



Advanced Data Indexing

(Προηγμένη ευρετηρίαση δεδομένων)

Αναζήτηση – Δέντρα (1^ο Μέρος)



Αναζήτηση (Searching)

Η Αναζήτηση

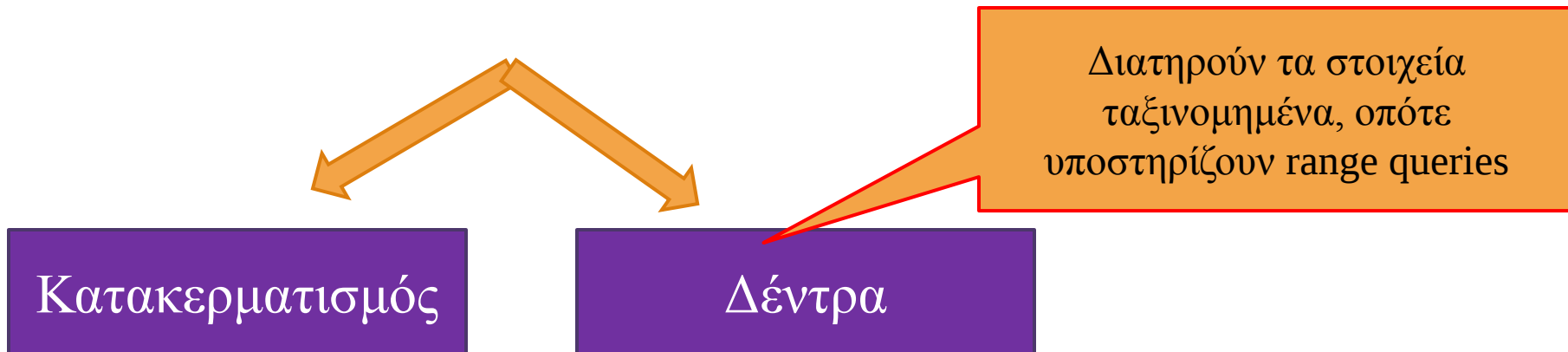
- **Searching (Αναζήτηση)** ενός αντικειμένου μέσα από N **ταξινομημένα** αντικείμενα.
- Είναι στοιχειώδης διαδικασία, πολύ σημαντική σχεδόν σε όλα τα προβλήματα που αφορούν δεδομένα.
- Προϋποθέτει τα αντικείμενα να είναι ταξινομημένα αν θέλουμε να έχουμε την βέλτιστη πολυπλοκότητα:
- $\text{Search}(N) = \Theta(\log_B N) \rightarrow$ Στην κύρια μνήμη ή ($D=1$)
- $\text{Search}(N) = \Theta(\log_{DB} N) \rightarrow$ Στο μοντέλο PDM ($D>1$)

Η Αναζήτηση απαιτεί Δομές

- Για να υποστηριχθεί η αποδοτική αναζήτηση (λογαριθμική πολυπλοκότητα) απαιτείται η χρήση δομών-ευρετηρίων (**indexing**) πάνω στα ταξινομημένα δεδομένα.
- Οι δομές αυτές πρέπει να είναι κατάλληλες ώστε οι εξής 3 πράξεις να υποστηρίζονται αποδοτικά:
 1. **Εύρεση (find)** ενός στοιχείου στο σύνολο.
 2. **Ένθεση (insert)** ενός νέου στοιχείου στο σύνολο.
 3. **Διαγραφή (delete)** ενός υπάρχοντος στοιχείου από το σύνολο.
- Απαραίτητη προϋπόθεση για τις πράξεις insert, delete είναι τα δεδομένα μετά από αυτές **να παραμένουν ταξινομημένα**.

Τι είδους Δομές;

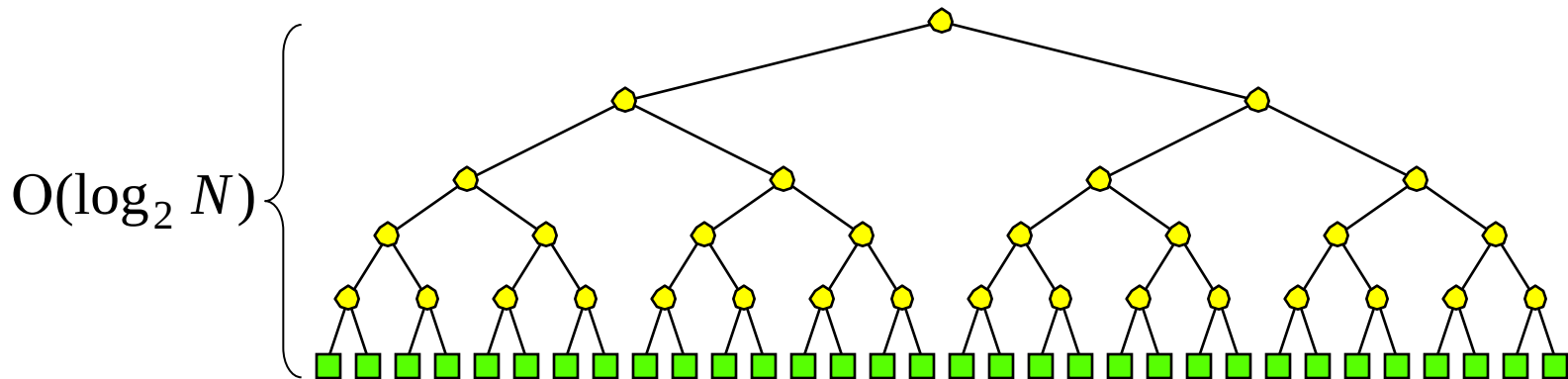
- Πολλές φορές απαιτείται ο σχεδιασμός δομών ανάλογα:
 1. **Με το είδος των δεδομένων** [π.χ. αριθμητικά, κειμένου (λεξικά), χωρικά, πολυμεσικά, κλπ.]
 2. **Με το είδος των ερωτημάτων** που θέλουμε να υποστηριχθούν [π.χ. simple find queries, nearest-neighbor queries, range queries, top-k queries, κλπ.]



Δέντρα Αναζήτησης Εσωτερικής Μνήμης

Δυαδικό Δέντρο Αναζήτησης (Binary Tree):

