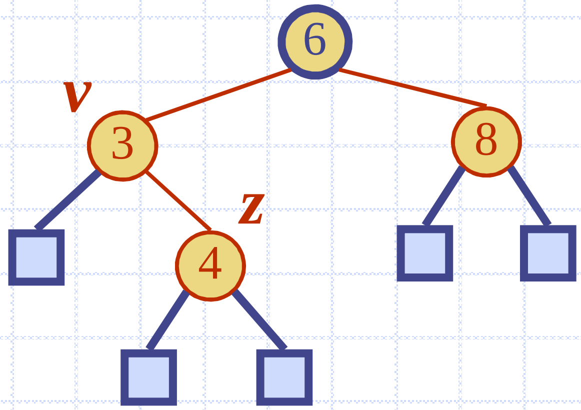
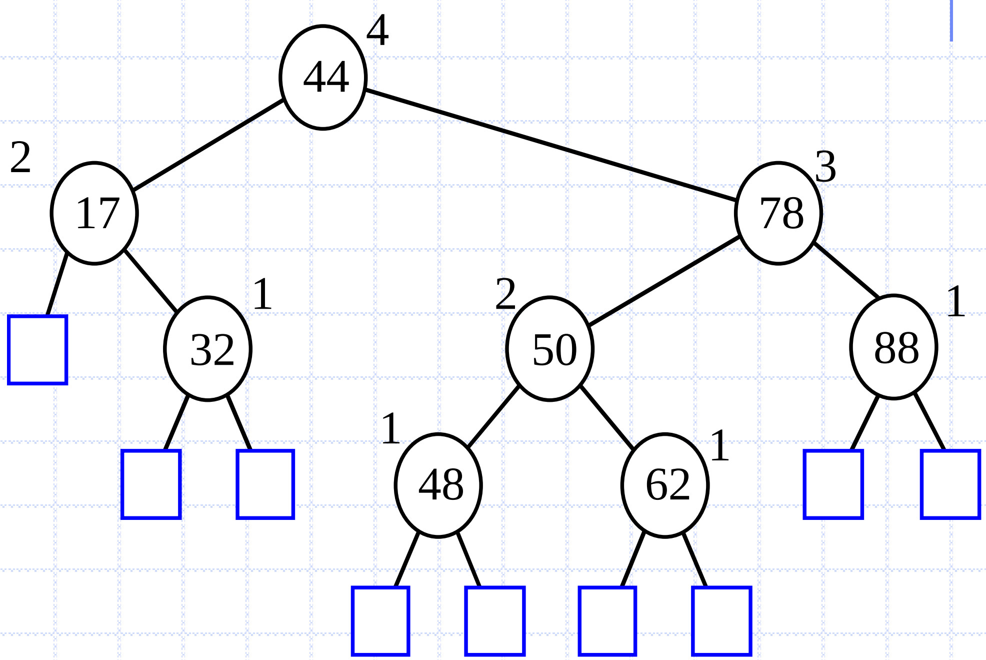
AVL Δένδρα



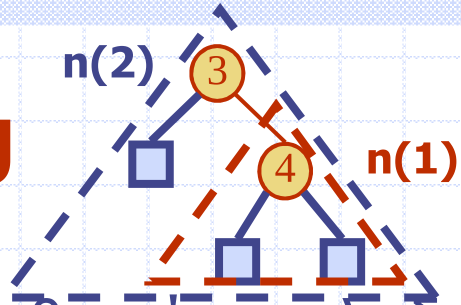
Ορισμός AVL Δένδρων

- Τα AVL δένδρα είναι ισοζυγισμένα
- Ένα AVL δένδρο είναι ένα δυαδικό δένδρο αναζήτησης τέτοιο που για κάθε εσωτερικό κόμβο v του T, τα ύψη των παιδιών του v μπορεί να διαφέρουν το πολύ κατά 1



Ένα παράδειγμα ενόςAVL δένδρου όπου δείχνουμε δίπλα σε κάθε κόμβο τα ύψη:

