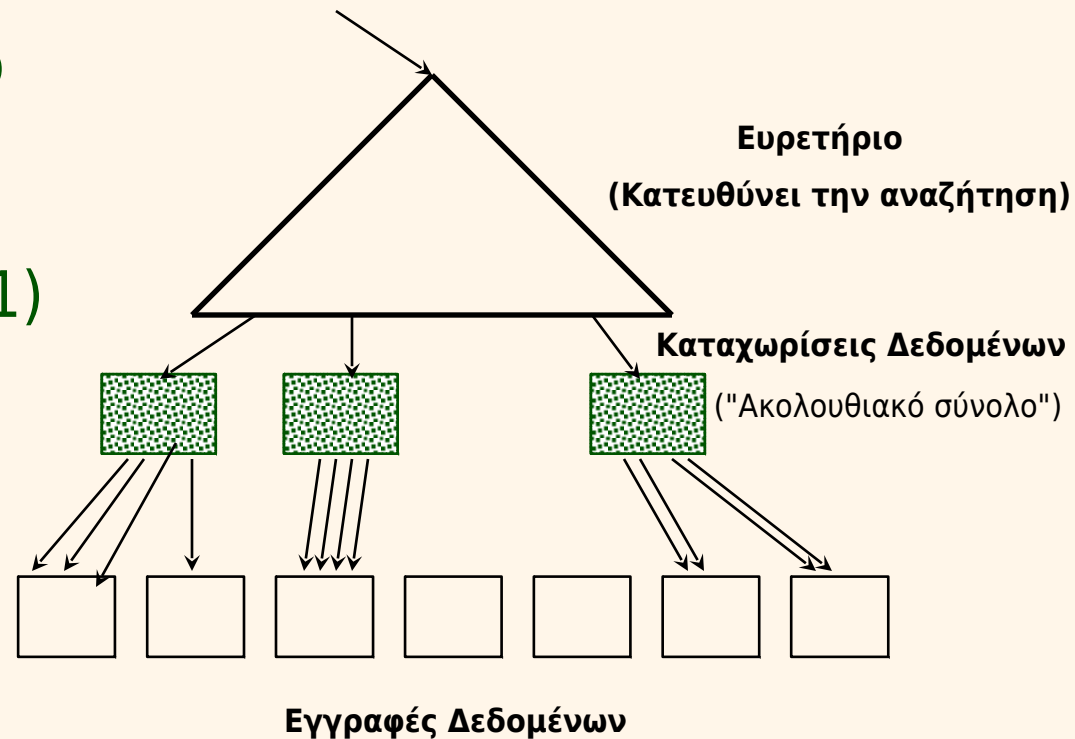
# Χρησιμοποιώντας B+Tree για Ταξινόμηση

- ❖ Σενάριο: Ο προς ταξινόμηση Πίνακας έχει ένα B+tree στα πεδία ταξινόμησης.
- ❖ **Ιδέα:** Μπορούμε να ανακτήσουμε τις εγγραφές με τη σειρά διασχίζοντας τα φύλλα.
- ❖ **Είναι καλή ιδέα;**
- ❖ Τί να προσέξουμε:
  - B+tree **συγκροτημένο** **Καλή ιδέα!**
  - B+ tree **μη-συγκροτημένο** **Θα μπορούσε να είναι μια πολύ κακή ιδέα!**