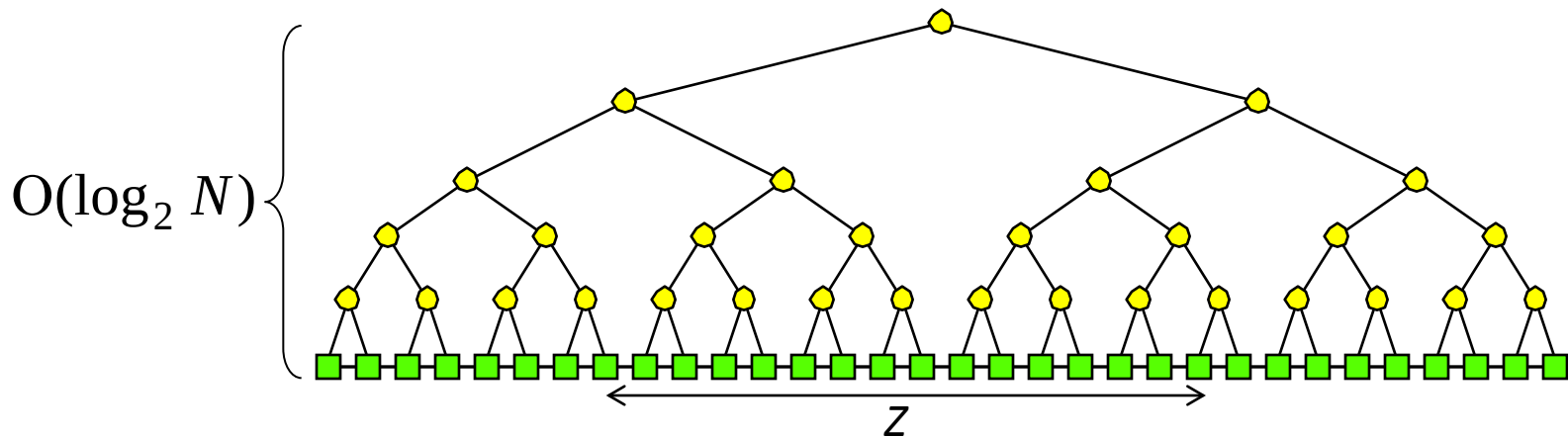
- Υποθέτουμε ότι τα στοιχεία είναι στα φύλλα.
- Η εύρεση (απλή αναζήτηση ενός στοιχείου) γίνεται διασχίζοντας ένα μονοπάτι από τη ρίζα μέχρι το φύλλο που βρίσκεται:



- Συνεπώς η εύρεση γίνεται σε $O(\log_2 N)$ χρόνο.
- Το δέντρο όμως πρέπει να είναι **ισοζυγισμένο**.
- Για να διατηρηθεί έτσι χρειάζονται ειδικοί αλγόριθμοι κατά την εισαγωγή και την διαγραφή στοιχείων.

Δέντρα Αναζήτησης Εσωτερικής Μνήμης

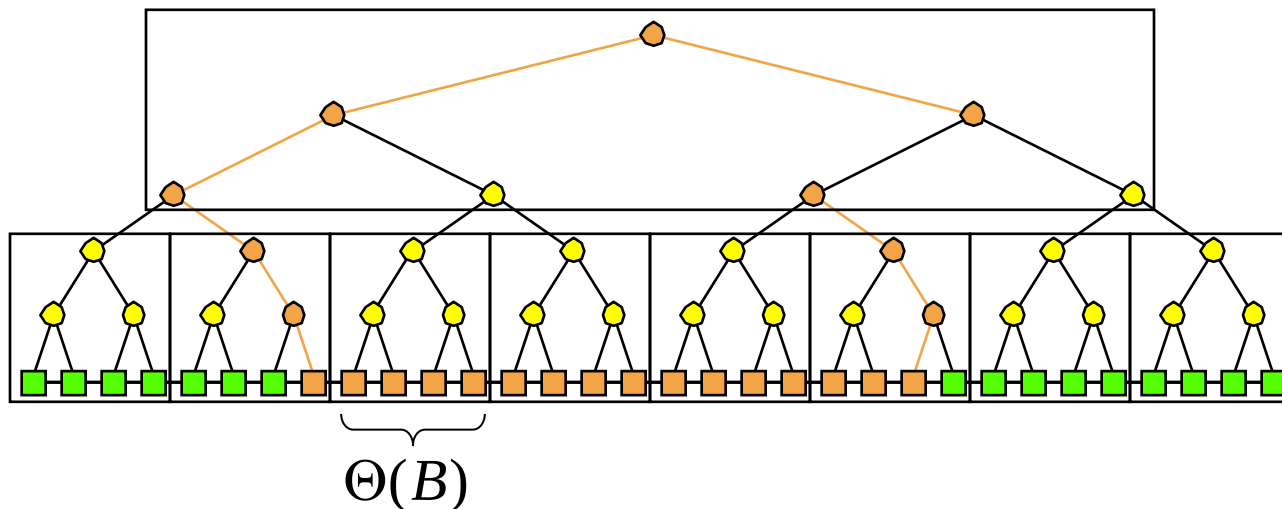
- Αν τα στοιχεία στα φύλλα είναι και μεταξύ τους συνδεδεμένα, τότε μπορούμε να υποστηρίξουμε εύρεση επομένου ή προηγούμενου καθώς και ερωτήματα περιοχής (**range queries**):



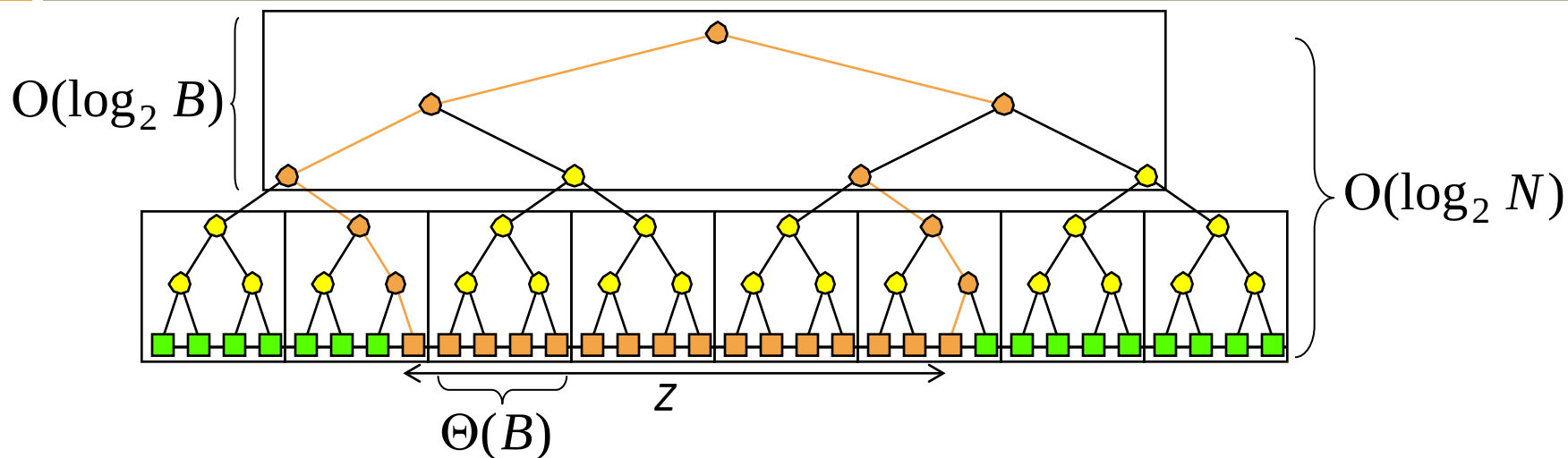
- Γίνεται η εύρεση των δύο άκρων με απλή αναζήτηση.
- Διασχίζουμε την συνδεδεμένη λίστα των φύλλων ανάμεσα στα δύο άκρα. Συνολικό κόστος: $O(Z + 2\log_2 N) = O(Z + \log_2 N)$

Δένδρα Αναζήτησης Εξωτερικής Μνήμης

- Το δέντρο είναι αποθηκευμένο σε πλήθος μπλοκ μεγέθους B .
- Στο μέγεθος ενός μπλοκ B χωράει ένα σταθερό πλήθος κόμβων (είτε εσωτερικών είτε φύλλων).
- Για να εκμεταλλευτούμε την **τοπικότητα** θα πρέπει στο ίδιο μπλοκ να είναι αποθηκευμένοι γειτονικοί κόμβοι του δέντρου.
- Εφόσον διαβάζονται και γράφονται ολόκληρα μπλοκ, στο κόστος μετράει και το ύψος του δέντρου ως προς τα μπλοκ.



