Δομές Δεδομένων

Δέντρα Αναζήτησης

Δημήτρης Μιχαήλ

Τμήμα Πληροφορικής και Τηλεματικής
Χαροκόπειο Πανεπιστήμιο

Το πρόβλημα

Αναζήτηση

Θέλουμε να διατηρήσουμε αντικείμενα με κλειδιά και να μπορούμε εκτός από προσθαφαίρεση αντικειμένων να κάνουμε γρήγορη αναζήτηση κάποιου αντικειμένου με βάση ένα κλειδί.

Παράδειγμα

Έχουμε αντικείμενα που αντιπροσωπεύουν τραπεζικές συναλλαγές και θέλουμε να κάνουμε αναζήτηση με βάση ένα κλειδί που είναι η ημερομηνία.

Αναζήτηση

Μπορούμε να υλοποιήσουμε μία δομή που να παρέχει αναζήτηση με διάφορους τρόπους.

Το ζητούμενο είναι να είναι η αναζήτηση αποδοτική ταυτόχρονα με τις υπόλοιπες λειτουργίες.

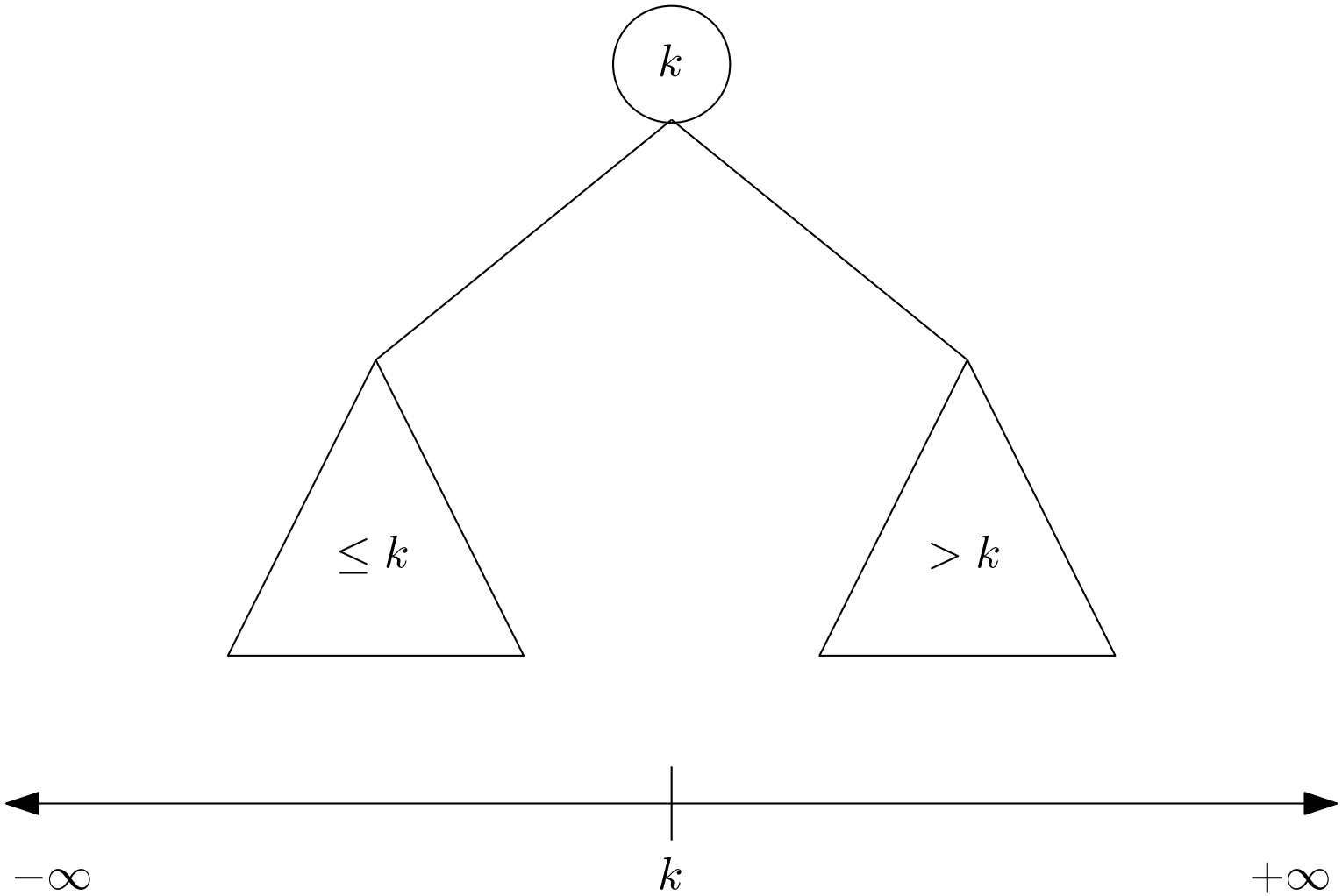
π.χ μία υλοποίηση αναζήτησης σε λίστα δεν είναι αποδοτική

Ορισμός

Ένα **δυαδικό δέντρο αναζήτησης** είναι ένα δυαδικό δέντρο όπου κάθε κόμβος είναι συσχετισμένος με ένα κλειδί και με την πρόσθετη ιδιότητα ότι το κλειδί οποιουδήποτε κόμβου είναι μεγαλύτερο (ή ίσο) από τα κλειδιά όλων των κόμβων του αριστερού υποδέντρου αυτού του κόμβου, και μικρότερο από τα κλειδιά όλων των κόμβων του δεξιού υποδέντρου αυτού του κόμβου.

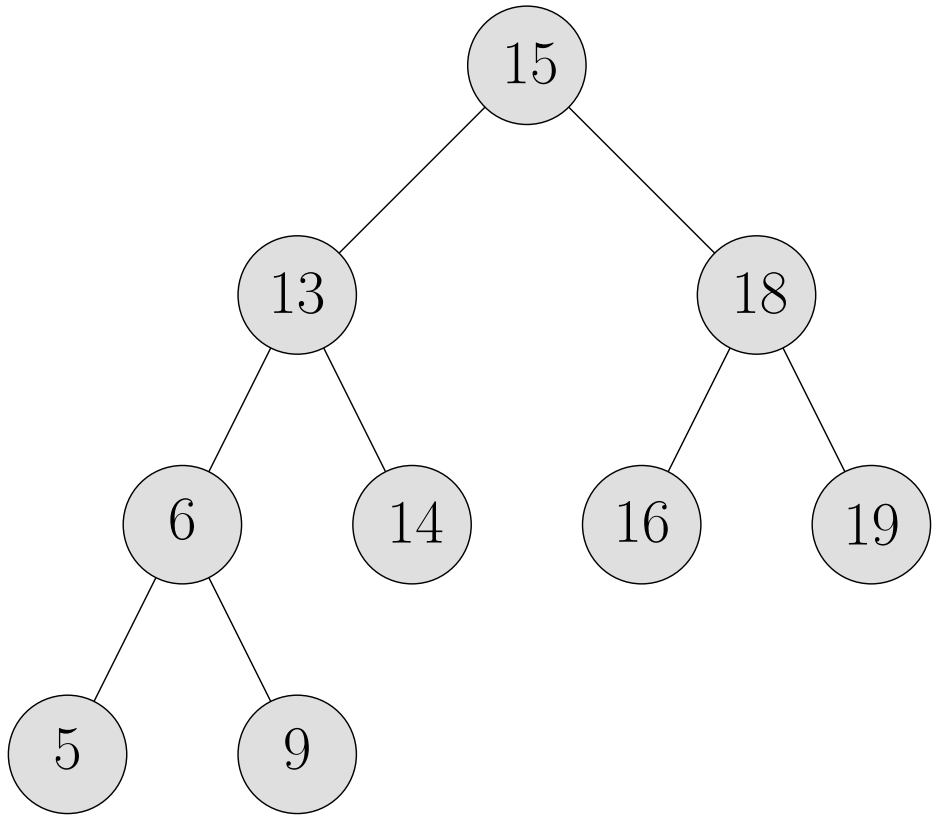
Δυαδικά Δέντρα Αναζήτησης (ΔΔΑ)

Binary Search Tree (BST)



Παράδειγμα

Δυαδικό Δέντρα Αναζήτησης (ΔΔΑ)



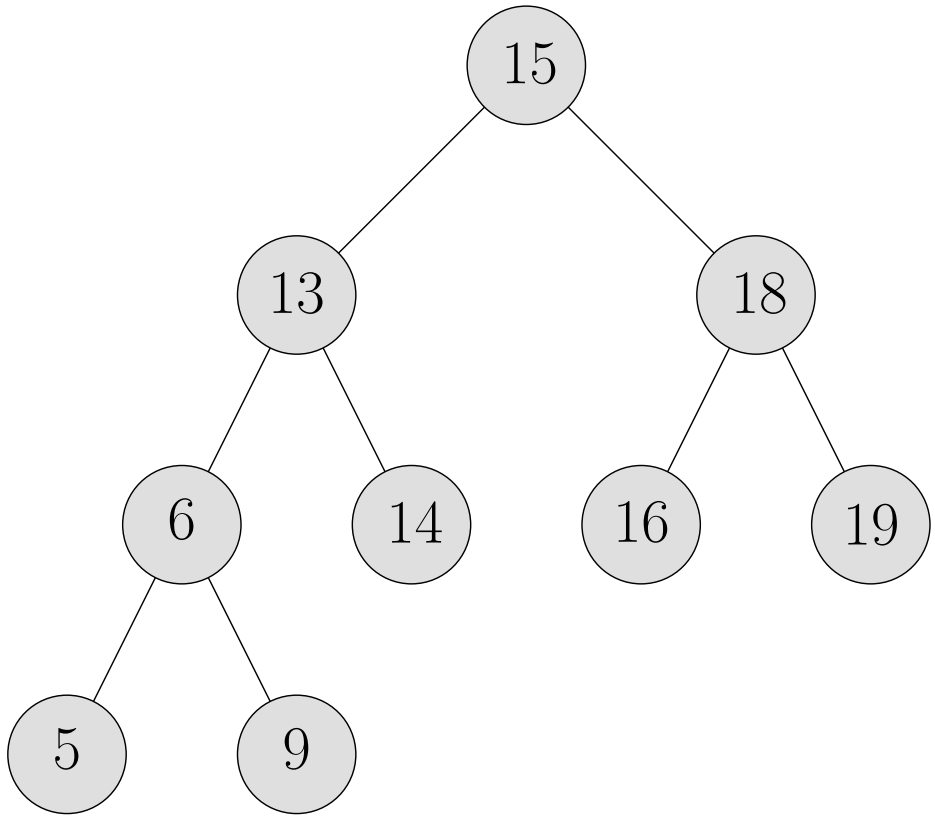
Αναζήτηση σε ΔΔΑ

Αλγόριθμος

- ξεκινάμε από την ρίζα
- εάν είμαστε σε ένα κόμβο και βρούμε το κλειδί που ψάχνουμε τότε επιστρέφουμε την πληροφορία αυτού του κόμβου
- αλλιώς πάμε δεξιά ή αριστερά ανάλογα με την σύγκριση του κλειδιού που ψάχνουμε και του κλειδιού του κόμβου