# Συγκροτημένα B+Tree για Ταξινόμηση

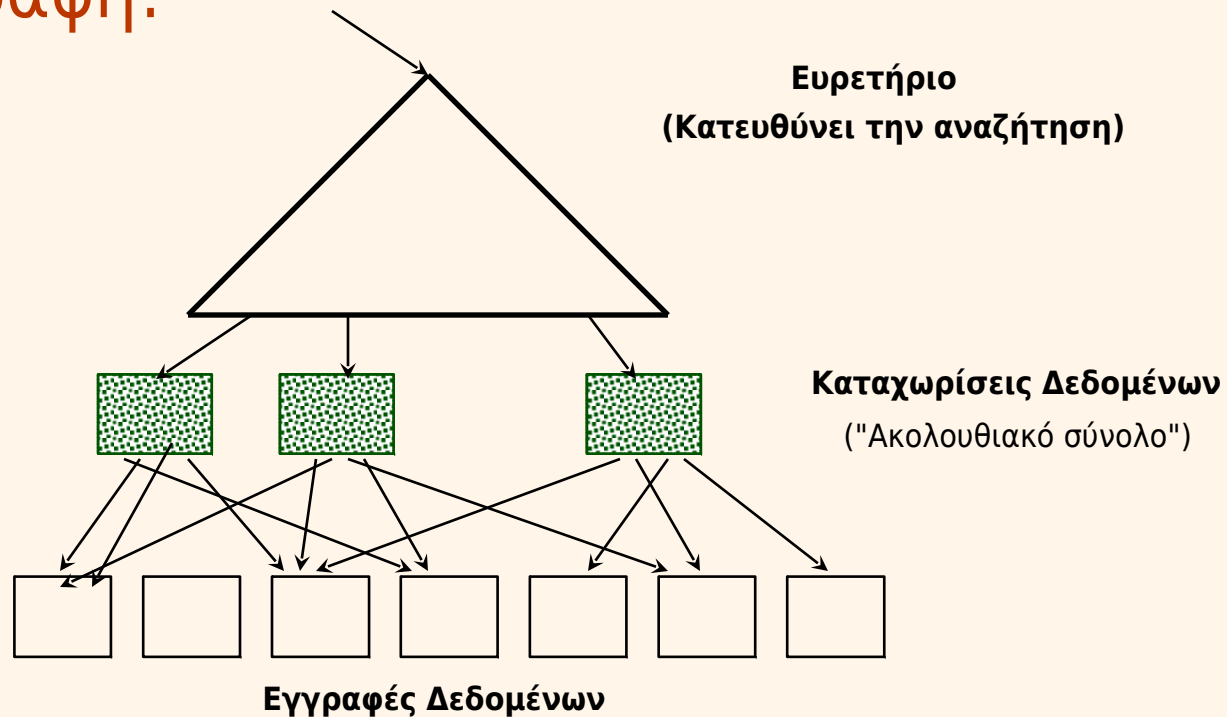
- ❖ Κόστος: από τη ρίζα μέχρι το αριστερότερο φύλλο, και μετά ανάκτηση όλων των φύλλων (Εναλλακτική 1)

- ❖ Τί συμβαίνει αν χρησιμοποιείται η Εναλλακτική 2; Επιπλέον κόστος για την ανάκτηση των εγγραφών: κάθε σελίδα χρειάζεται να ανακτηθεί μόνο μια φορά. ➡ **Πάντα καλύτερο από την εξωτερική ταξινόμηση!**



# Μη-Συγκροτημένα B+Tree για Ταξινόμηση

- ❖ Εναλλακτική (2) για τις καταχωρίσεις δεδομένων – κάθε καταχώριση δεδομένων περιέχει το *rid* μιας εγγραφής. Γενικά, **μια I/O ανά εγγραφή!**



# Εξωτ. Ταξ. vs. Μη-Συγκροτημένου Ευρ.

N	Sorting	p=1	p=10	p=100
100	200	100	1,000	10,000
1,000	2,000	1,000	10,000	100,000
10,000	40,000	10,000	100,000	1,000,000
100,000	600,000	100,000	1,000,000	10,000,000
1,000,000	8,000,000	1,000,000	10,000,000	100,000,000
10,000,000	80,000,000	10,000,000	100,000,000	1,000,000,000

- $p$ : # εγγραφών ανά σελίδα
- $B=1,000$  και μέγεθος μπλοκ=32 για την ταξινόμηση
- $p=100$  είναι μια πιο ρεαλιστική τιμή

# Σύνοψη

- ❖ Η Εξωτερική ταξινόμηση είναι σημαντική – τα DBMS μπορεί να δεσμεύουν τμήμα της ενδιάμεσης μνήμης για την ταξινόμηση!
- ❖ Η Εξωτερική ταξινόμηση με συγχώνευση ελαχιστοποιεί το κόστος I/O:
  - Πέρασμα 0: Παράγει ταξινομημένους **συρμούς** μεγέθους  **$B$**  (# πλαισίων). Επόμενα περάσματα: **συγχωνεύουν** τους συρμούς.
  - # συρμών που συγχωνεύονται κάθε φορά: εξαρτάται από το  **$B$** , και το **μέγεθος του μπλοκ**.
  - Μεγαλύτερο μέγεθος μπλοκ σημαίνει μικρότερο κόστος I/O ανά σελίδα.
  - Μεγαλύτερο μέγεθος μπλοκ σημαίνει μικρότερο # συρμών που συγχωνεύονται.
  - Στην πράξη, το # συρμών σπάνια είναι μεγαλύτερο από 2 ή 3.



# Σύνοψη (συνέχεια)

- ❖ Η επιλογή του αλγόριθμου εσωτερικής ταξινόμησης μπορεί να παίζει ρόλο:
  - Quicksort: πολύ γρήγορος!
  - Heap/tournament sort: πιο αργός (2x), μεγαλύτεροι συρμοί
- ❖ Οι καλύτερες ταξινομήσεις είναι πολύ γρήγορες:
  - Παρά τα 40+ χρόνια έρευνας, ακόμα βελτιώνονται!
- ❖ Το Συγκροτημένο B+tree κάνει για ταξινόμηση – το μη-συγκροτημένο συνήθως είναι πολύ κακό.