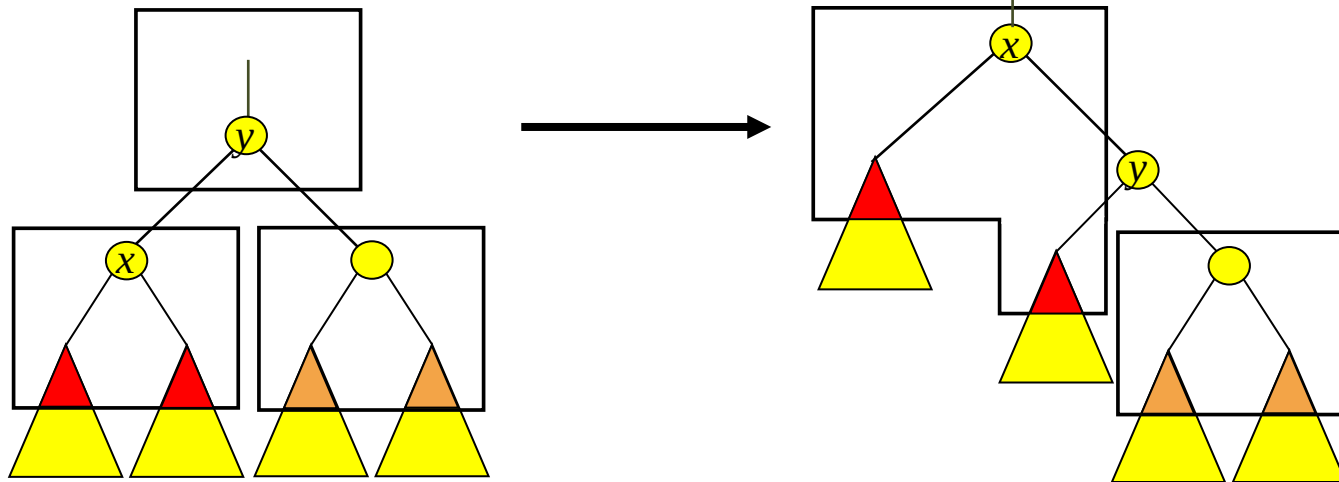
Δένδρα Αναζήτησης Εξωτερικής Μνήμης



- Το ύψος του δέντρου ως προς τα μπλοκ είναι:
 $O(\log_2 N)/O(\log_2 B) = O(\log_B N)$
- Τα κλειδιά των Z αντικειμένων που επιστρέφονται ως απάντηση σε ένα ερώτημα εξάγονται επίσης σε πλήθος μπλοκ.
- I/O κόστος ερωτημάτων αναζήτησης (search): **$O(\log_B N)$** [Βέλτιστο]
- I/O κόστος ερωτημάτων περιοχής (range): **$O(\log_B N + Z/B)$** [Βέλτιστο]
- Απαιτούμενος χώρος (ισοζυγισμένο): **$O(N/B)$** [Βέλτιστος]

Δένδρα Αναζήτησης Εξωτερικής Μνήμης

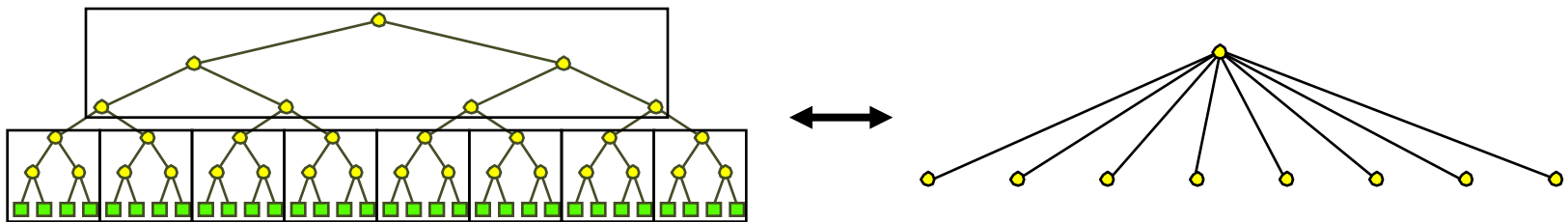
- Για να διατηρηθεί η **τοπικότητα** πρέπει να διατηρηθεί η σειρά των κατά πλάτους μπλοκ (BFS) σε ενημερώσεις (**insert, delete, update**)
- Μία τεχνική είναι η επαναζύγισση με περιστροφές:



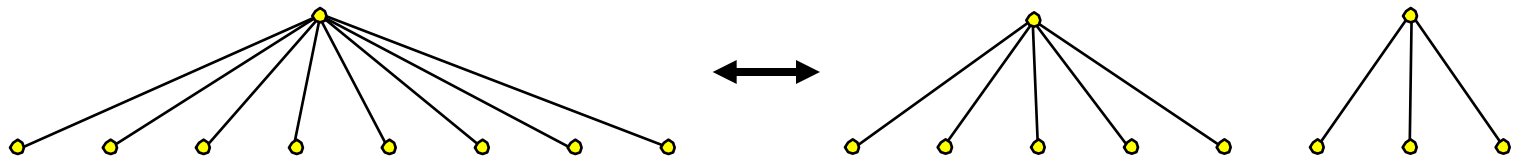
- Φαίνεται δύσκολο να διατηρηθεί το BFS κατά την περιστροφή!
- Αν μάλιστα έχουμε συχνές ενημερώσεις που προκαλούν επαναζυγίσεις σε μεγάλο ποσοστό, συμφέρει καλύτερα να κτίσουμε όλο το δέντρο από την αρχή! (**Bulkload**)

Δένδρα Αναζήτησης Εξωτερικής Μνήμης

- Επίσης συμφέρει τα μπλοκ κατά BFS να αντιστοιχούν σε δέντρο με βαθμό $\Theta(B) \rightarrow B$ -Δέντρα

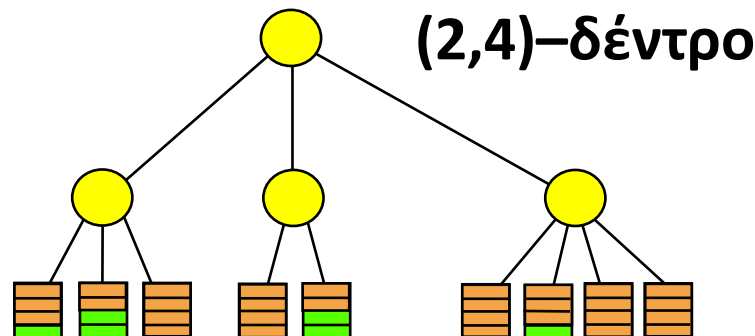


- Αν επιτρέψουμε την **διακύμανση του βαθμού** θα πετύχουμε πιο αποδοτική επαναζύγιση (διάσπαση, συγχώνευση, διαμοίραση) $\rightarrow (\alpha, b)$ -Δέντρα