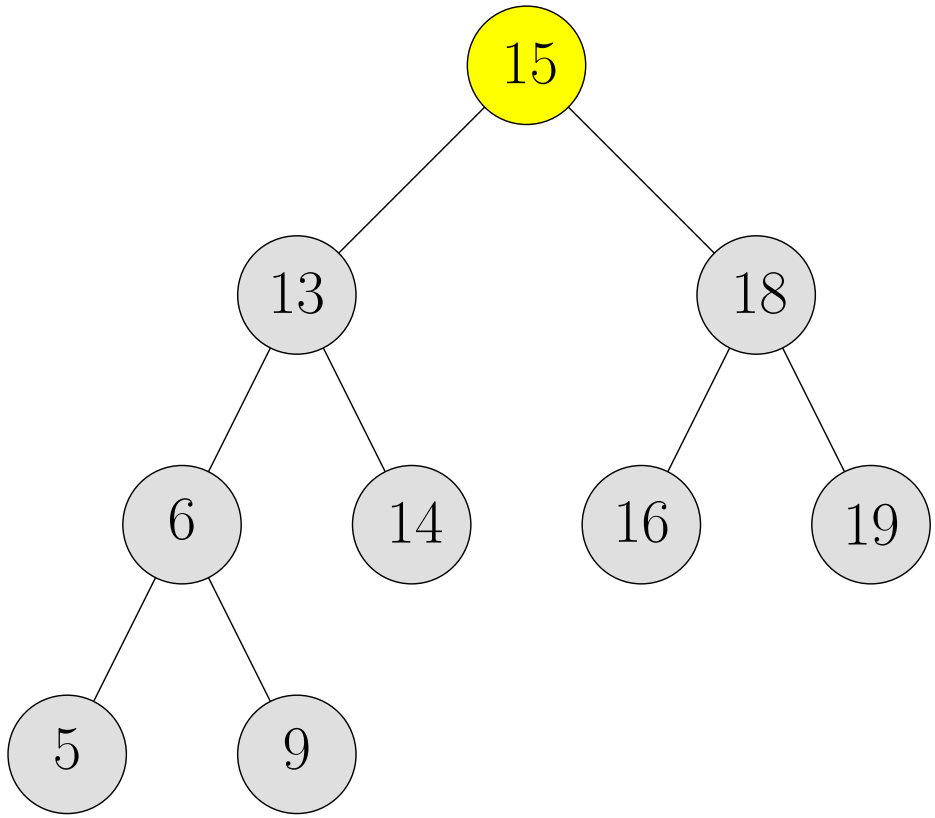
Αναζήτηση σε ΔΔΑ

Αναζήτηση του 9



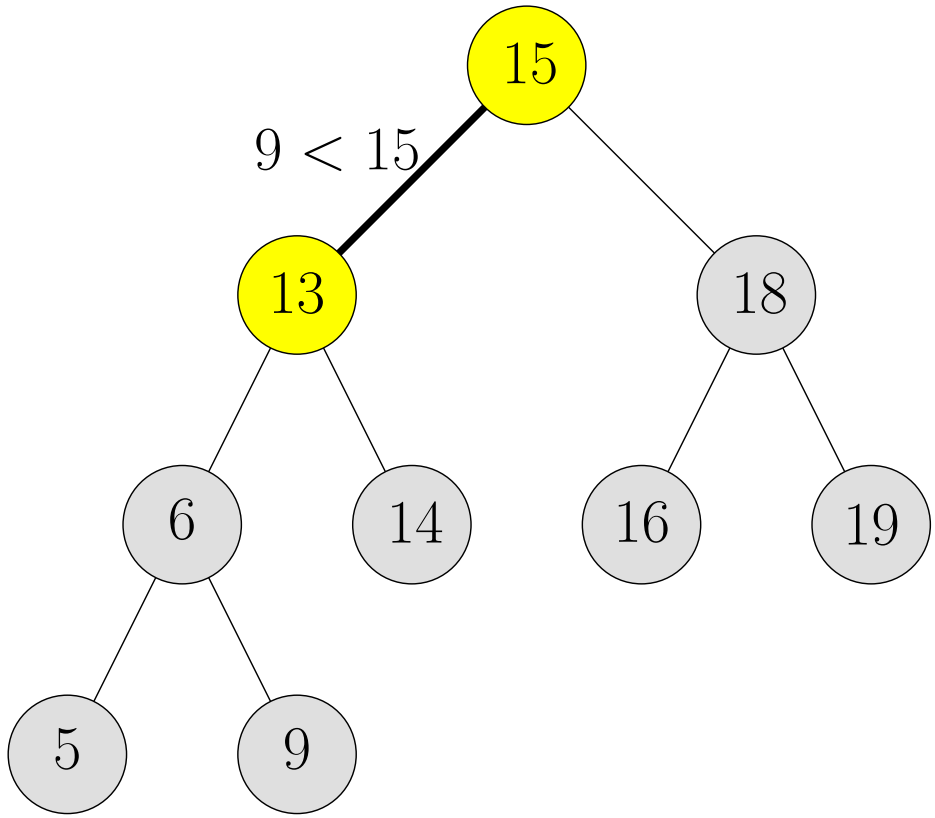
Αναζήτηση σε ΔΔΑ

Αναζήτηση του 9



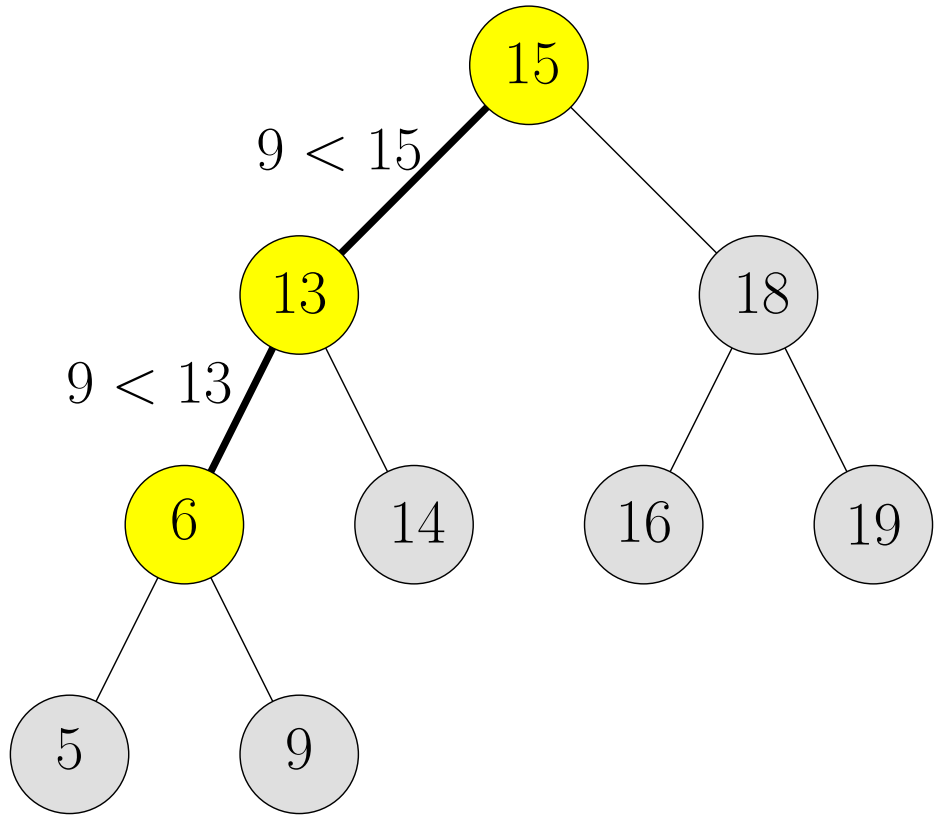
Αναζήτηση σε ΔΔΑ

Αναζήτηση του 9



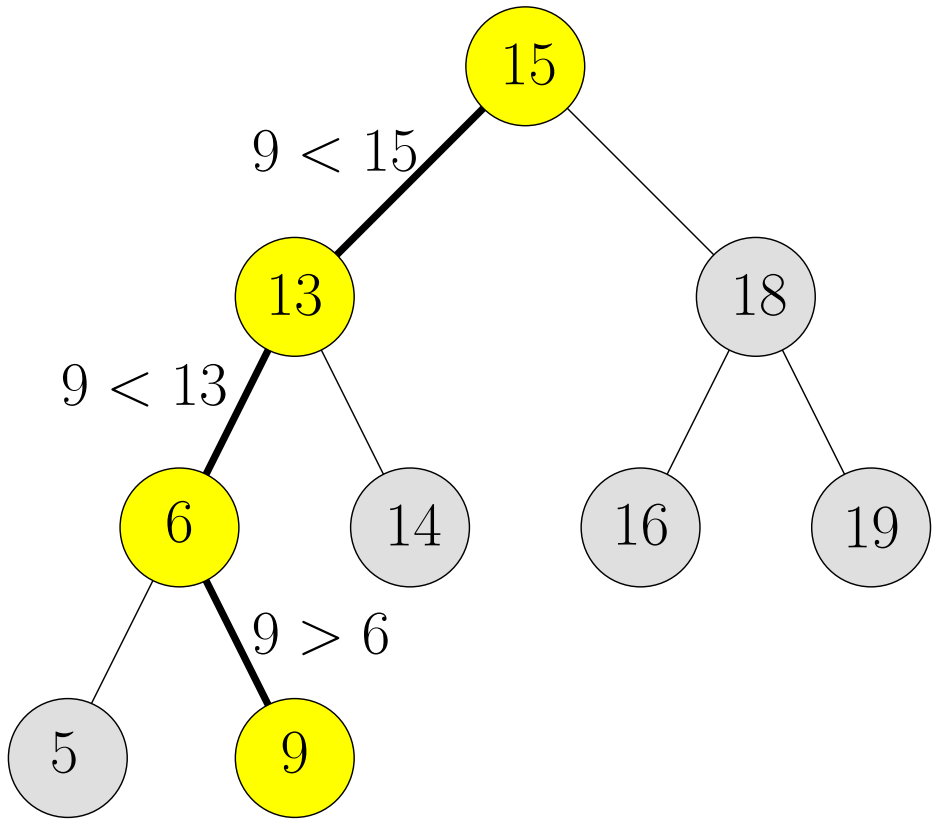
Αναζήτηση σε ΔΔΑ

Αναζήτηση του 9



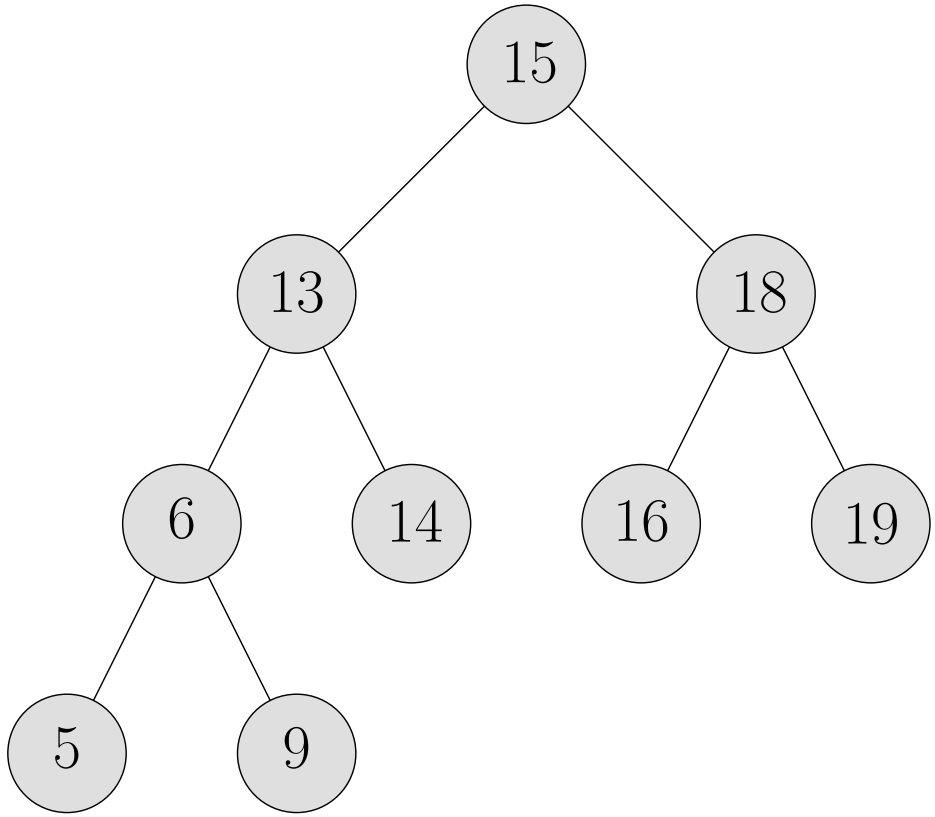
Αναζήτηση σε ΔΔΑ

Αναζήτηση του 9



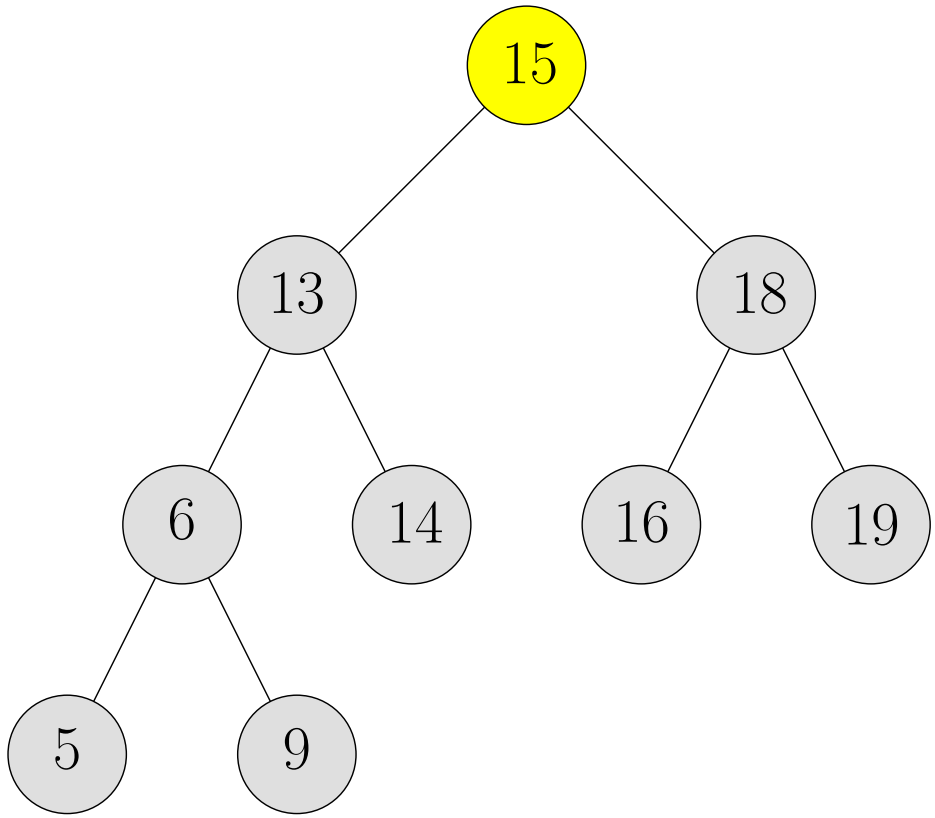
Αναζήτηση σε ΔΔΑ

Αναζήτηση του 17



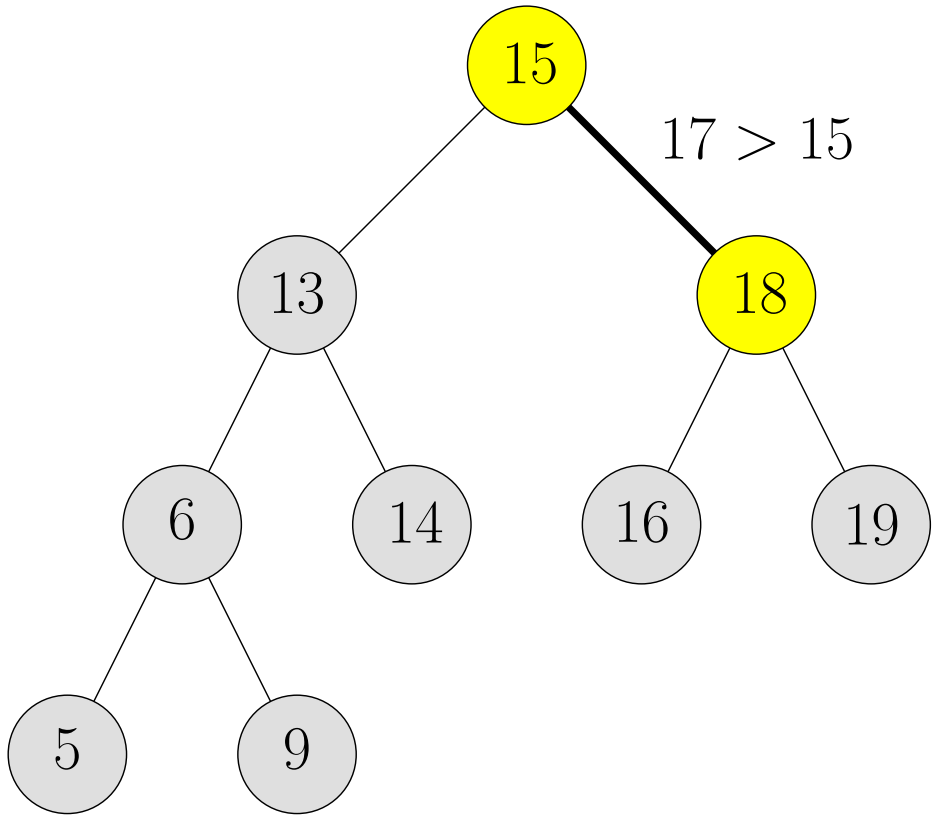
Αναζήτηση σε ΔΔΑ

Αναζήτηση του 17



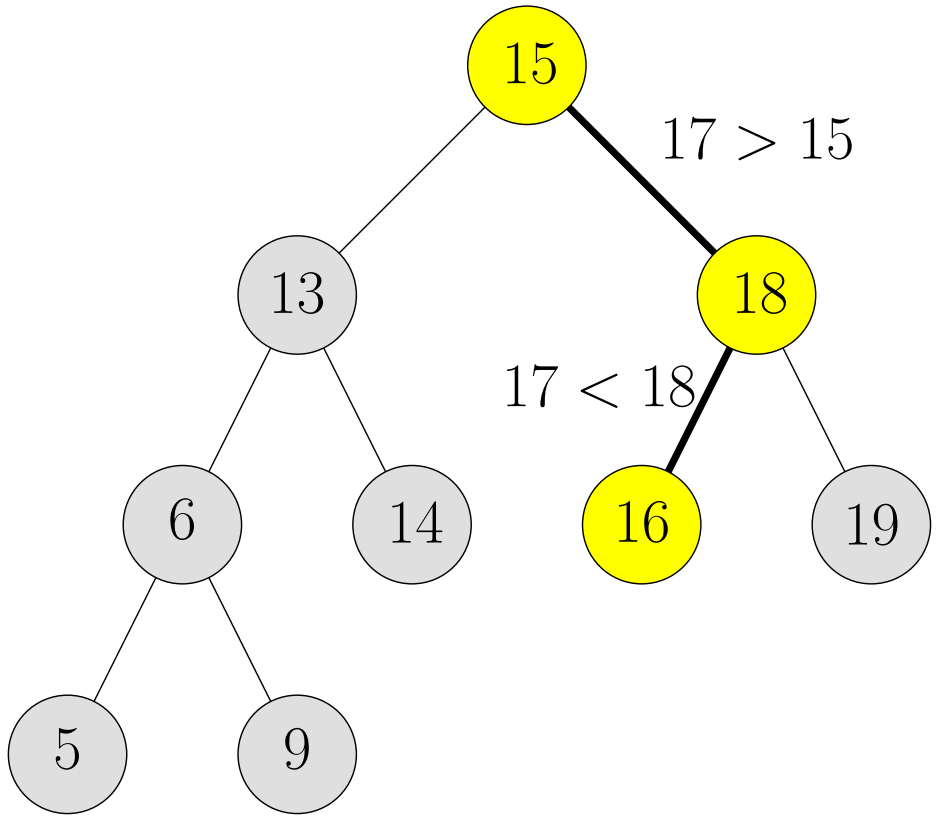
Αναζήτηση σε ΔΔΑ

Αναζήτηση του 17