Ύψος ενός AVL Δένδρου



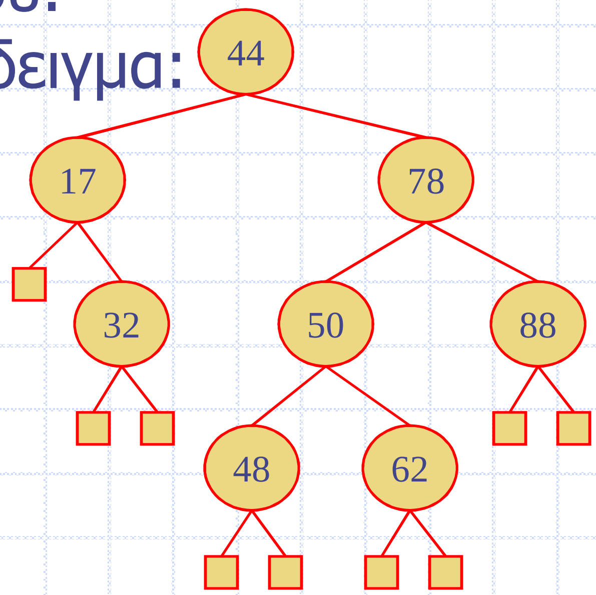
- Γεγονός: Το ύψος ενός AVL δένδρου που αποθηκεύει n κλειδιά είναι O(log n).
- Απόδειξη: Έστω n(h): το ελάχιστο πλήθος εσωτερικών κόμβων ενός AVL δένδρου ύψους h.
- Εύκολα διαπιστώνουμε ότι n(1) = 1 και n(2) = 2
- Για n > 2, ένα AVL δένδρο ύψους h περιέχει τον κόμβο της ρίζας, ένα AVL υποδένδρο ύψους n-1 και ένα άλλο ύψους n-2.
- Δηλαδή, n(h) = 1 + n(h-1) + n(h-2)
- Ξέροντας ότι n(h-1) > n(h-2), έχουμε n(h) > 2n(h-2). Δηλ.

  n(h) > 2n(h-2), n(h) > 4n(h-4), n(h) > 8n(n-6), … (με επαγωγή),

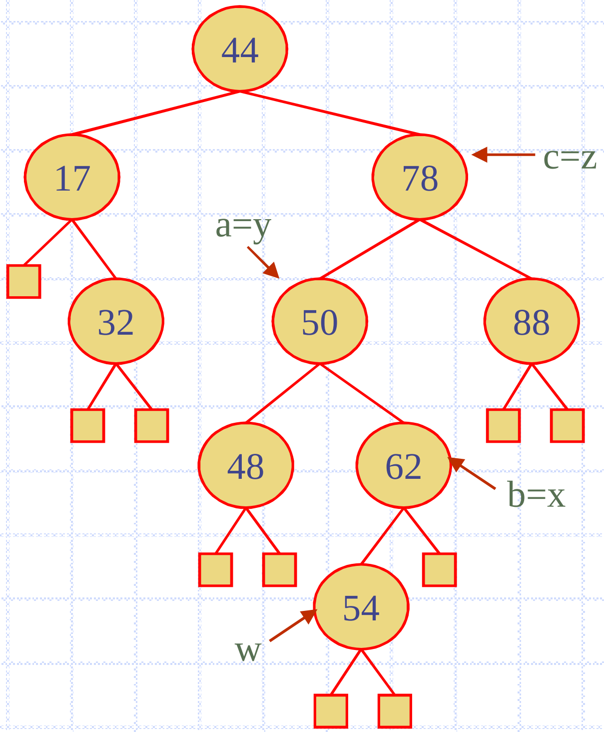
  $n(h) > 2^i n(h-2i)$
- Λύνοντας την βασική περίπτωση έχουμε : $n(h) > 2^{h/2-1}$
- Λογαριθμίζοντας: h < 2log n(h) +2
- Επομένως το ύψος ενός AVL δένδρου είναι O(log n)

Εισαγωγή

- Η εισαγωγή αποτελεί μια αναζήτηση δυαδικού δένδρου
- Επιτυγχάνεται πάντα με επέκταση ενός εξωτερικού κόμβου.
- Παράδειγμα:



πριν την εισαγωγή



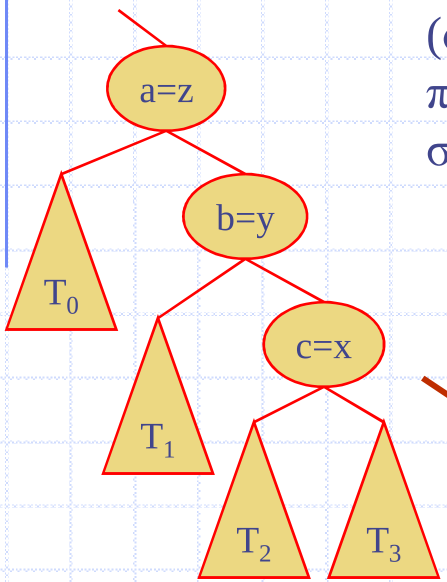
μετά την εισαγωγή

Αλγόριθμος: restructure(x):
Είσοδος: ένας κόμβος x ενός δυαδικού δένδρου αναζήτησης T με γονέα τον y και παππού τον z
Έξοδος: Το δένδρο T μετά την αναδόμηση τριών κόμβων που περιλαμβάνει τους x, y και z

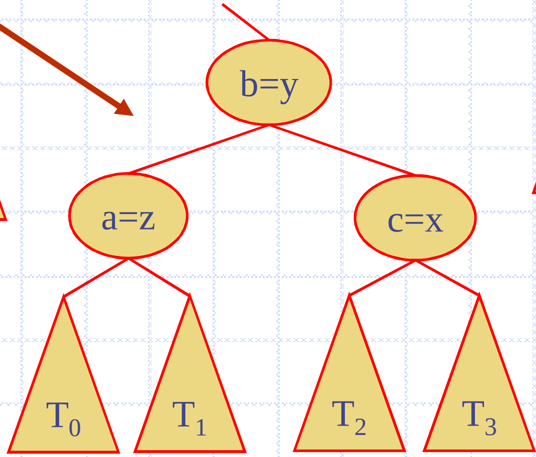
1. Έστω (a,b,c) η από αριστερά προς τα δεξιά ενδοδιατεταγμένη σειρά των κόμβων x, y, και z, και έστω (T_0, T_1, T_2, T_3) η ενδοδιατεταγμένη σειρά των τεσσάρων υποδένδρων των x, y, και z, που δεν έχουν ρίζα τα x, y ή z.
2. Αντικατάσταση του υποδένδρου με ρίζα το z με ένα νέο υποδένδρο με ρίζα το b.
3. Θέσε a το αριστερό παιδί του b και έστω T_0 και T_1 το αριστερό και δεξιό υποδένδρο του a, αντίστοιχα.
4. Θέσε c το δεξιό παιδί του b και έστω T_2 και T_3 το αριστερό και δεξιό υποδένδρο του c, αντίστοιχα.

Αναδόμηση τριών κόμβων

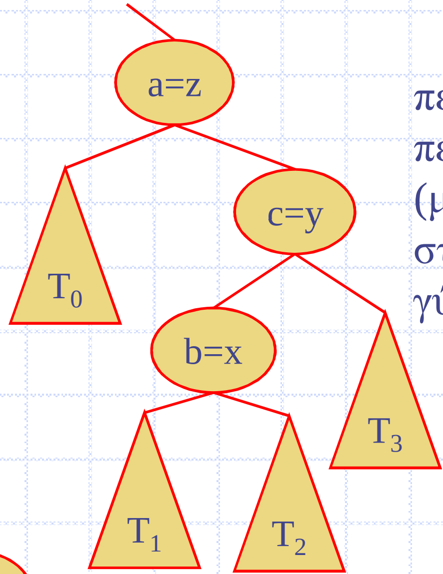
- έστω (*a*,*b*,*c*) μια ενδοδιατεταγμένη σειρά των *x*, *y*, *z*
- εκτελούμε τις απαιτούμενες περιστροφές να πάει το *b* στον πιό ψηλό κόμβο του δένδρου



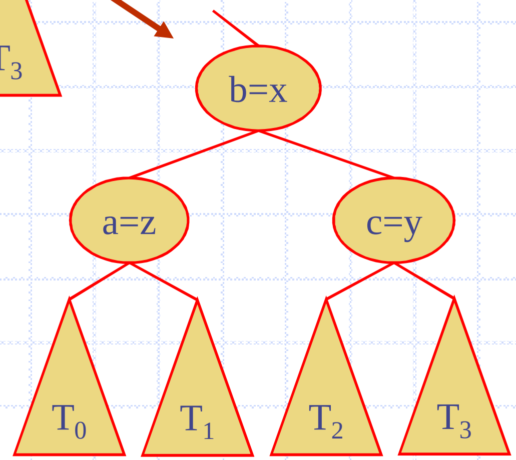
(οι άλλες δύο περιπτώσεις είναι συμμετρικές)