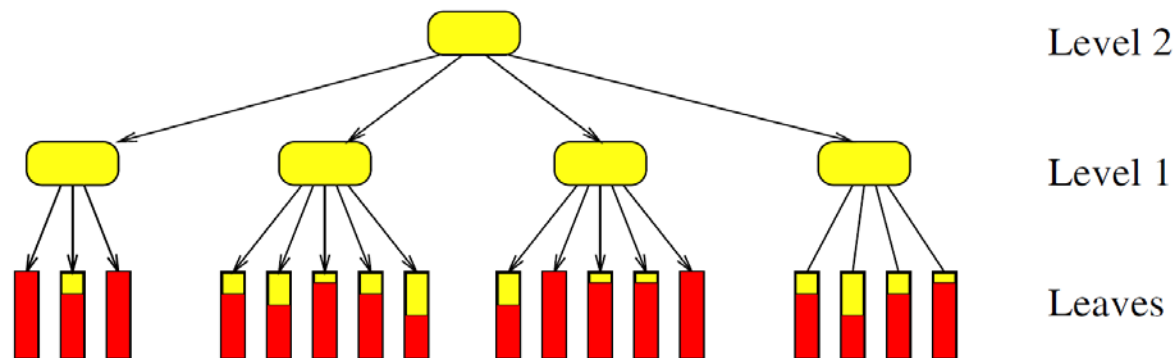
(a,b) -Δέντρα

- Τα **B-Δέντρα** είναι ειδική περίπτωση μίας γενικότερης ομάδας δέντρων τα **(a,b) -Δέντρα**.
- Ένα Δέντρο **T** είναι **(a,b) -Δέντρο** ($a \geq 2$ και $b \geq 2a-1$) όταν:
 - ▣ Όλα τα **φύλλα** του T είναι στο **ίδιο επίπεδο** και περιέχουν από **a** έως **b** στοιχεία ($min=a, max=b$)
 - ▣ Εκτός της ρίζας, όλοι οι κόμβοι έχουν **βαθμό** μεταξύ **a** και **b**
 - ▣ Η **ρίζα** έχει βαθμό μεταξύ **2** και **b** και περιέχει από **1** ως **$b-1$** στοιχεία.



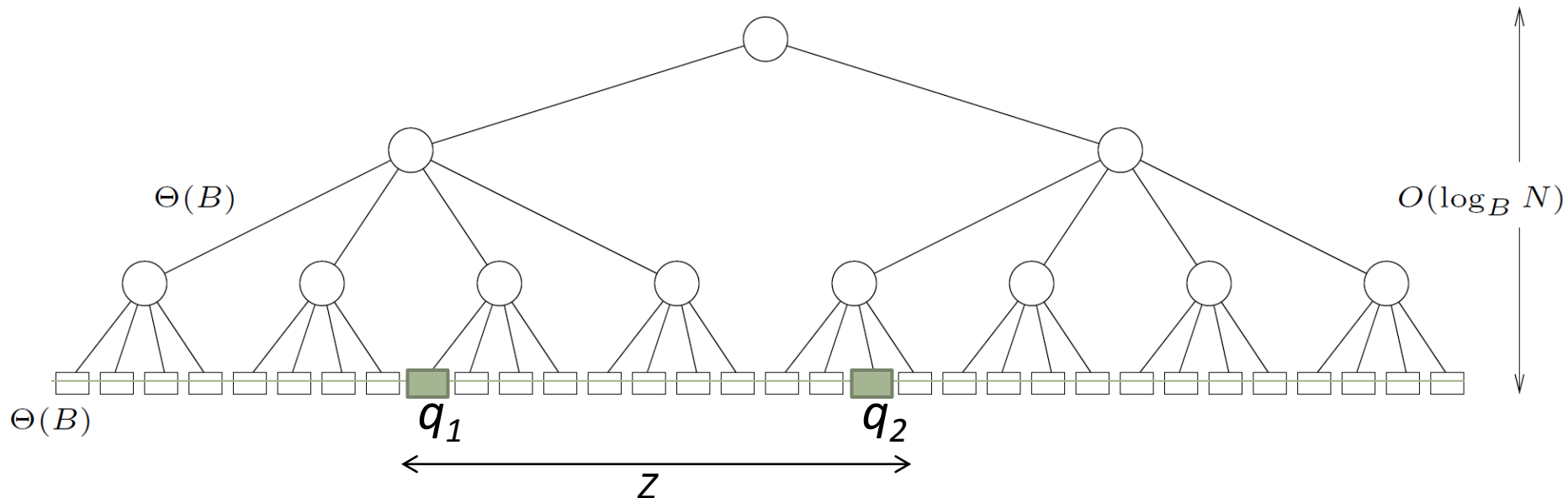
(a,b) -Δέντρα

- Συνήθως τα N στοιχεία είναι αποθηκευμένα μόνο στα φύλλα, ενώ οι εσωτερικοί κόμβοι χρησιμοποιούνται βοηθητικά για να καθοδηγούν τα μονοπάτια των αναζητήσεων.
- Έτσι το (a,b) -δέντρο χρησιμοποιεί **γραμμικό χώρο**: $O(N/B)$ και έχει ύψος $O(\log_a N)$.
- Αν επιλέξουμε $a,b = \Theta(B)$ [B -Δέντρα] τότε θα έχουμε ύψος ως προς μπλοκ: $O(\log_a N)/O(\log_a B) = O(\log_B N)$. Συνεπώς:
- **I/O κόστος ερωτημάτων αναζήτησης (search): $O(\log_B N)$**



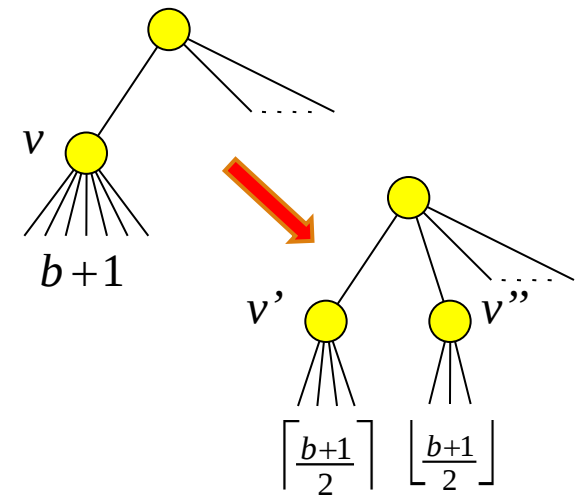
(α, b) -Δέντρα

- Στα ερωτήματα περιοχής (διάστημα $[q_1, q_2]$) εντοπίζονται τα άκρα q_1, q_2 σε $O(\log_B N)$ και διασχίζονται τα φύλλα που πάλι είναι συνδεδεμένα μεταξύ τους σε σειρά. Αν Z τα στοιχεία στο $[q_1, q_2]$:
- I/O κόστος ερωτημάτων περιοχής: **$O(\log_B N + Z/B)$**