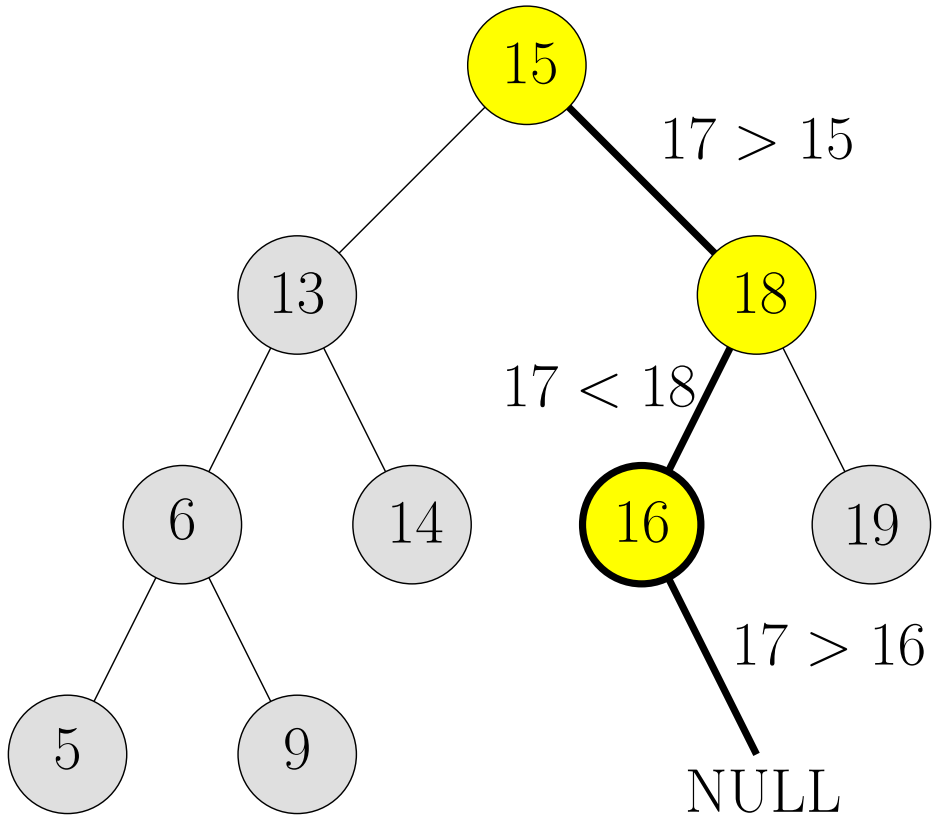
Αναζήτηση σε ΔΔΑ

Αναζήτηση του 17



Αναζήτηση σε ΔΔΑ

Αναζήτηση του 17



Διάσχιση inorder

κανόνας

Επισκεπτόμαστε πρώτα το αριστερό υποδέντρο, μετά τον κόμβο και μετά το δεξιό υποδέντρο

Επισκεπτόμαστε τους κόμβους με αύξουσα σειρά κλειδιού.

Η διάσχιση πέρνει γραμμικό χρόνο $\mathcal{O}(n)$ όπου n τα στοιχεία του δέντρου.

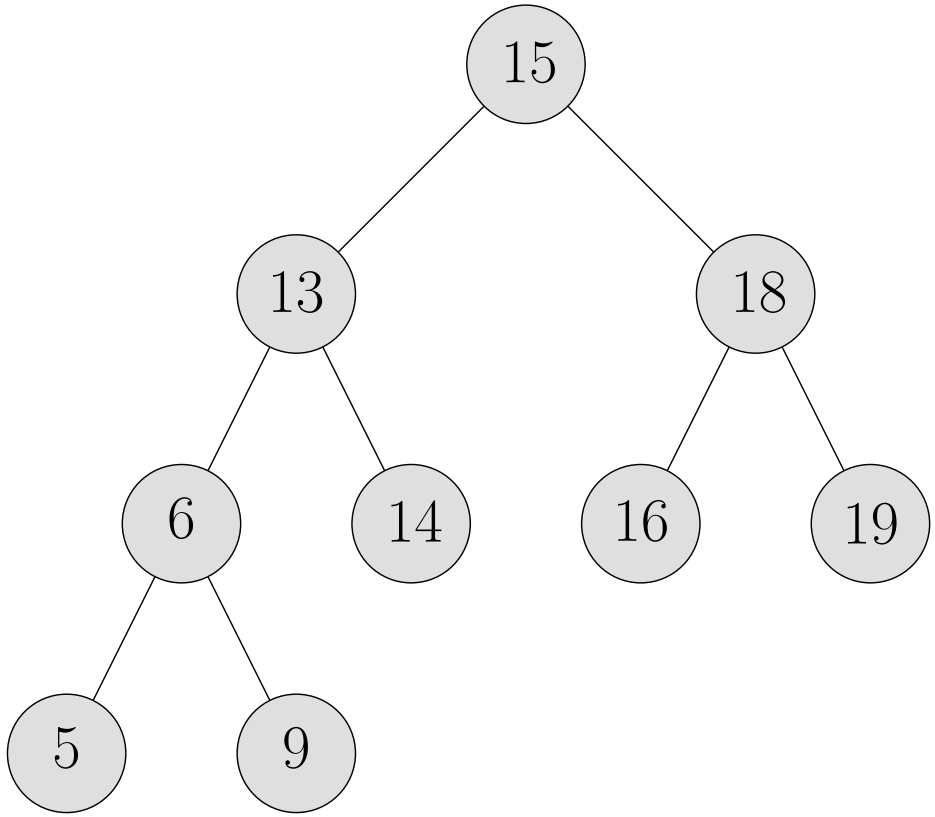
Εισαγωγή σε ΔΔΑ

Αλγόριθμος

- τρέχουμε πρώτα μία αναζήτηση
- εάν βρούμε το κλειδί δεν κάνουμε τίποτα
- αλλιώς προσθέτουμε ένα καινούριο κόμβο με το νέο κλειδί στο σημείο που σταμάτησε η αναζήτηση