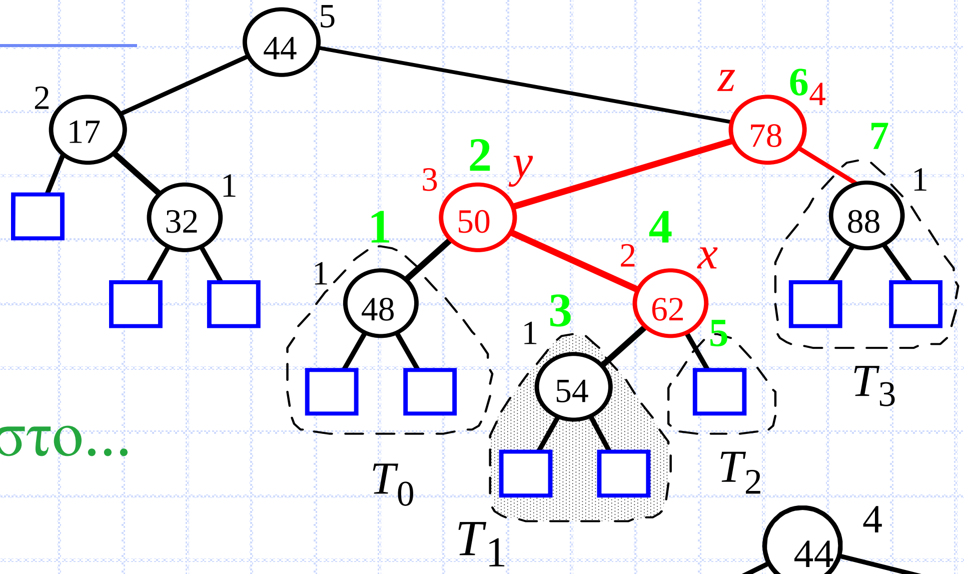
περίπτωση 1: απλή περιστροφή
(μια αριστερή περιστροφή γύρω *a*)



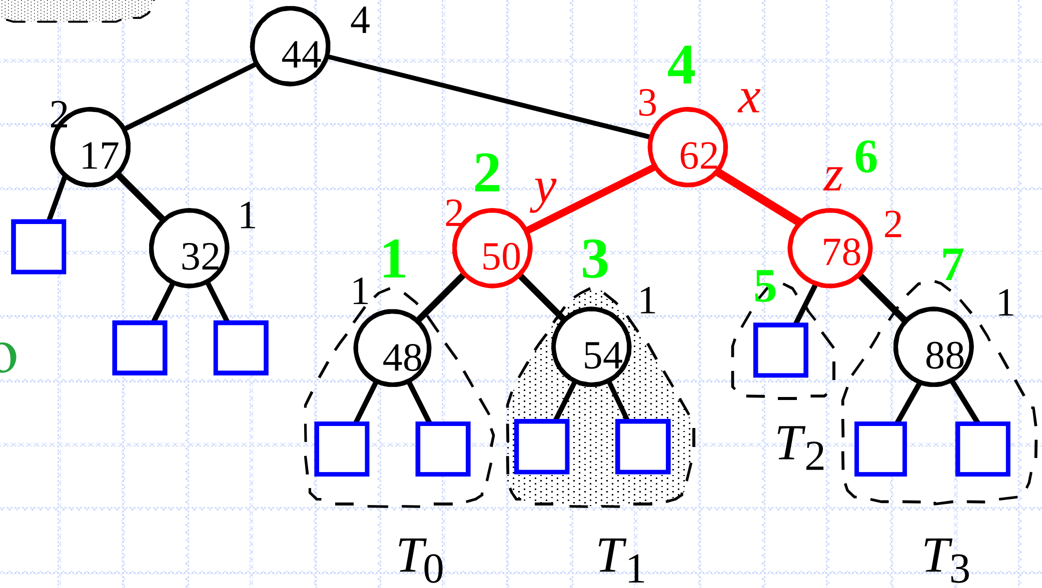
περίπτωση 2: διπλή περιστροφή
(μια δεξιά γύρω από το *c*, στη συνέχεια μια αριστερή γύρω από το *a*)