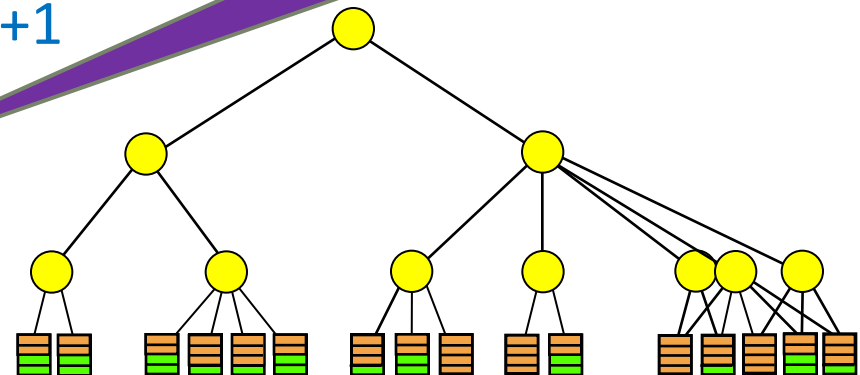


Ένθεση ενός Στοιχείου

- Βρίσκεται και **εισάγεται** το νέο στοιχείο στο φύλλο που αντιστοιχεί (v)
- Αν το v φτάσει τα $b+1$ στοιχεία/παιδιά τότε **διασπάται** (split) σε δύο νέα φύλλα v' , v'' με $\lceil \frac{b+1}{2} \rceil \leq b$ και $\lfloor \frac{b+1}{2} \rfloor \geq a$ στοιχεία.
- Στον πατέρα του v **διαγράφεται** ο δείκτης για τον v και **προστίθενται** οι δείκτες για τους v' , v'' .
- Αν και ο πατέρας του v φτάσει τα $b+1$ στοιχεία διασπάται ομοίως.
- Συνεχίζεται ο έλεγχος **αναδρομικά** προς τα πάνω μέχρι και την ρίζα (αν χρειαστεί).

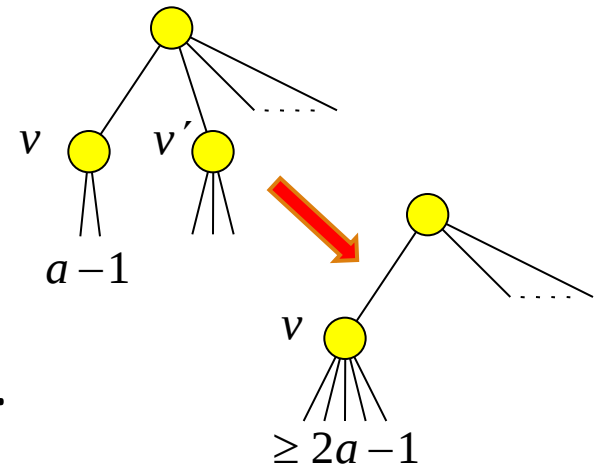


Κόστος: $O(\log_\alpha N)$

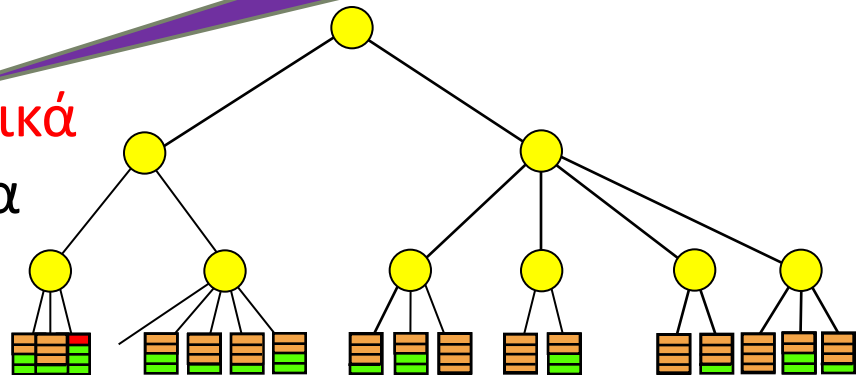


Διαγραφή ενός Στοιχείου

- Βρίσκεται και **διαγράφεται** το στοιχείο από το φύλλο που αντιστοιχεί (v)
- Αν το v φτάσει τα $a-1$ στοιχεία/παιδιά τότε **συγχωνεύεται** (fuse) με έναν αδερφό του v' .
- **Μετακινούνται** όλα τα στοιχεία του v' στον v .
- **Διαγράφεται** ο δείκτης για τον v' .
- Αν ο v έχει $>b$ (και $<a-1+b<2b$) στοιχεία/παιδιά τότε διασπάται (split).
- Συνεχίζεται ο έλεγχος **αναδρομικά** προς τα πάνω μέχρι και την ρίζα (αν χρειαστεί).

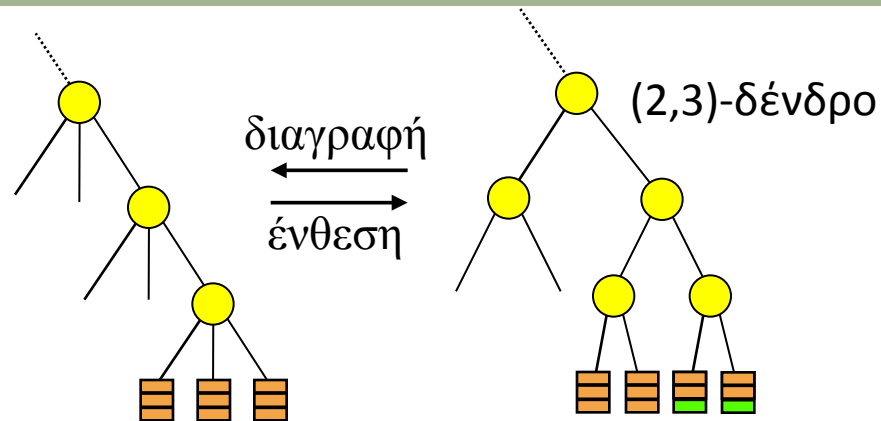


Κόστος: $O(\log_a N)$



Άλλες Ιδιότητες των (a,b) -Δέντρων

- Αν **$b=2a-1$** κάθε ενημέρωση προκαλεί πολλές επαναζυγίσεις: **$O(\log_a N)$**



- Το χειρότερο αυτό κόστος επιτυγχάνεται αν γίνεται εισαγωγή και διαγραφή του ίδιου στοιχείου διαδοχικά.
- Αν $b \geq 2a$ οι ενημερώσεις προκαλούν επιμερισμένες επαναζυγίσεις: $O(\frac{1}{a} \log_a N)$

B-Δέντρα

- **B-Δέντρα** είναι απλά (a,b) -δέντρα όπου $a,b = \Theta(B)$.
 - Απαιτούμενος χώρος: $O(N/B)$
 - I/O κόστος ερωτημάτων αναζήτησης (search): $O(\log_B N)$
 - I/O κόστος ερωτημάτων περιοχής (range): $O(\log_B N + Z/B)$
- **B⁺-Δέντρα** είναι τα B-Δέντρα που **τα στοιχεία τους είναι αποθηκευμένα μόνο στα φύλλα**, ενώ οι εσωτερικοί κόμβοι χρησιμοποιούνται βοηθητικά για να καθοδηγούν τα μονοπάτια των αναζητήσεων.