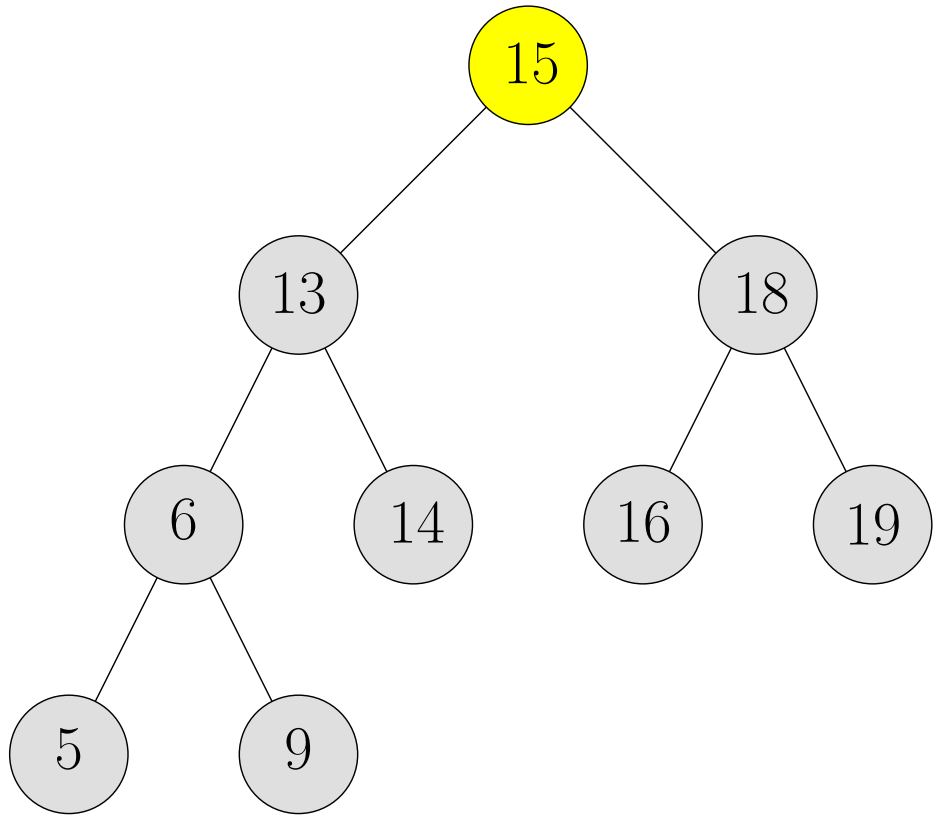
Εισαγωγή σε ΔΔΑ

Εισαγωγή του 17



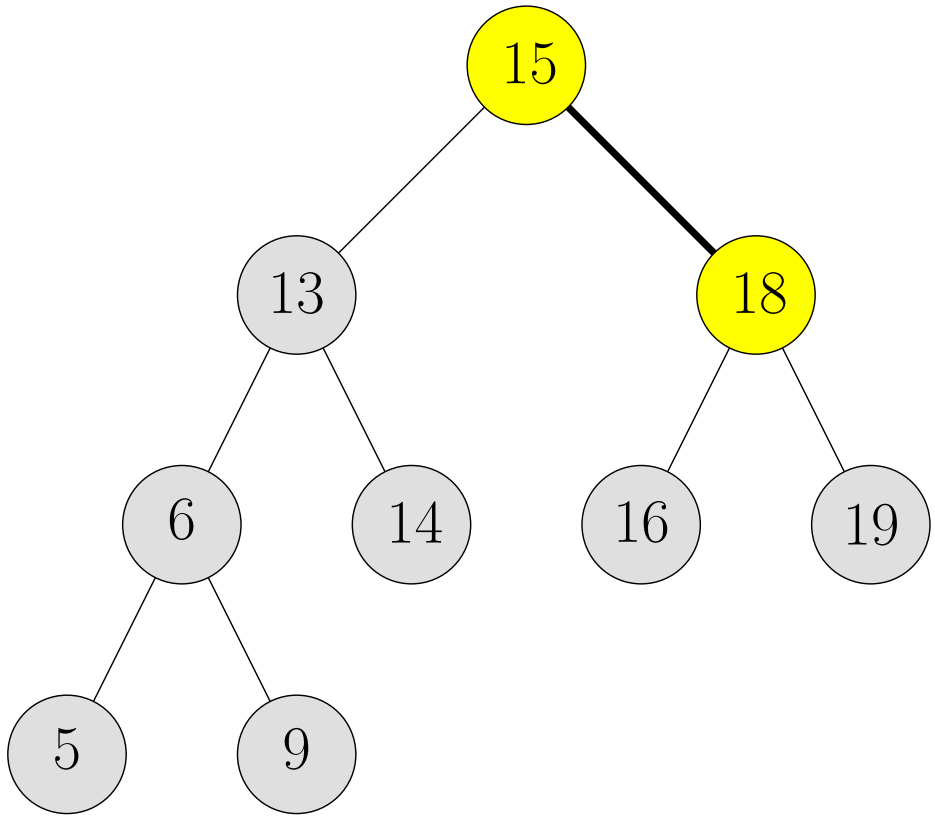
Εισαγωγή σε ΔΔΑ

Εισαγωγή του 17



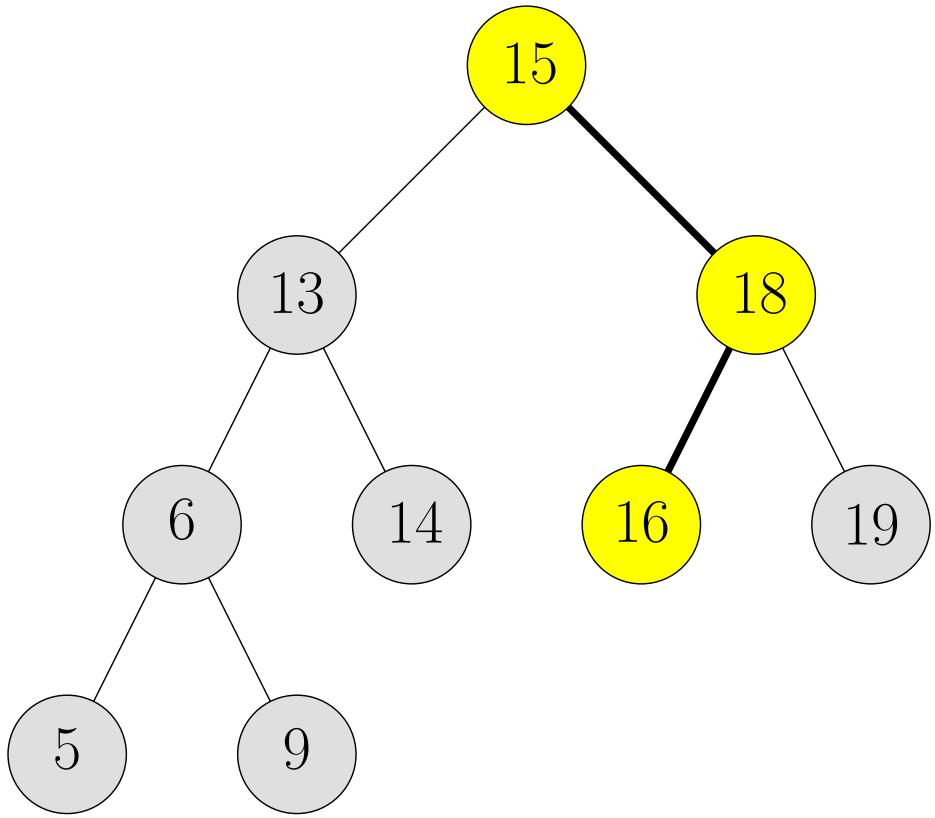
Εισαγωγή σε ΔΔΑ

Εισαγωγή του 17



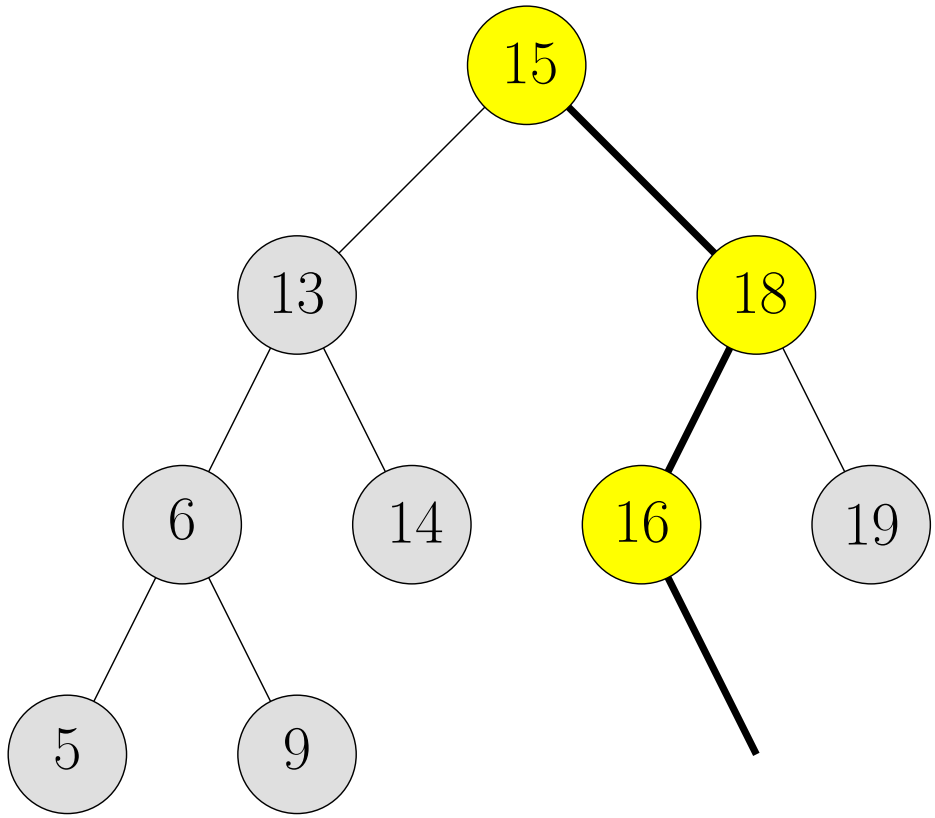
Εισαγωγή σε ΔΔΑ

Εισαγωγή του 17



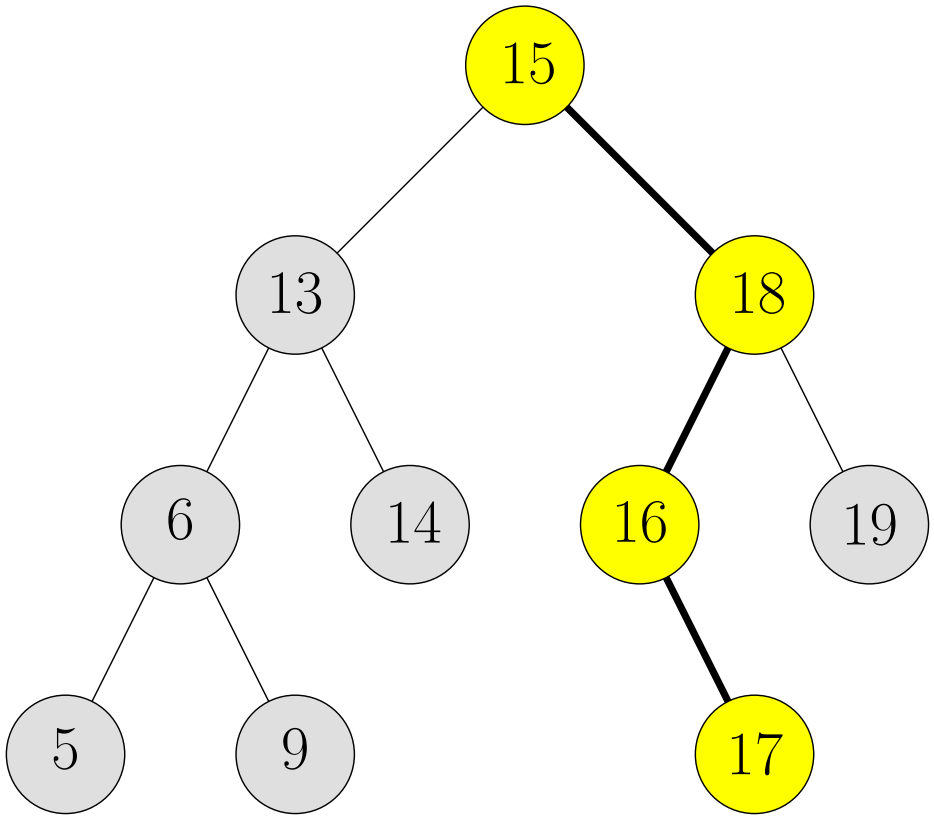
Εισαγωγή σε ΔΔΑ

Εισαγωγή του 17



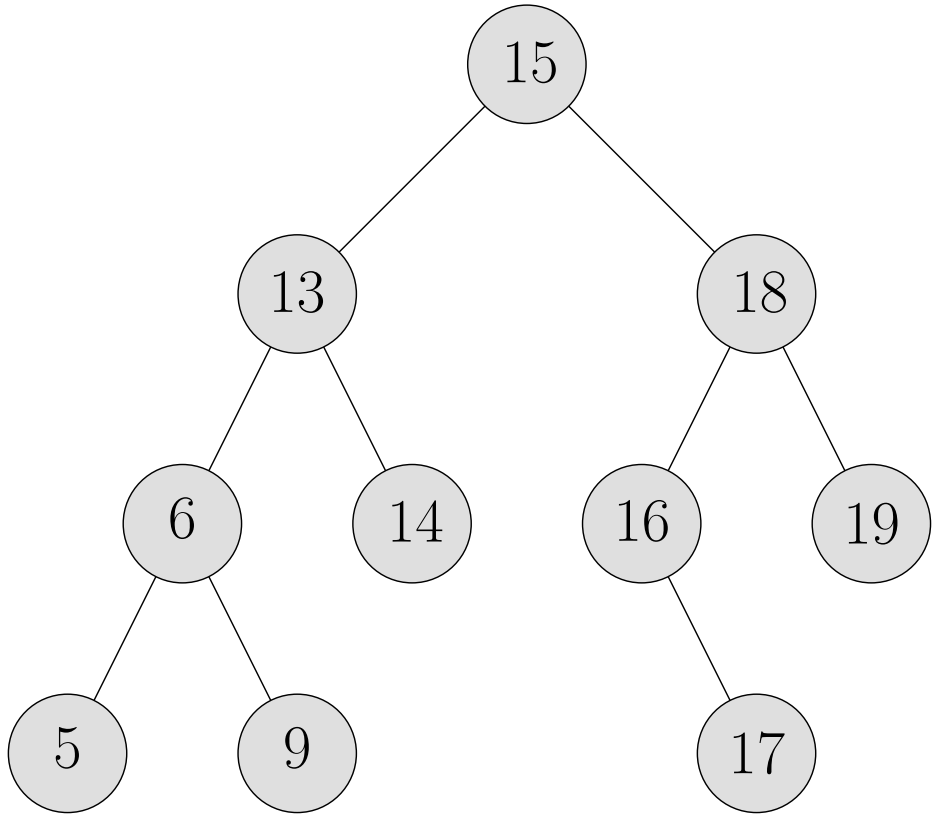
Εισαγωγή σε ΔΔΑ

Εισαγωγή του 17



Εισαγωγή σε ΔΔΑ