Συνέχεια Παράδειγμα Εισαγωγής



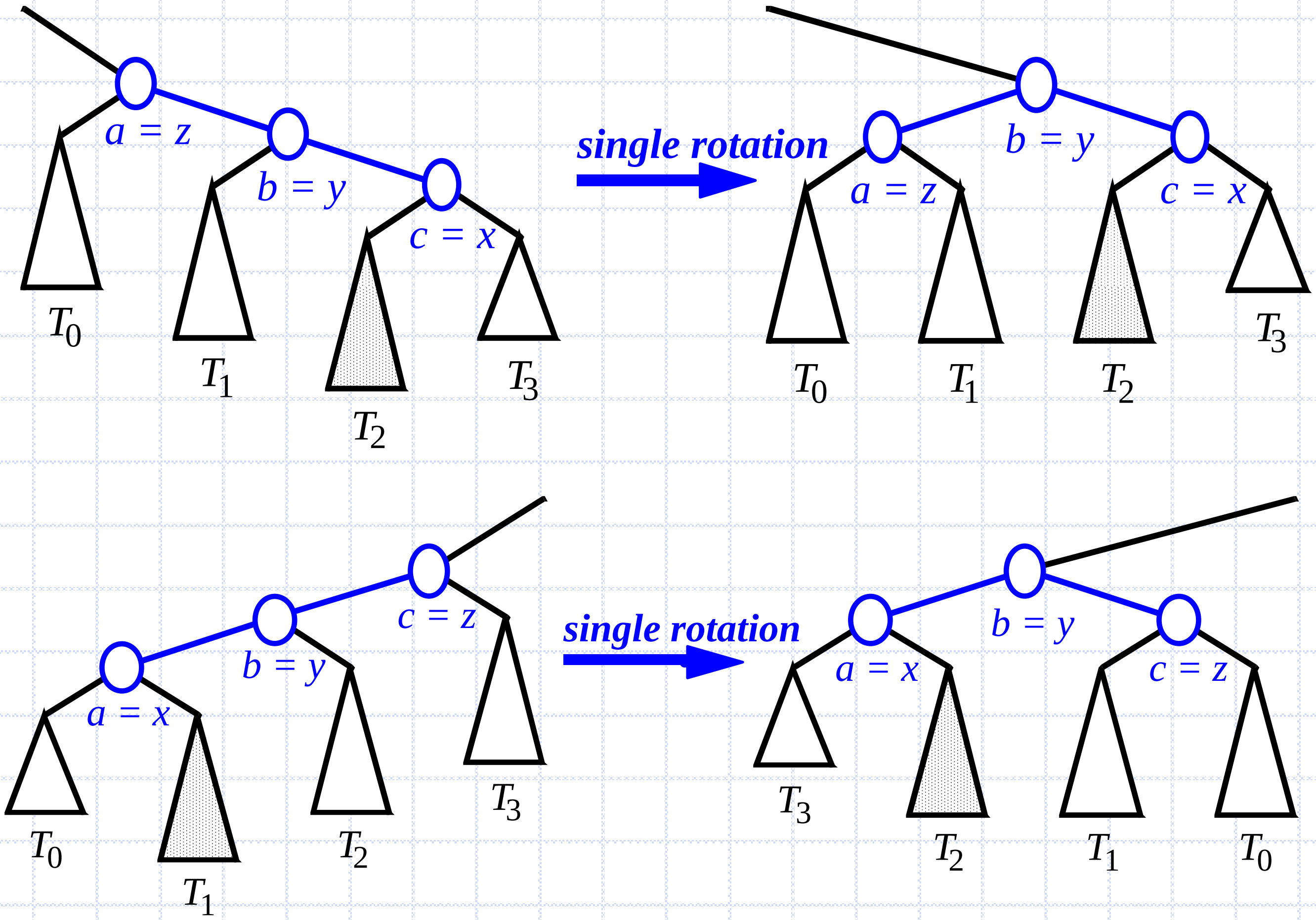
αζύγιστο...