Κατασκευή B -Δέντρων

□ Μέθοδος Insertion-Load:

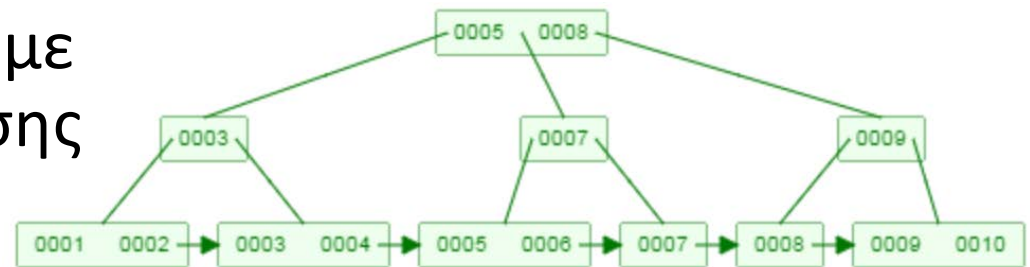
- Τα N στοιχεία εισέρχονται ένα-ένα χωρίς κάποια διατεταγμένη σειρά απαραίτητα.
- I/O Κόστος κατασκευής: $O(N \log_B N)$

□ Μέθοδος Bulk-Load:

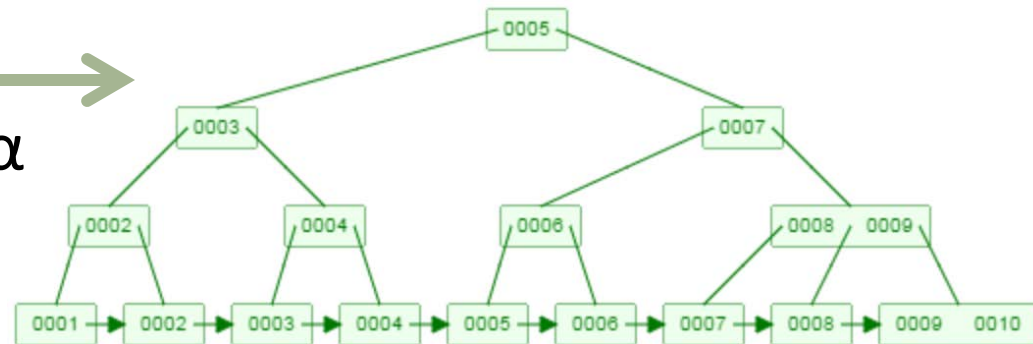
- Ταξινομούνται όλα τα N στοιχεία στην δευτερεύουσα μνήμη.
- Κατασκευάζονται τα φύλλα συνδέοντας τα ταξινομημένα στοιχεία.
- Χτίζεται το δένδρο από κάτω προς τα πάνω (ανά επίπεδο).
- I/O Κόστος κατασκευής: $\Theta\left(\frac{N}{B} \log_{M/B} \frac{N}{B}\right)$

Παραδείγματα

- Insertion Load και λειτουργία του B^+ -Tree με λεπτομέρειες υλοποίησης (Διαφάνειες).



- Γιατί δεν συμφέρει το Insertion Load σε ήδη ταξινομημένα δεδομένα (Προσομοιωτής).



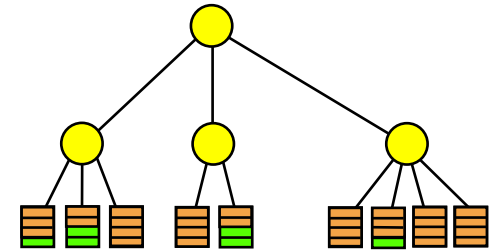
- Εισαγωγές, διαγραφές, διασπάσεις, συγχωνεύσεις, επαναζυγίσεις σε B^+ -Tree (Προσομοιωτής).



<https://www.cs.usfca.edu/~galles/visualization/BPlusTree.html> (University of San Francisco)

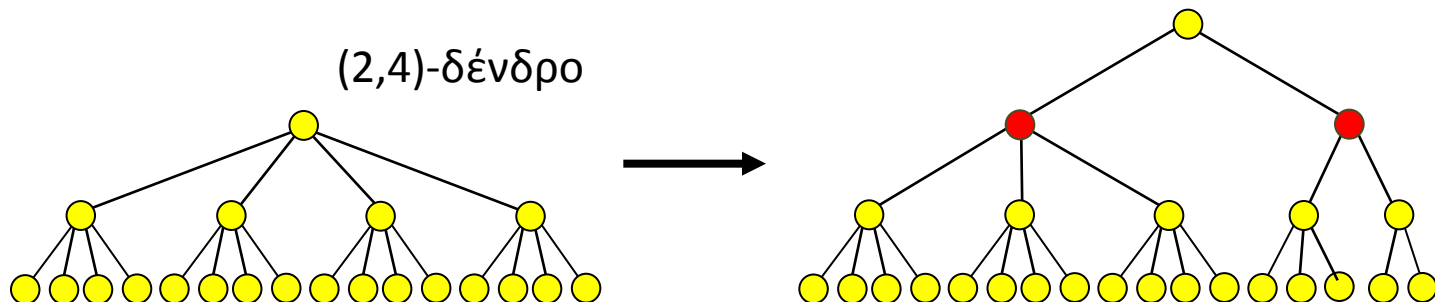
B -Δέντρα παραλλαγές των (α, b)

- B -Δέντρο με παράμετρο διακλάδωσης (branching factor) b και παράμετρο φύλλων k ($b, k \geq 8$):
 - Όλα τα φύλλα είναι στο ίδιο επίπεδο και περιέχουν από $k/4$ έως k στοιχεία.
 - Εκτός της ρίζας, όλοι οι κόμβοι έχουν βαθμό μεταξύ $b/4$ και b .
 - Η ρίζα έχει βαθμό μεταξύ 2 και b .
- B -Δέντρο με παράμετρο φύλλων $k = \Omega(B)$:
 - Χρησιμοποιεί $O(N/B)$ χώρο.
 - Έχει ύψος $O(\log_b(N/B))$.
 - Κάνει $O((1/(bk)) \log_b(N/B))$ επιμερισμένες επαναζυγίσεις (split/fuse) εσωτερικών κόμβων.
 - Το I/O κόστος κατασκευής του (bulk-load) είναι: $\Theta(\frac{N}{B} \log_{M/B} \frac{N}{B})$
- B -Δέντρο με παράμετρο διακλάδωσης $\Theta(B^c)$, $0 < c \leq 1$, και παράμετρο φύλλων $\Theta(B)$:
 - Χρησιμοποιεί $O(N/B)$ χώρο.
 - Έχει I/O κόστος ενημερώσεων/αναζητήσεων: $O(\log_{B^c} N) = O(\log_B N)$.
 - Έχει I/O κόστος ερωτημάτων περιοχής: $O(\log_B N + Z/B)$.



Η ανάγκη προσθήκης βαρών

- Όταν δευτερεύουσες δομές ή πράξεις είναι συνδεδεμένες με τους εσωτερικούς κόμβους, τότε δεν πρέπει να γίνονται συχνά επαναζυγίσεις στους πιο σημαντικούς.
- Το **βάρος** $w(v)$ ενός κόμβου v ορίζεται ως το πλήθος όλων των στοιχείων που περιλαμβάνονται σε όλα τα φύλλα του υπόδεντρου του.
- Είναι χρήσιμο ένα δέντρο στο οποίο κάθε κόμβος v με βάρος $w(v)$ συμμετέχει σε **μία μόνο πράξη επαναζύγισης για κάθε $\Omega(w(v))$ ενημερώσεις** που γίνονται κάτω από αυτόν.
- Οι κόμβοι των (α, b) -Δέντρων και κατά συνέπεια και των B -Δέντρων δεν έχουν αυτή την ιδιότητα.



