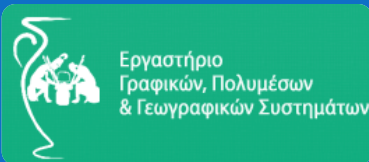


# Εκτενείς Δομές Δεδομένων (Κεφ. 5)

Δομές Δεδομένων

Τεστέμπασης Αθανάσιος  
Ε.Τ.Υ.



17 Μαρτίου 2015

[testebasis.thanos@gmail.com](mailto:testebasis.thanos@gmail.com)

# Περιεχόμενα

1. Εισαγωγή
2. AVL Δέντρα
3. (a,b) Δέντρα

# Εισαγωγή

- Δομές που βασίζονται σε συγκρίσεις : *Ισοζυγισμένα δέντρα εύρεσης* ( δέντρα τα φύλλα των οποίων απέχουν της ίδιας τάξεως μεγέθους, απόσταση απο τη ρίζα)
  - Υψοζυγισμένα δέντρα (κριτήριο ζύγισης κάθε κόμβου αποτελεί το ύψος των υποδέντρων του) :
    - AVL
    - B-TREES
    - (a,b) TREES
    - RED – BLACK TREE
  - Βαροζυγισμένα δέντρα (κριτήριο ζύγισης κάθε κόμβου αποτελεί το βάρος των υποδέντρων του) :
    - BB[a] TREES
    - SKIP LISTS
    - INTERPOLATION SEARCH TREES
    - ΒΑΡΟΖΥΓΙΣΜΈΝΑ B-TREES

# AVL TREE

- Δυαδικό ισοζυγισμένο δέντρο – σε κάθε κόμβο τα ύψη των υποδέντρων του διαφέρουν το πολύ κατά ένα.
- Υψοζύγισση του  $u$  :  $hb(u) = \text{ύψος} ( R(u) ) - \text{ύψος} ( L(u) )$   $hb(u) \{ +1 , 0 , -1 \}$
- Το ύψος  $h$  ενός δένδρου AVL με  $n$  στοιχεία είναι  $O(\log n)$ .

# AVL TREE – Access(x)

- Ξεκινάμε απο τη ρίζα και ελέγχουμε σε κάθε κόμβο  $u$  :
  - αν  $x < \text{val}(u)$  συνεχίζουμε αριστερά
  - αν  $x > \text{val}(u)$  συνεχίζουμε δεξιάεως οτου βρούμε το  $x$
- Ο χρόνος της  $\text{Access}(x)$  είναι  $\Theta(\log n)$

# AVL TREE – Insert( $x$ , $T$ )

## (1/2)

- Ο πρώτος κόμβος στο μονοπάτι από τον κόμβο  $v$  (που εισήχθη στο δένδρο) προς τη ρίζα, του οποίου το balance ήταν  $+1$  ή  $-1$  (πριν την εισαγωγή) ονομάζεται *κρίσιμος κόμβος* (κρίσιμο μονοπάτι αντίστοιχα).
- Αν ο κόμβος αυτός αποκτά balance  $+2$  ή  $-2$  μετά την εισαγωγή, τότε είναι ο πρώτος κόμβος στο μονοπάτι από τον  $v$  στη ρίζα για τον οποίο θα πρέπει να γίνουν κατάλληλες ενέργειες ώστε να διορθωθεί το balance του.
- Διακρίνουμε περιπτώσεις ανάλογα με το είδος των δύο πρώτων ακμών του μονοπατιού από τον κρίσιμο κόμβο  $w$  προς τον εισαχθέντα κόμβο  $v$  (απλή περιστροφή / διπλή περιστροφή)

# AVL TREE – Insert( $x$ , $T$ )

## (2/2)

- Περίπτωση RR (Right-Right)
  - Και οι δύο ακμές οδηγούν δεξιά. Εκτελούμε μια αριστερή περιστροφή γύρω από τον κρίσιμο κόμβο.
- Περίπτωση LL (Left-Left)
  - Και οι δύο ακμές οδηγούν αριστερά. Είναι συμμετρική της περίπτωσης RR! Μία δεξιά περιστροφή γύρω από τον κρίσιμο κόμβο αρκεί για να επιλυθεί το πρόβλημα με το balance του!
- Περίπτωση RL (Right-Left)
  - Η πρώτη ακμή οδηγεί δεξιά και η δεύτερη αριστερά. Απαιτούνται δύο περιστροφές, μια δεξιά περιστροφή γύρω από τον επόμενο του κρίσιμου κόμβου στο μονοπάτι που οδηγεί στον  $n$  και μια αριστερή περιστροφή γύρω από τον κρίσιμο κόμβο.
- Περίπτωση LR (Left-Right)
  - Η πρώτη ακμή οδηγεί αριστερά και η δεύτερη δεξιά. Είναι συμμετρική της περιπτώσεως RL. Απαιτούνται δύο περιστροφές, μια αριστερή περιστροφή γύρω από τον επόμενο του κρίσιμου κόμβου στο μονοπάτι που οδηγεί στον  $n$  και μια δεξιά περιστροφή γύρω από τον κρίσιμο κόμβο.