...ζυγισμένο

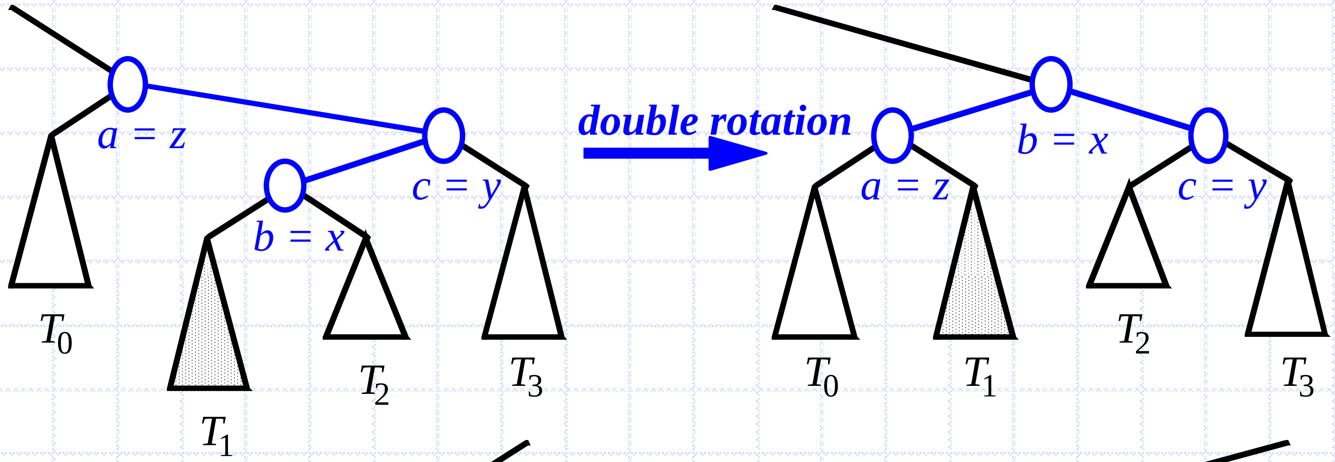
Αναδόμηση (σαν απλές περιστροφές)

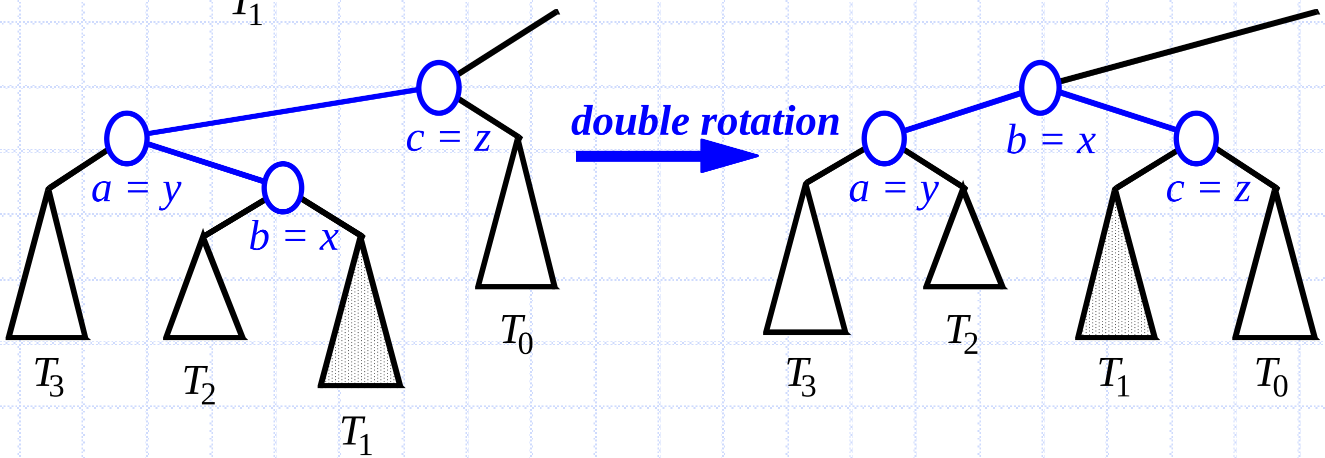
- Απλές Περιστροφές:



Αναδόμηση (σαν διπλές περιστροφές)

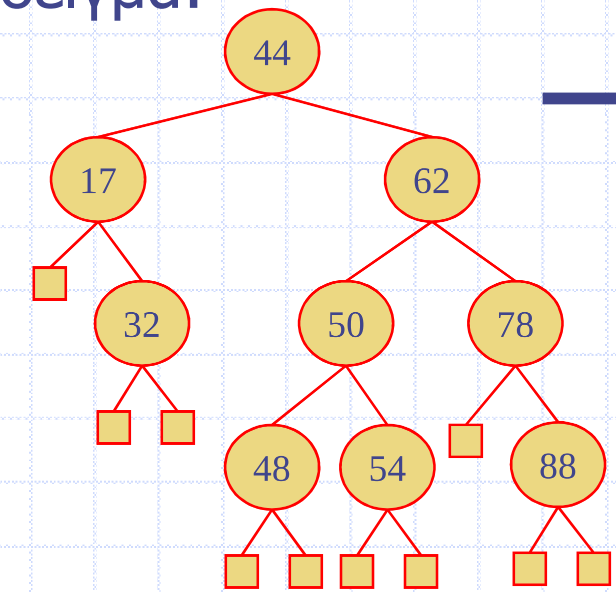
- διπλές περιστροφές:



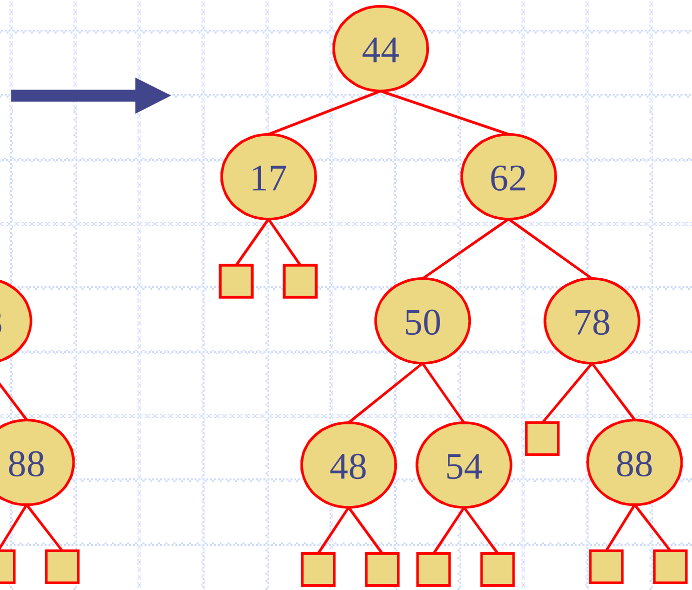


Διαγραφή

- Η διαγραφή ξεκινά σαν μια αναζήτηση σε ένα δυαδικό δένδρο αναζήτησης, που σημαίνει ο κόμβος που διαγράφεται θα γίνει ένας κενός εξωτερικός κόμβος. Ο γονέας του, w, μπορεί να διαταράξει την ισορροπία.
- Παράδειγμα:



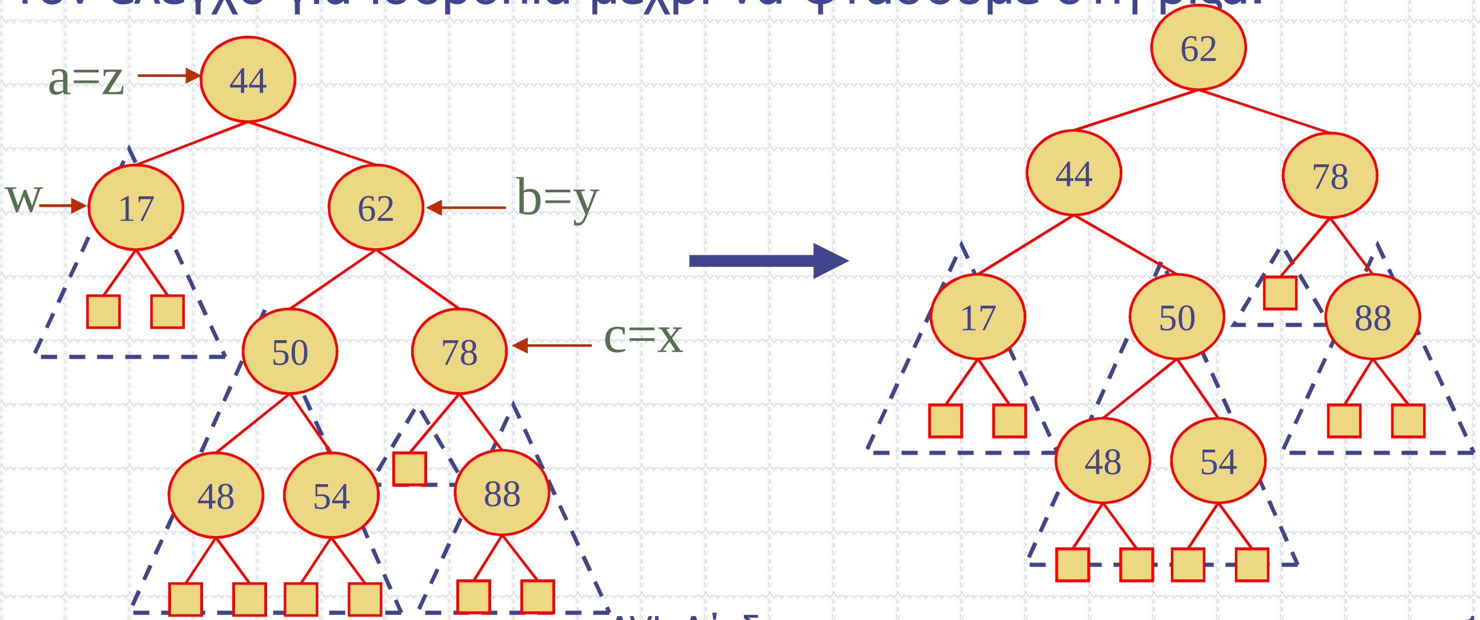
πριν την διαγραφή του 32



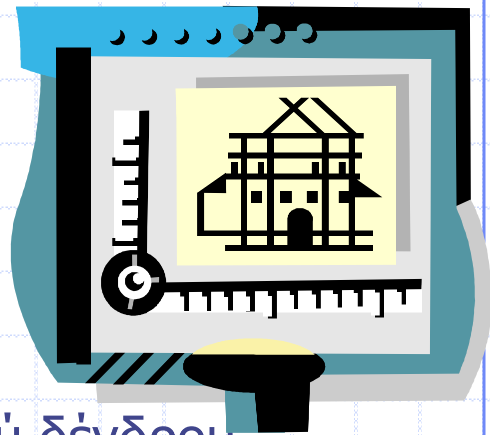
μετά την διαγραφή

Εξισορόπηση μετά τη διαγραφή

- Έστω z ο πρώτος μη ισοροπημένος κόμβος που συναντάμε ανεβαίνοντας το δένδρο από τον w. Επίσης, έστω y το παιδί του z με το μεγαλύτερο ύψος, και έστω x το παιδί του y με το μεγαλύτερο ύψος
- Εκτελούμε την restructure(x) για αποκατάσταση της ισοροπίας στον z
- Καθώς η αναδόμηση μπορεί να διαταράξει την ισοροπία ενός άλλου κόμβου ψηλότερα στο δένδρο, πρέπει να συνεχίσουμε τον έλεγχο για ισοροπία μέχρι να φτάσουμε στη ρίζα.



Απόδοση AVL Δένδρων



- μια απλή αναδόμηση απαιτεί χρόνο O(1)
  - χρησιμοποιώντας συνδεδεμένη δομή δυαδικού δένδρου
- get απαιτεί O(log n) χρόνο
  - το ύψος του δένδρου είναι O(log n), δεν απαιτεί αναδόμηση
- put απαιτεί O(log n) χρόνο
  - αρχική εύρεση θέλει O(log n)
  - Αναδόμηση του δένδρου προς τα πάνω, για διαχείριση των υψών θέλει O(log n)
- remove απαιτεί O(log n) χρόνο
  - Αρχική εύρεση είναι O(log n)
  - Αναδόμηση του δένδρου προς τα πάνω, με διαχείριση των υψών είναι O(log n)