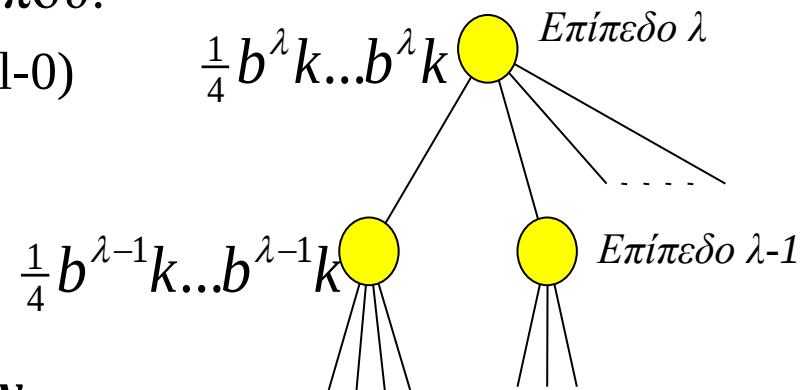
Βαροζυγισμένα B -Δέντρα

□ Βασική Ιδέα:

- Τοποθετούνται Περιορισμοί Βάρους στους κόμβους.
- Είναι καλύτερα ζυγισμένα από τα B -Δέντρα μετά από επαναζυγίσεις λόγω διασπάσεων συγχωνεύσεων και ενημερώσεων.

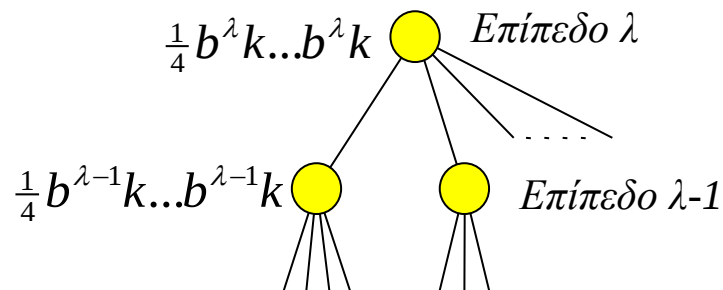
□ Βαροζυγισμένο B -Δέντρο με παράμετρο διακλάδωσης b και παράμετρο φύλλων k ($b \geq 8$, $k \geq 8$) είναι το B -Δέντρο όπου:

- Όλα τα φύλλα είναι στο ίδιο επίπεδο (level-0) και κάθε φύλλο v έχει βάρος: $k/4 \leq w(v) \leq k$.
- Κάθε εσωτερικός κόμβος v στο επίπεδο λ έχει βάρος: $w(v) \leq b^\lambda k$.
- Εκτός της ρίζας, κάθε εσωτερικός κόμβος v στο επίπεδο λ έχει βάρος: $w(v) \geq (1/4)b^\lambda k$.
- Η ρίζα έχει περισσότερα από ένα παιδιά.



Βαροζυγισμένα B -Δέντρα

- Οι περιορισμοί στους κόμβους από τα βάρη στην πραγματικότητα προκαλούν περιορισμούς και στον βαθμό τους (όπως έχουμε και στα απλά B -Δέντρα):
 - Ελάχιστος Βαθμός: $\frac{1}{4} b^\lambda k / b^{\lambda-1} k = \frac{1}{4} b$
 - Μέγιστος Βαθμός: $b^\lambda k / \frac{1}{4} b^{\lambda-1} k = 4b$
- Συνεπώς, κάθε βαροζυγισμένο δέντρο έχει ύψος: $O(\log_b \frac{N}{k})$



Ένθεση Στοιχείου

□ Εύρεση κατάλληλου φύλλου u και ένθεση του νέου στοιχείου.

□ Έλεγχος βαρών στο μονοπάτι από το u μέχρι τη ρίζα: $\frac{1}{4}b^{\lambda+1}k \dots b^{\lambda+1}k$

■ Αν ο κόμβος v στο επίπεδο λ έχει $w(v)=b^{\lambda}k+1$

τότε διασπάται σε v' και v'' με ιδανικά βάρη:

$$w(v') = \left\lfloor \frac{1}{2}(b^{\lambda}k + 1) \right\rfloor \quad w(v'') = \left\lceil \frac{1}{2}(b^{\lambda}k + 1) \right\rceil$$

□ Επειδή όμως είναι απίθανο να πετύχουμε επαναζυγίσεις

με ιδανικά βάρη μπορούμε πάντα να επιλέξουμε κατάλληλο πλήθος παιδιών

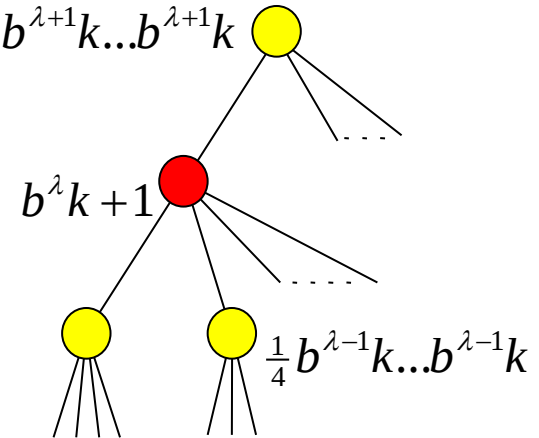
στη μοιρασιά στους v' και v'' ώστε τα βάρη τους να κυμαίνονται από:

$$\frac{1}{2}b^{\lambda}k - b^{\lambda-1}k \geq \frac{3}{8}b^{\lambda}k > \boxed{\frac{1}{4}b^{\lambda}k} \quad \text{μέχρι} \quad \frac{1}{2}b^{\lambda}k + b^{\lambda-1}k + 1 \leq \frac{5}{8}b^{\lambda}k + 1 < \boxed{b^{\lambda}k}$$

και να διατηρηθούν έτσι οι ιδιότητες του δέντρου.

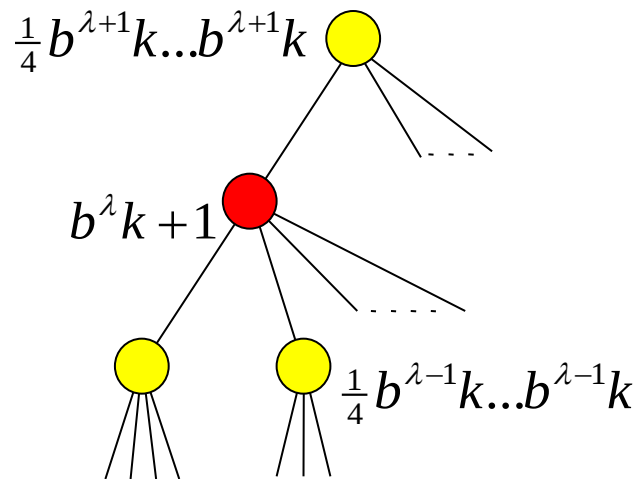
■ Αυτό συμβαίνει καθώς οι κόμβοι του επιπέδου $(\lambda-1)$ έχουν το πολύ βάρος:

$$b^{\lambda-1}k \leq \frac{1}{8}b^{\lambda}k \quad \text{αφού } b \geq 8.$$



Ένθεση Στοιχείου

- Με τον τρόπο αυτό απαιτούνται $\Omega(b^\lambda k)$ ενημερώσεις κάτω από τους κόμβους v' και v'' μέχρι την επόμενη πράξη επαναζύγισης.
- Πιο συγκεκριμένα απαιτούνται:
 $\frac{1}{8}b^\lambda k + 1$ διαγραφές ή $\frac{3}{8}b^\lambda k$ ενθέσεις κάτω από αυτούς.