Εισαγωγή του 17



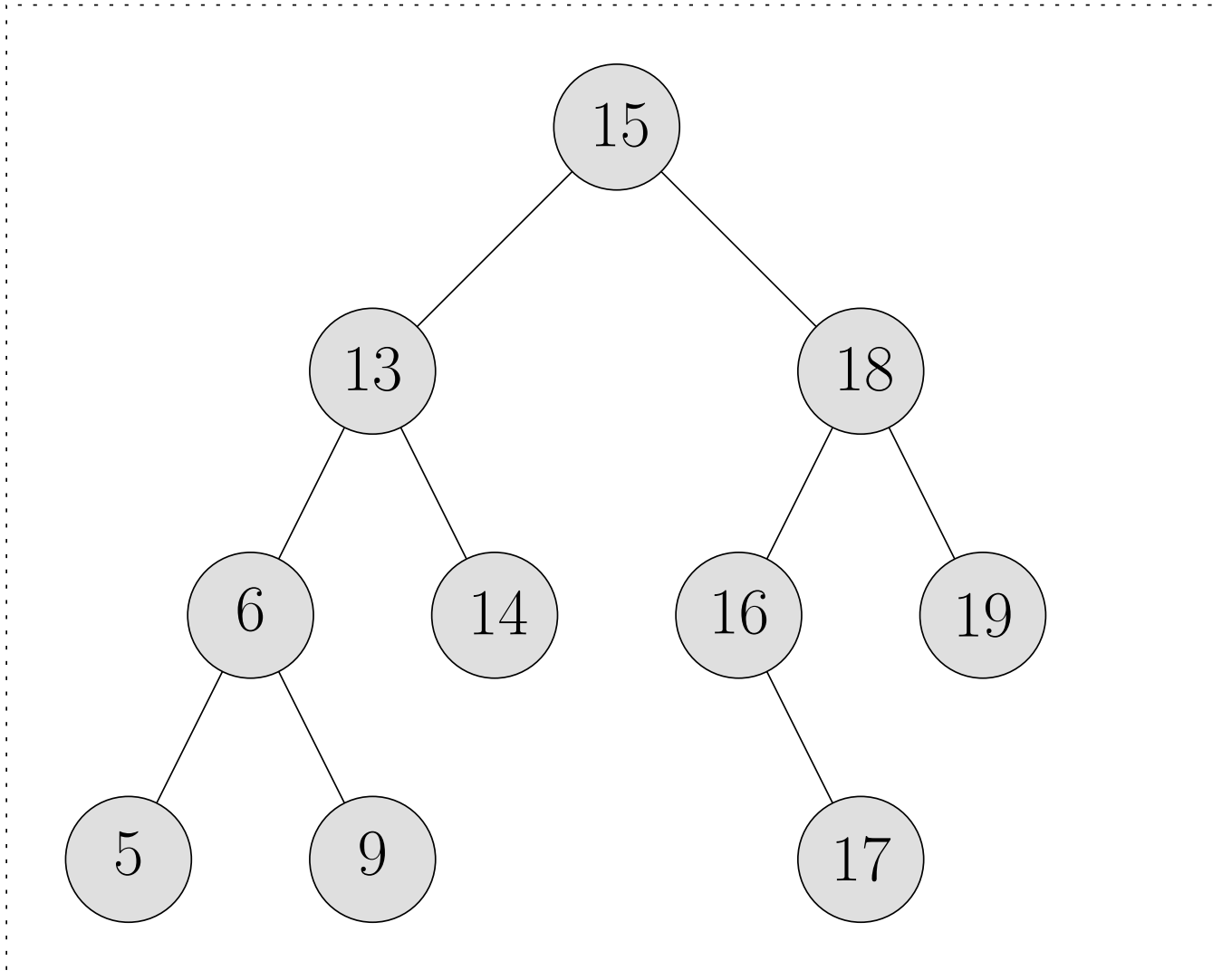
Διαγραφή σε ΔΔΑ

Αλγόριθμος

- τρέχουμε πρώτα μία αναζήτηση
- εάν δεν βρούμε το κλειδί δεν κάνουμε τίποτα
- αλλιώς υπάρχουν 3 περιπτώσεις ανάλογα με τον αριθμό των παιδιών του κόμβου που πρέπει να αφαιρέσουμε

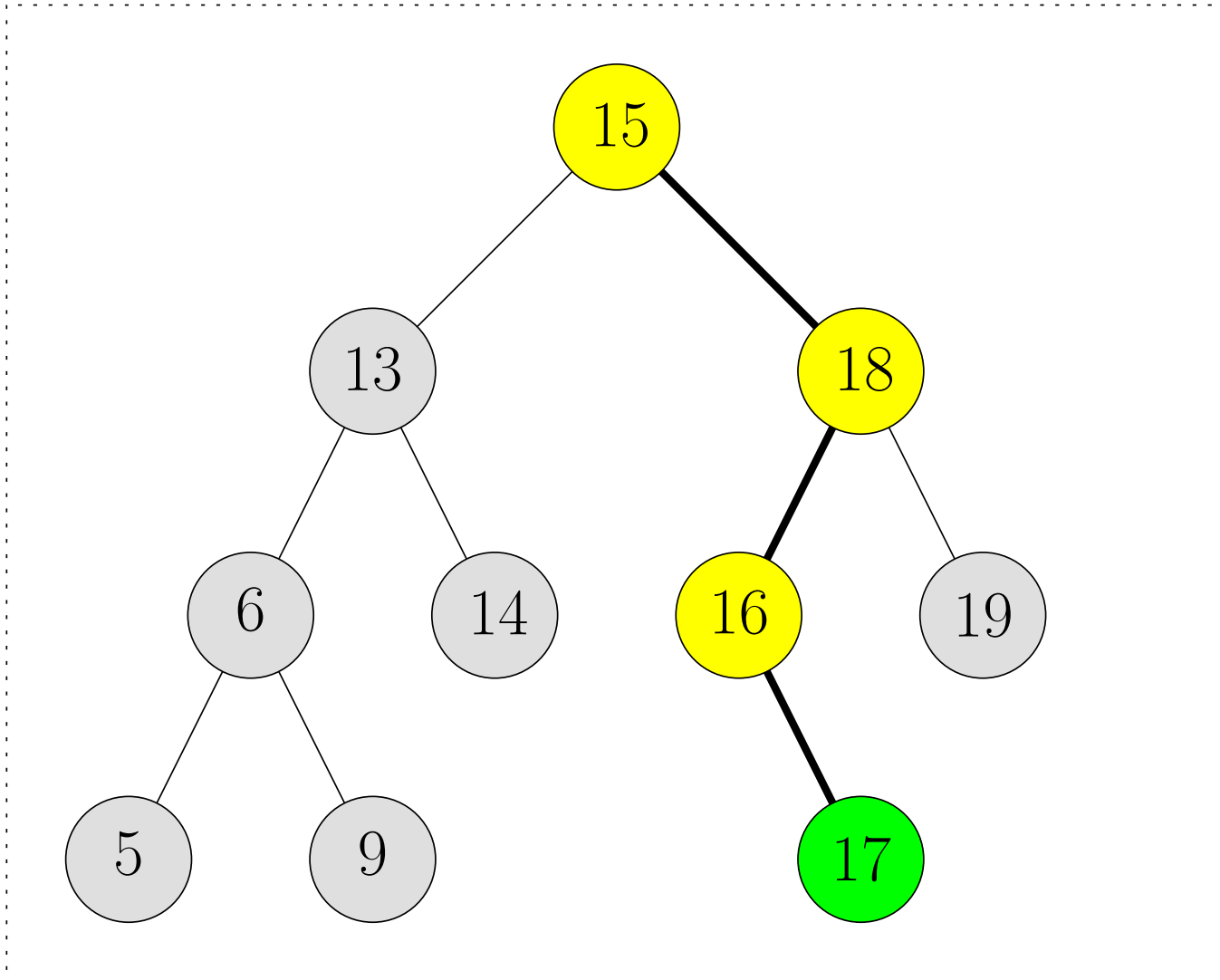
Διαγραφή σε ΔΔΑ

Περίπτωση 1: Διαγραφή του 17 (κανένα παιδί)



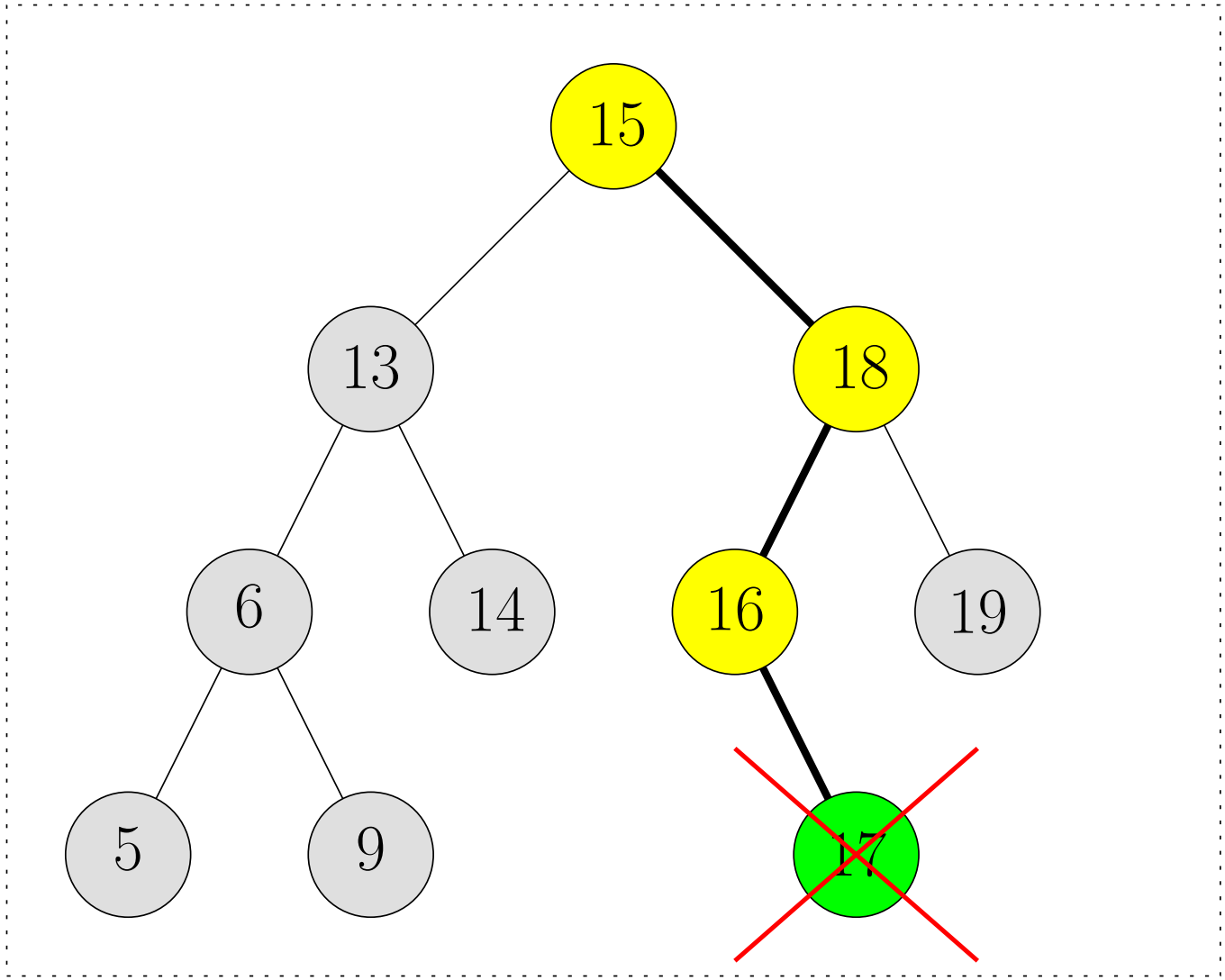
Διαγραφή σε ΔΔΑ

Περίπτωση 1: Διαγραφή του 17 (κανένα παιδί)