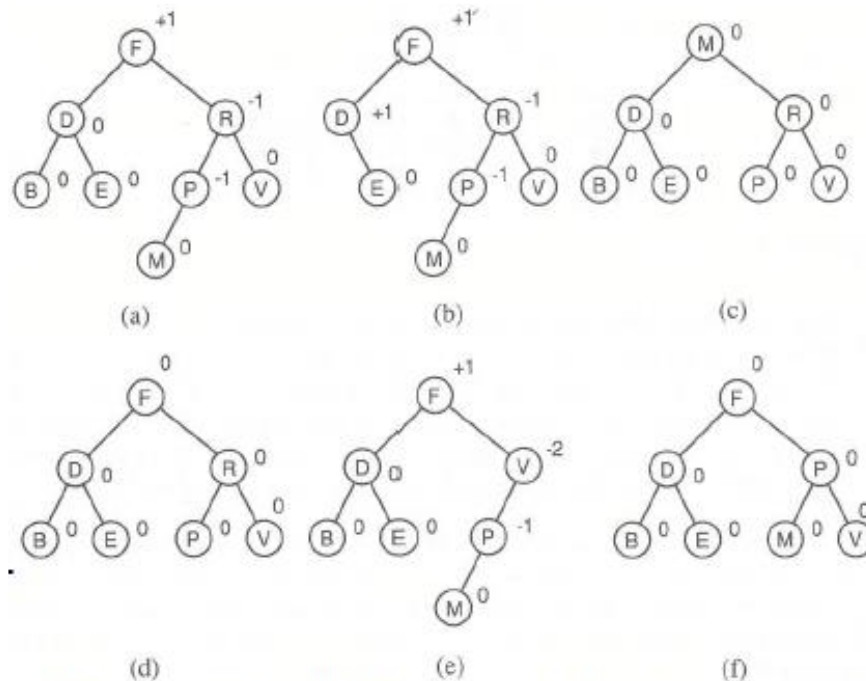
# AVL TREE – Delete( $x$ , $T$ )

## (1/2)

- Ακολουθούμε τον ‘αλγόριθμο’ διαγραφής σε δυαδικό δένδρο:
  - Access( $x$ )
  - Διαγραφή του ίδιου του κόμβου  $x$  αν είναι φύλλο.
  - Αντικατάστασή του από το παιδί του αν έχει μόνο ένα παιδί.
  - Αντικατάστασή του από τον επόμενό του στην ενδοδιατεταγμένη διάταξη αν έχει δύο παιδιά.
  - Balance



# AVL TREE – Delete( x , T ) (2/2)



(a) Αρχικό δένδρο, (b) Διαγραφή του B, (c) Διαγραφή του F,  
(d) Διαγραφή του M, (e) Διαγραφή του R

# (a,b) TREES

- Έστω  $a, b$  ακέραιοι τέτοιοι ώστε  $a \geq 2$  και  $b \geq 2a-1$ . Ένα δέντρο  $T$  είναι  $(a, b)$  αν
  - Όλα τα φύλλα του  $T$  έχουν το ίδιο βάθος (δηλαδή το δέντρο είναι πλήρως ζυγισμένο)
  - Για κάθε κόμβο  $u$  του  $T$ , ισχύει  $p(u) \leq b$   $\{p(u) = \text{αριθμός των παιδιών του } u\}$
  - Για κάθε κόμβο  $u$  του  $T$ , με εξέρεση τη ρίζα, ισχύει  $p(u) \geq a$
  - Για τη ρίζα  $r$ , ισχύει  $p(r) \geq 2$
- Όταν  $b = 2a - 1$  τότε το  $(a,b)$  tree ονομάζεται B-tree.

## (a,b) TREES – Access(x)

- Ίδια διαδικασία όπως στα AVL trees

## (a,b) TREES – Insert ( $x$ , $T$ )

- Διαδικασία Access( $x$ )
- Δημιουργία ενός νέου κενού φύλλου στο οποίο αποθηκεύουμε το  $x$ . Αν  $x < y$  τότε τοποθετείται αριστερά, αν  $x > y$  τοποθετείτε δεξιά.
- Balance
  - Split

## (a,b) TREES – Delete ( $x$ , $T$ )

- Διαδικασία Access( $x$ )
- Σβήνεται το φύλλο με τη τιμή  $x$  και στον πατέρα  $u$  του  $x$  διαγράφεται το κλειδί  $k_i(u)$  αν το φύλλο  $x$  είναι  $i$ -στο παιδί του  $u$  ή  $k_{p(u)-1}(u)$  αν το  $p(u)$ -στο παιδί του.
- Balance
  - Share(διαμοιρασμός)
  - Fusion(σύμπτυξη)