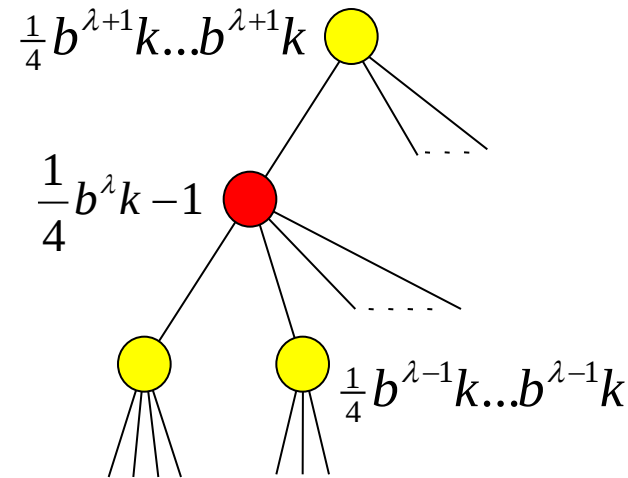
Διαγραφή Στοιχείου

- Εύρεση κατάλληλου φύλλου u και διαγραφή του στοιχείου.
- Έλεγχος βαρών στο μονοπάτι από το u μέχρι τη ρίζα:

- Αν ο κόμβος v στο επίπεδο λ έχει $w(v) = (1/4)b^\lambda k - 1$ τότε συγχωνεύεται με έναν αδερφό του. Ο τελικός κόμβος, έστω v' , θα έχει βάρος:

$$\frac{1}{4}b^\lambda k - 1 + \frac{1}{4}b^\lambda k \leq w(v') \leq \frac{1}{4}b^\lambda k - 1 + b^\lambda k$$

$$\Leftrightarrow \frac{1}{2}b^\lambda k - 1 \leq w(v') \leq \frac{5}{4}b^\lambda k - 1$$



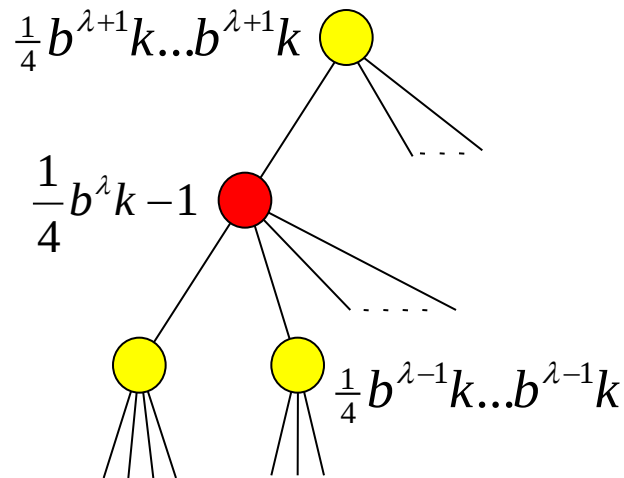
- Αν $w(v') \geq (7/8)b^\lambda k$ τότε διασπάμε σε δύο κόμβους (share) με κατάλληλη μοιρασιά ώστε τα βάρη να κυμαίνονται από:

$$\frac{7}{16}b^\lambda k - b^{\lambda-1}k - 1 \geq \frac{5}{16}b^\lambda k - 1 > \boxed{\frac{1}{4}b^\lambda k} \text{ μέχρι } \frac{5}{8}b^\lambda k + b^{\lambda-1}k \leq \frac{6}{8}b^\lambda k < \boxed{b^\lambda k}$$

και να διατηρηθούν έτσι οι ιδιότητες του δέντρου.

Διαγραφή Στοιχείου

- Με τον τρόπο αυτό πάλι απαιτούνται $\Omega(b^\lambda k)$ ενημερώσεις μέχρι την επόμενη πράξη επαναζύγισης.
- Πιο συγκεκριμένα απαιτούνται:
 $\frac{1}{8}b^\lambda k + 1$ ενθέσεις ή $\frac{1}{16}b^\lambda k$ διαγραφές κάτω από αυτούς.



Βαροζυγισμένα B -Δέντρα στη δευτερεύουσα μνήμη

- Βαροζυγισμένο B -Δέντρο με παράμετρο διακλάδωσης b και παράμετρο φύλλων $k=\Omega(B)$:
 - Χρησιμοποιεί γραμμικό χώρο: $O(N/B)$.
 - Έχει ύψος: $O(\log_b(N/B))$.
 - Απαιτεί $O(\log_b(N/B))$ επαναζυγιστικές πράξεις (splits/fuses) μετά από κάποια ενημέρωση.
 - Μεταξύ δύο διαδοχικών επαναζυγίσεων σε έναν κόμβο v απαιτούνται το λιγότερο $\Omega(w(v))$ ενημερώσεις.
 - Το I/O κόστος κατασκευής του (bulk-load) είναι: $\Theta(\frac{N}{B} \log_{M/B} \frac{N}{B})$
 - Έχει I/O κόστος ενημερώσεων/αναζητήσεων: $O(\log_b(N/B))$.
 - Έχει I/O κόστος ερωτημάτων περιοχής: $O(\log_b(N/B) + Z/B)$.

Βαροζυγισμένα B -Δέντρα στη δευτερεύουσα μνήμη

- Βαροζυγισμένο B -Δέντρο με παράμετρο διακλάδωσης $b = \Theta(B^c)$, $0 < c \leq 1$, και παράμετρο φύλλων $k = \Theta(B)$, (π.χ. $b = B/4$, $k = B$):
 - Χρησιμοποιεί γραμμικό χώρο: $O(N/B)$.
 - Έχει ύψος: $O(\log_B N)$.
 - Απαιτεί $O(\log_B N)$ επαναζυγιστικές πράξεις (splits/fuses) μετά από κάποια ενημέρωση.
 - Μεταξύ δύο διαδοχικών επαναζυγίσεων σε έναν κόμβο v απαιτούνται το λιγότερο $\Omega(w(v))$ ενημερώσεις.
 - Το I/O κόστος κατασκευής του (bulk-load) είναι: $\Theta(\frac{N}{B} \log_{M/B} \frac{N}{B})$
 - Έχει I/O κόστος ενημερώσεων/αναζητήσεων: $O(\log_B N)$.
 - Έχει I/O κόστος ερωτημάτων περιοχής: $O(\log_B N + Z/B)$.

A photograph of a dirt road winding through a dense forest. The road is light brown and leads into the distance, flanked by tall, slender trees with vibrant green foliage. The sunlight filters through the leaves, creating a dappled light effect on the road and the forest floor.

ΤΕΛΟΣ

Allegheny National Forest, Pennsylvania