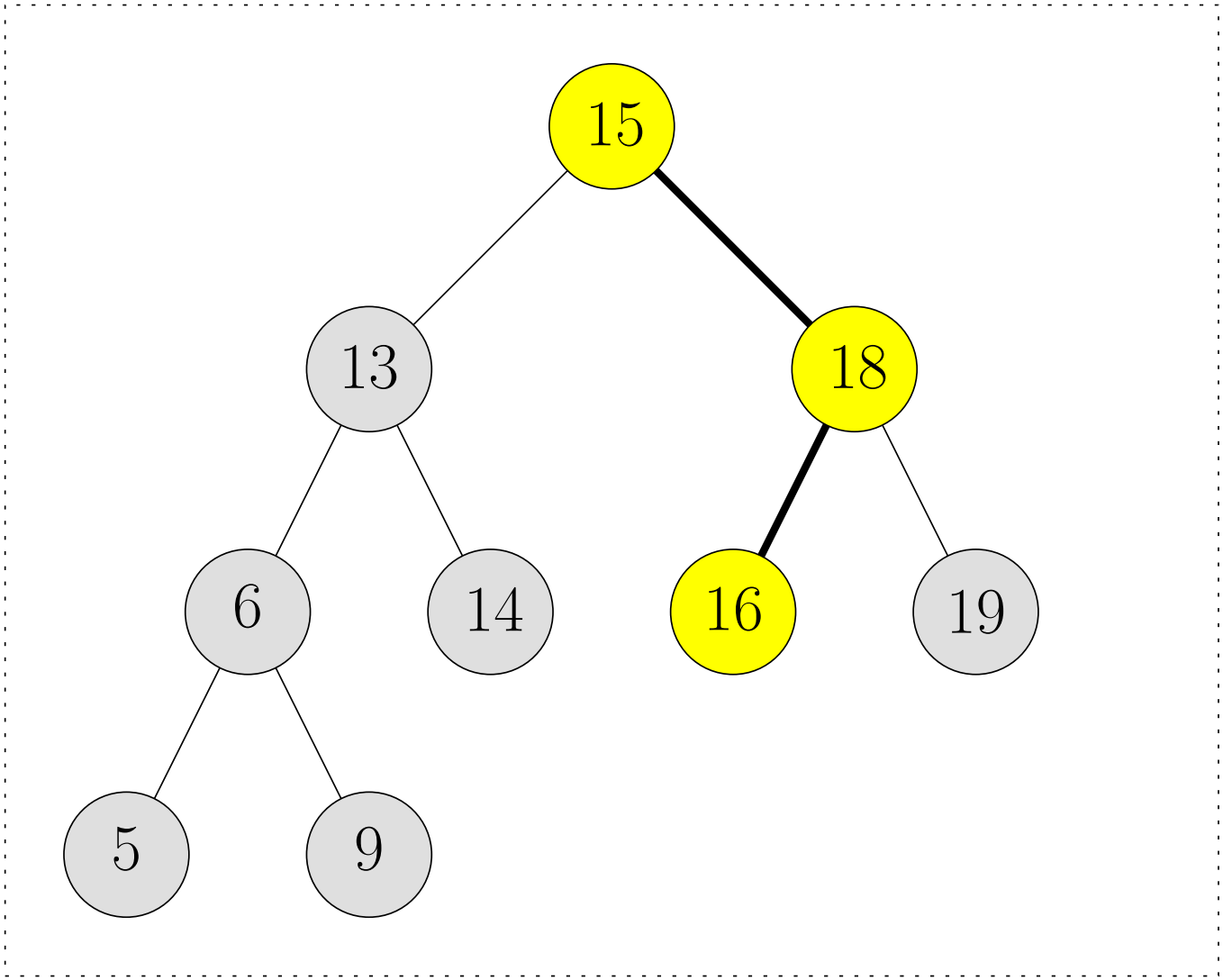
Διαγραφή σε ΔΔΑ

Περίπτωση 1: Διαγραφή του 17 (κανένα παιδί)



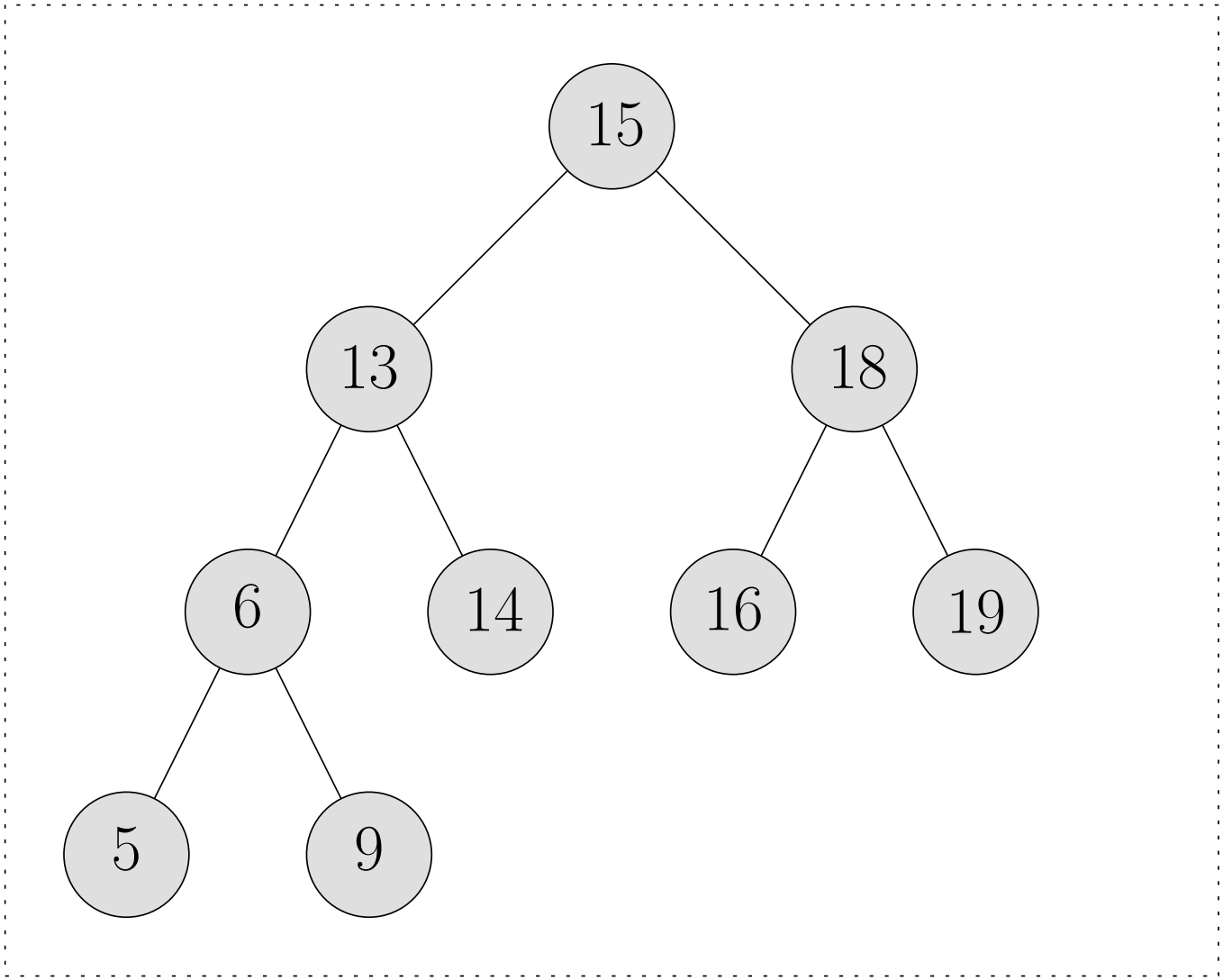
Διαγραφή σε ΔΔΑ

Περίπτωση 1: Διαγραφή του 17 (κανένα παιδί)



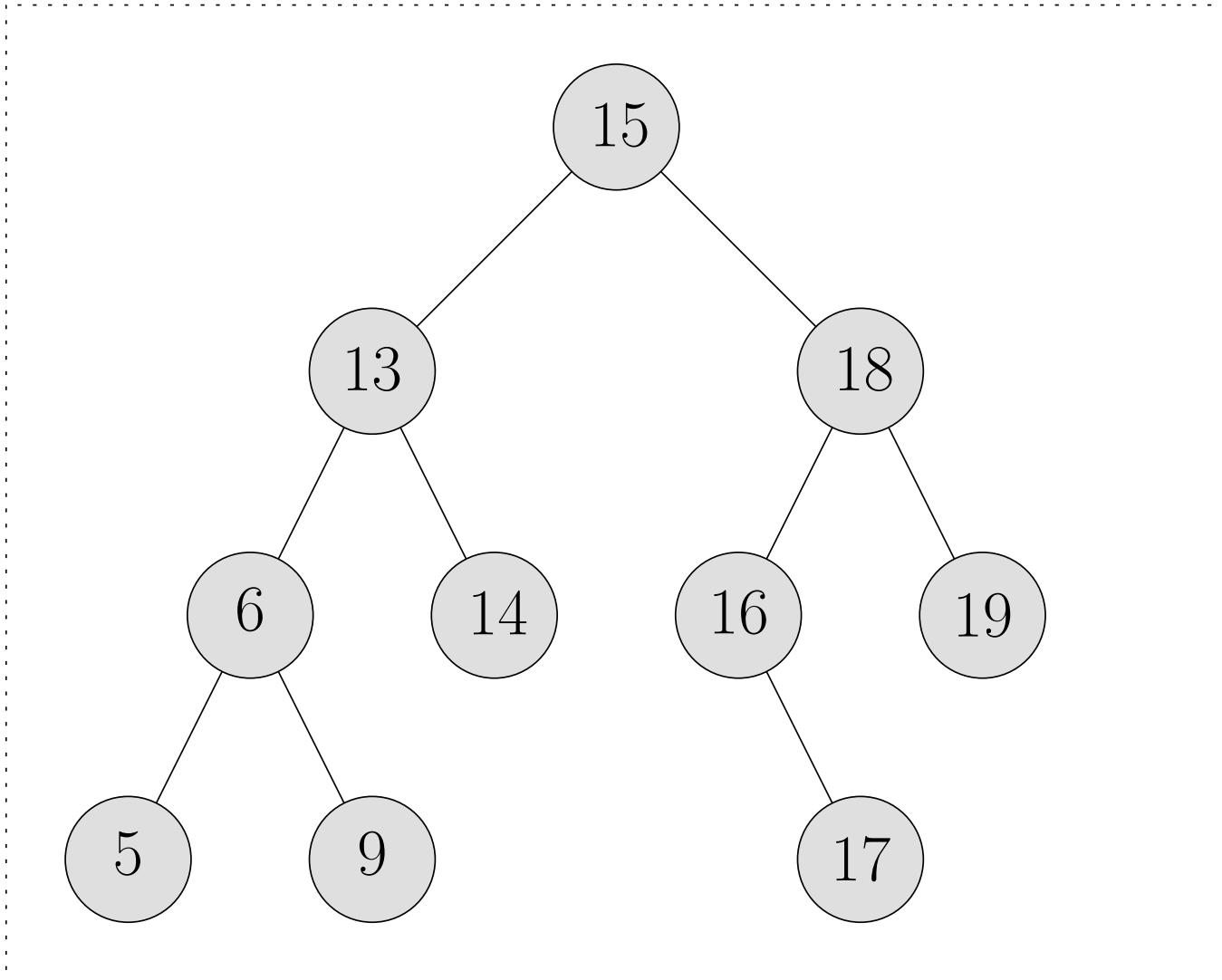
Διαγραφή σε ΔΔΑ

Περίπτωση 1: Διαγραφή του 17 (κανένα παιδί)



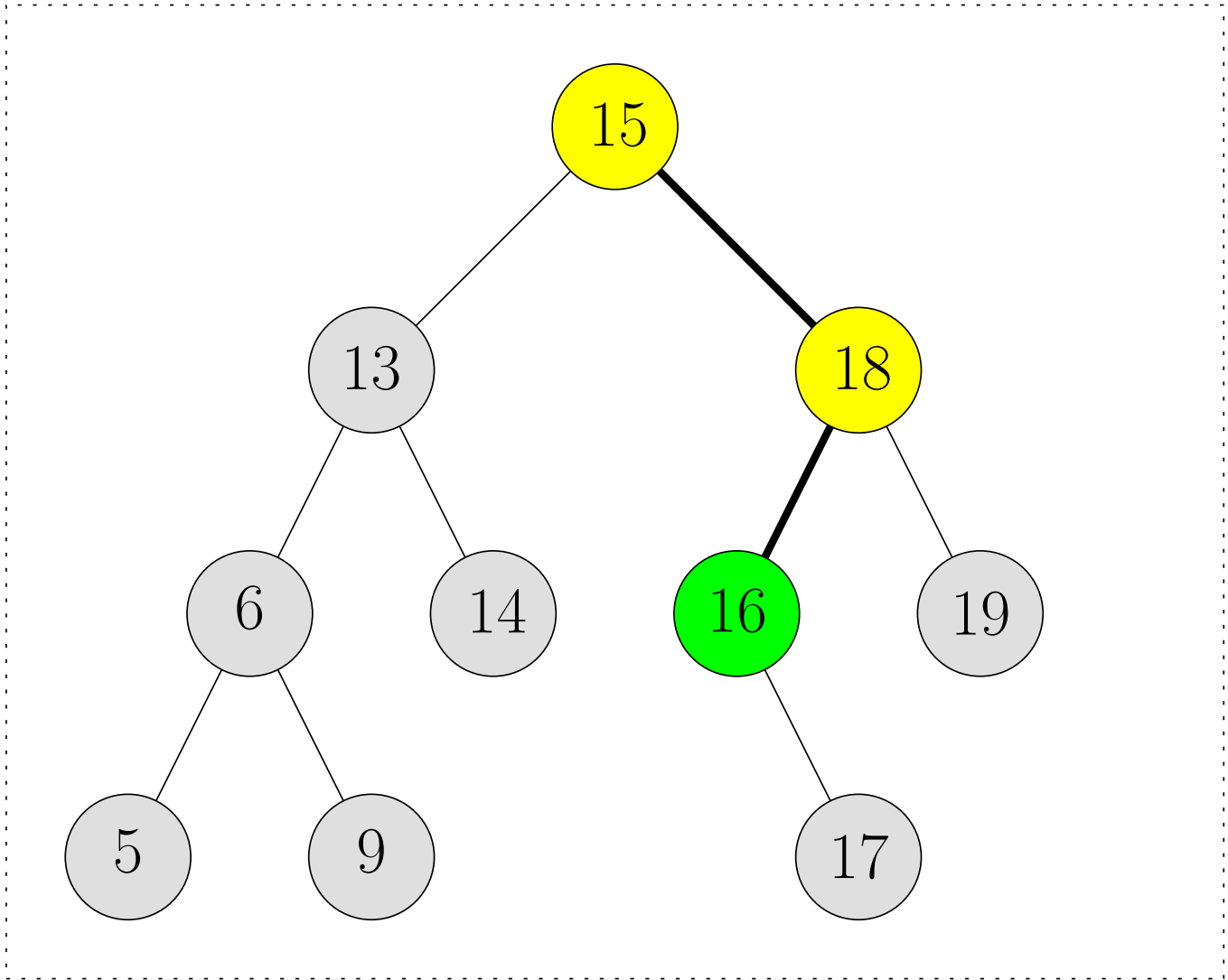
Διαγραφή σε ΔΔΑ

Περίπτωση 2: Διαγραφή του 16 (ένα παιδί)



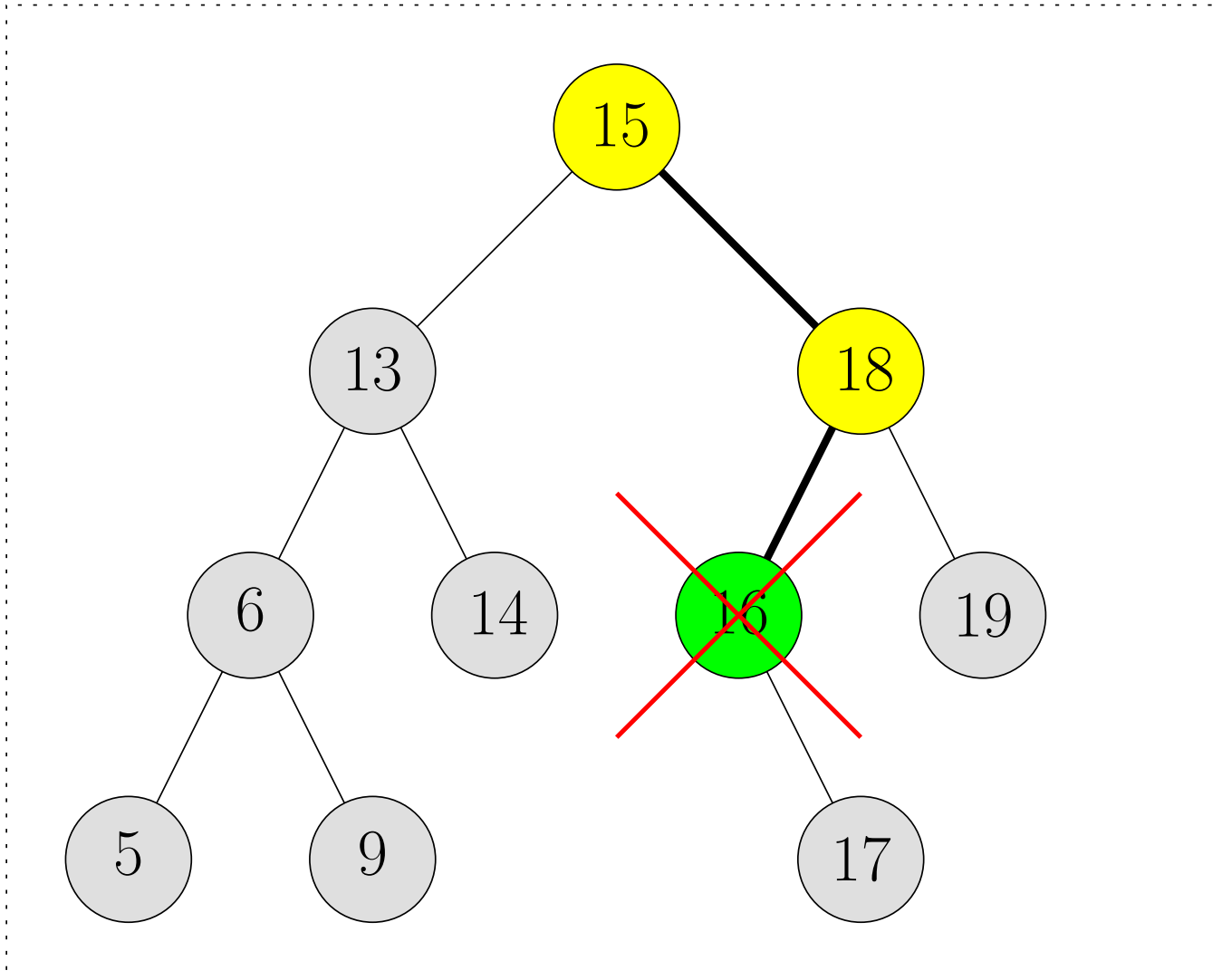
Διαγραφή σε ΔΔΑ

Περίπτωση 2: Διαγραφή του 16 (ένα παιδί)



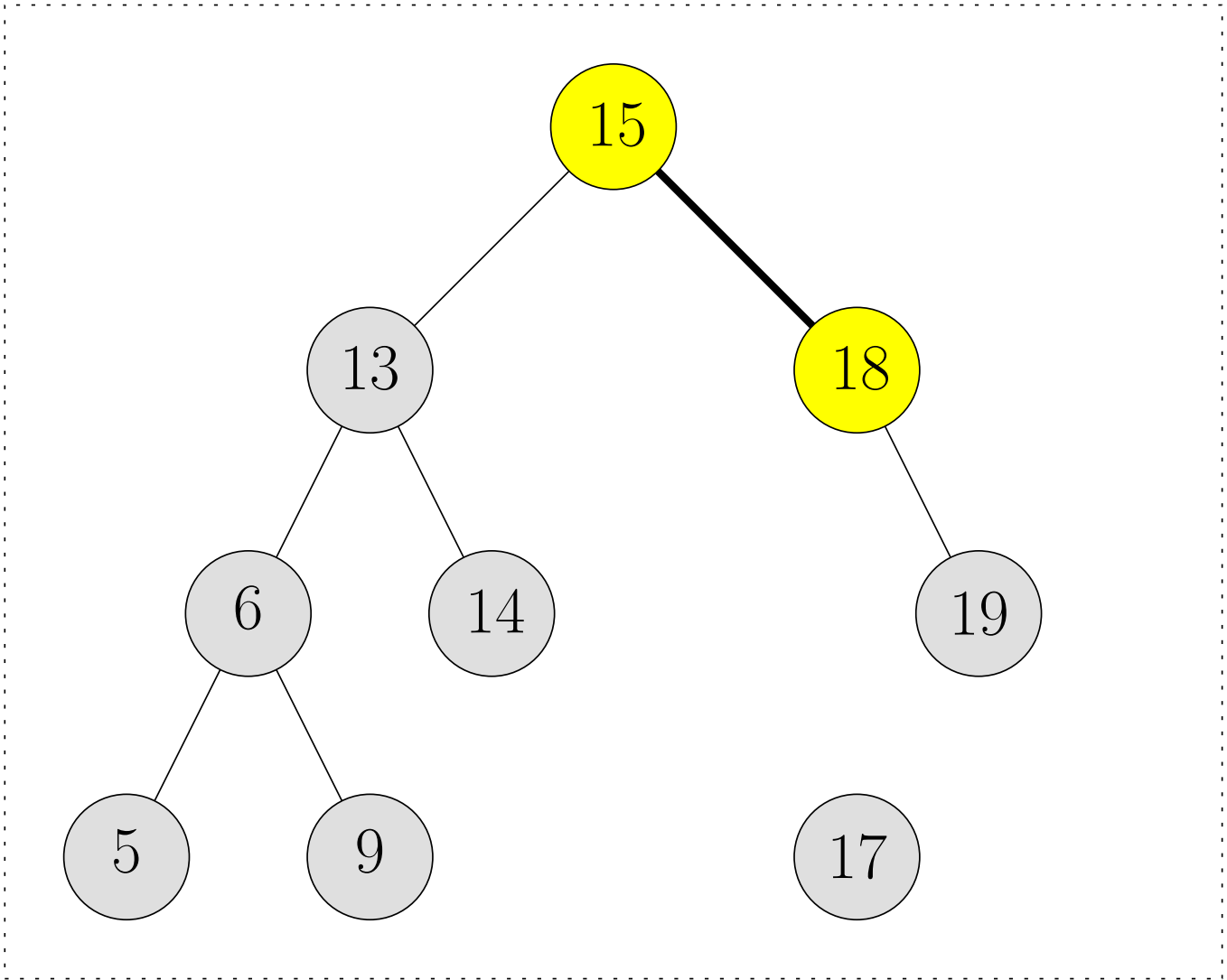
Διαγραφή σε ΔΔΑ

Περίπτωση 2: Διαγραφή του 16 (ένα παιδί)



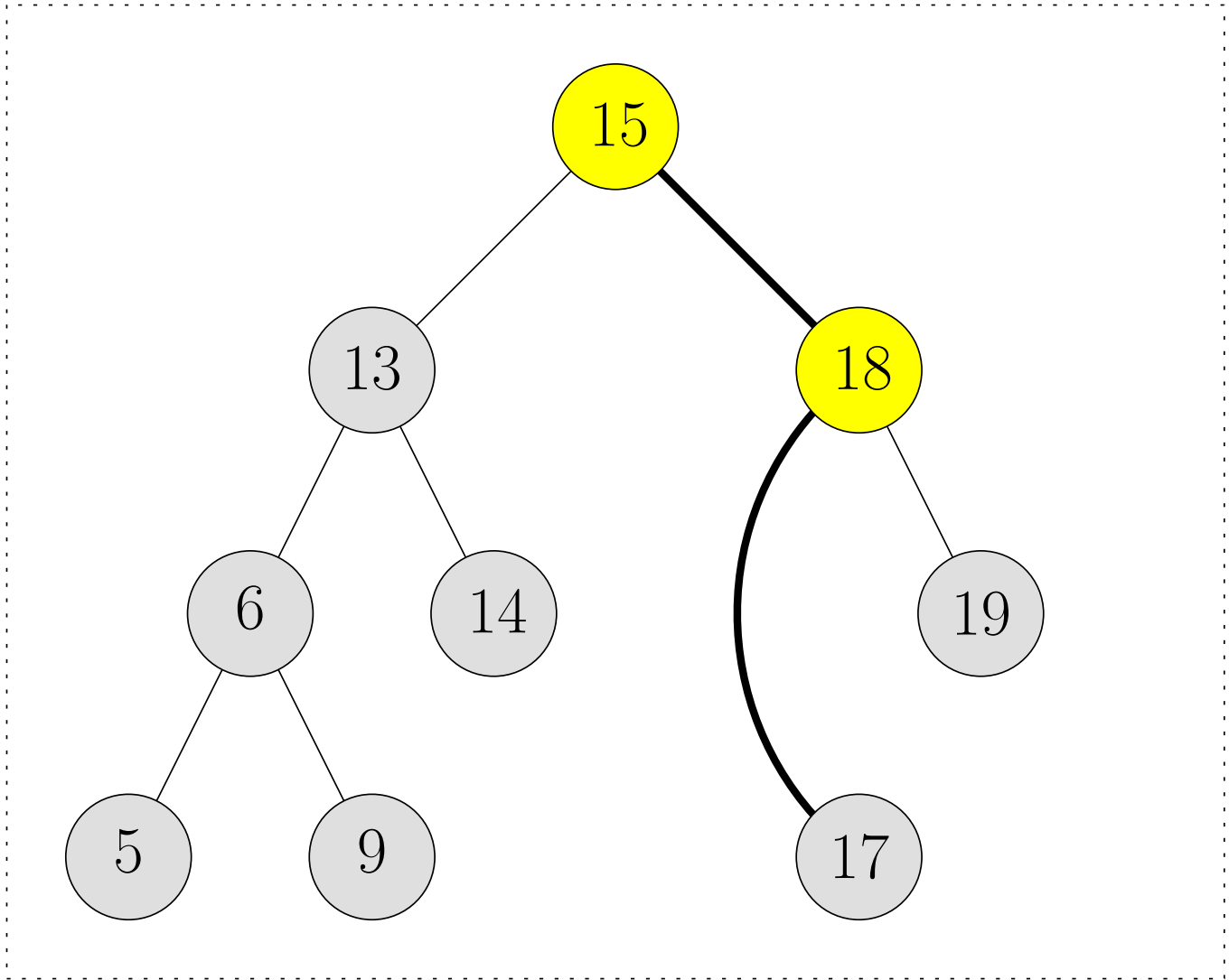
Διαγραφή σε ΔΔΑ

Περίπτωση 2: Διαγραφή του 16 (ένα παιδί)



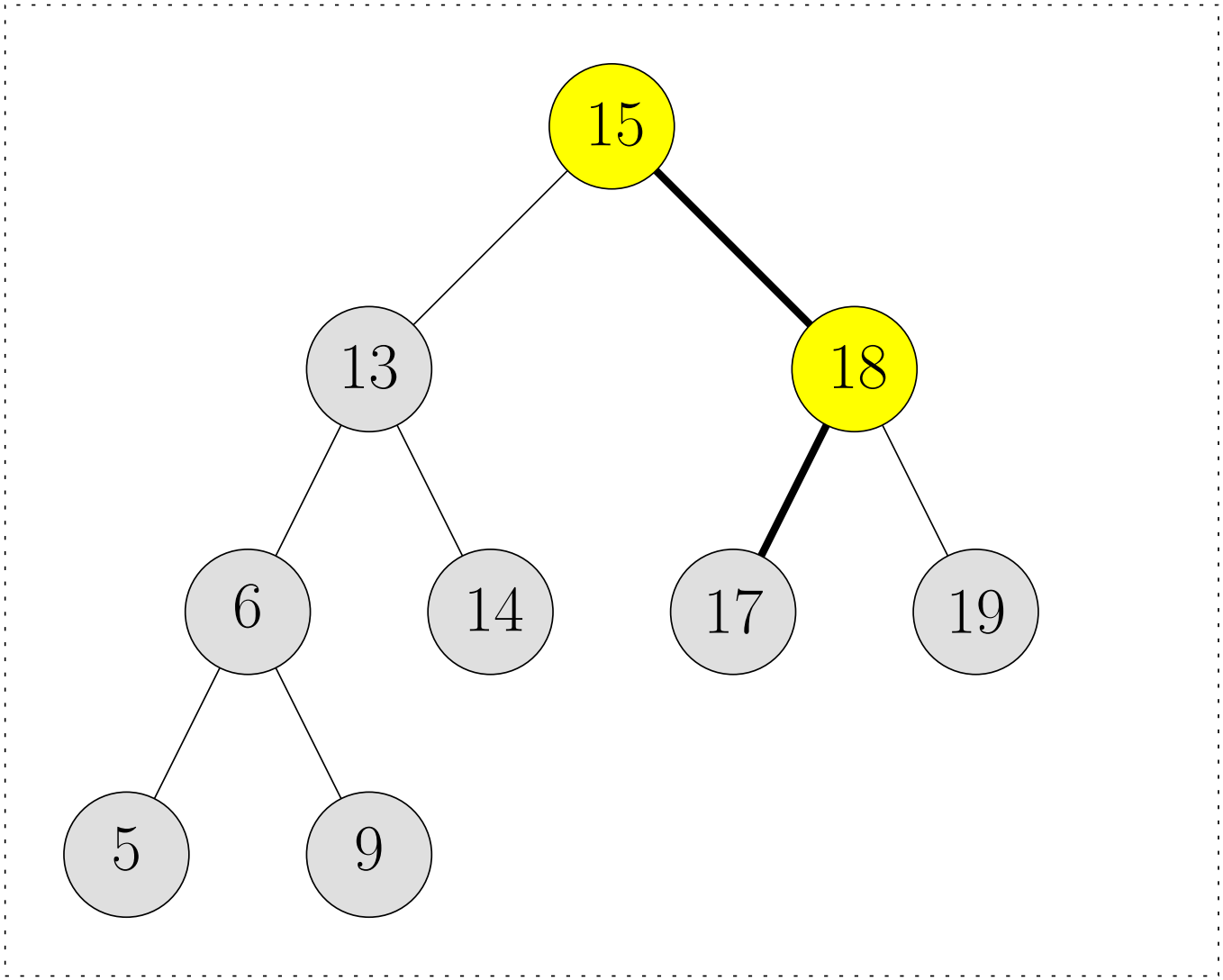
Διαγραφή σε ΔΔΑ

Περίπτωση 2: Διαγραφή του 16 (ένα παιδί)



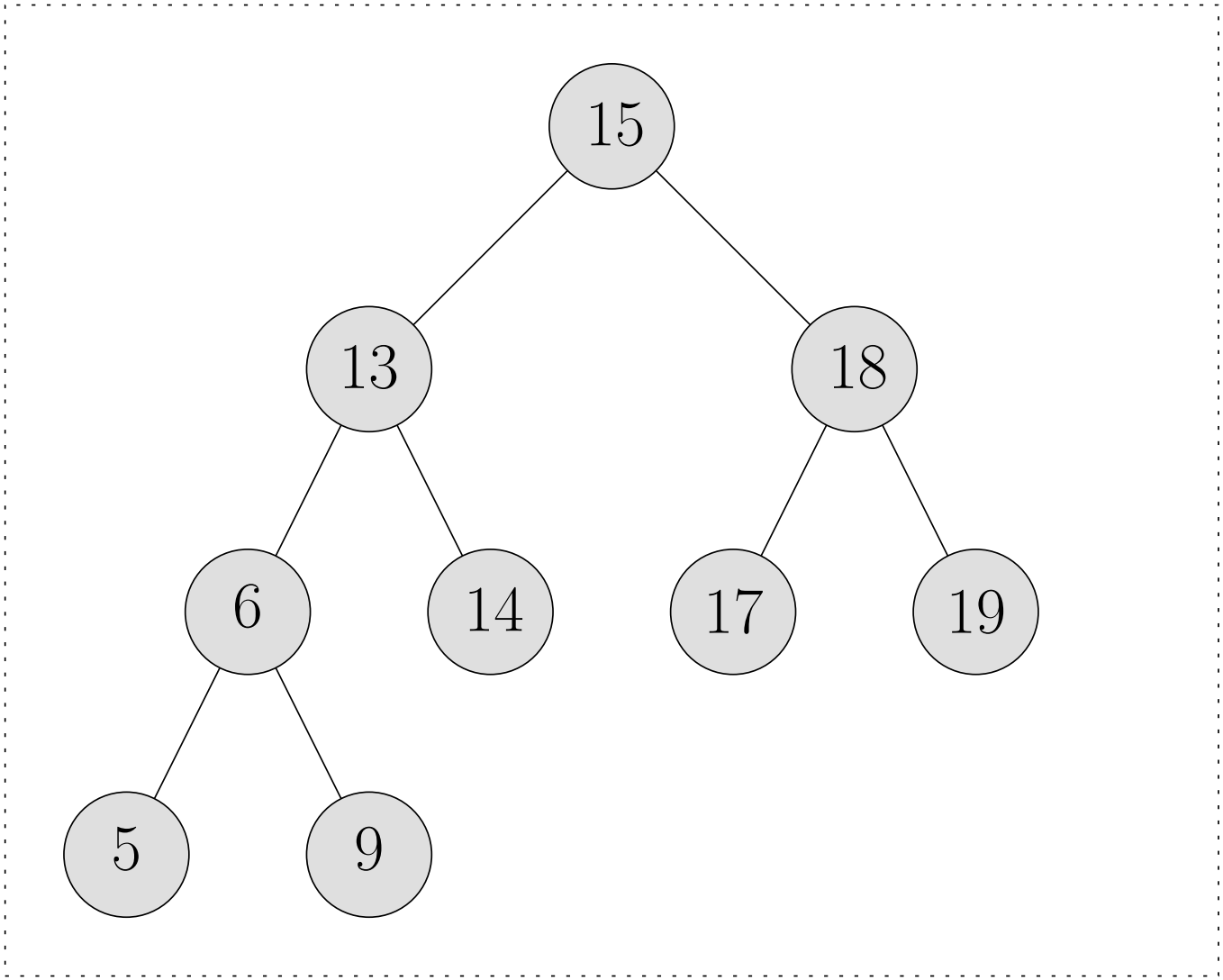
Διαγραφή σε ΔΔΑ

Περίπτωση 2: Διαγραφή του 16 (ένα παιδί)



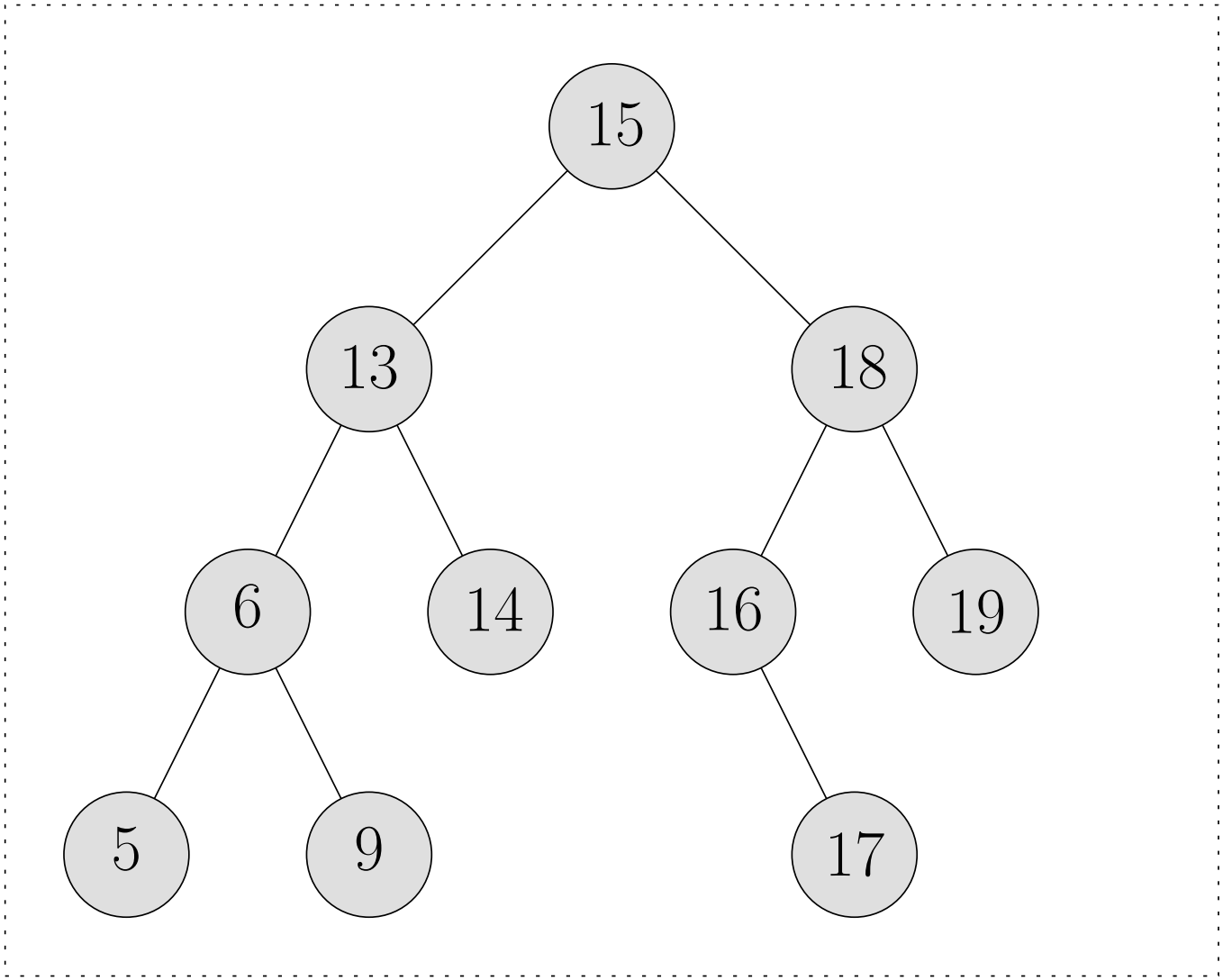
Διαγραφή σε ΔΔΑ

Περίπτωση 2: Διαγραφή του 16 (ένα παιδί)



Διαγραφή σε ΔΔΑ

Περίπτωση 3: Διαγραφή του 15 (δύο παιδιά)



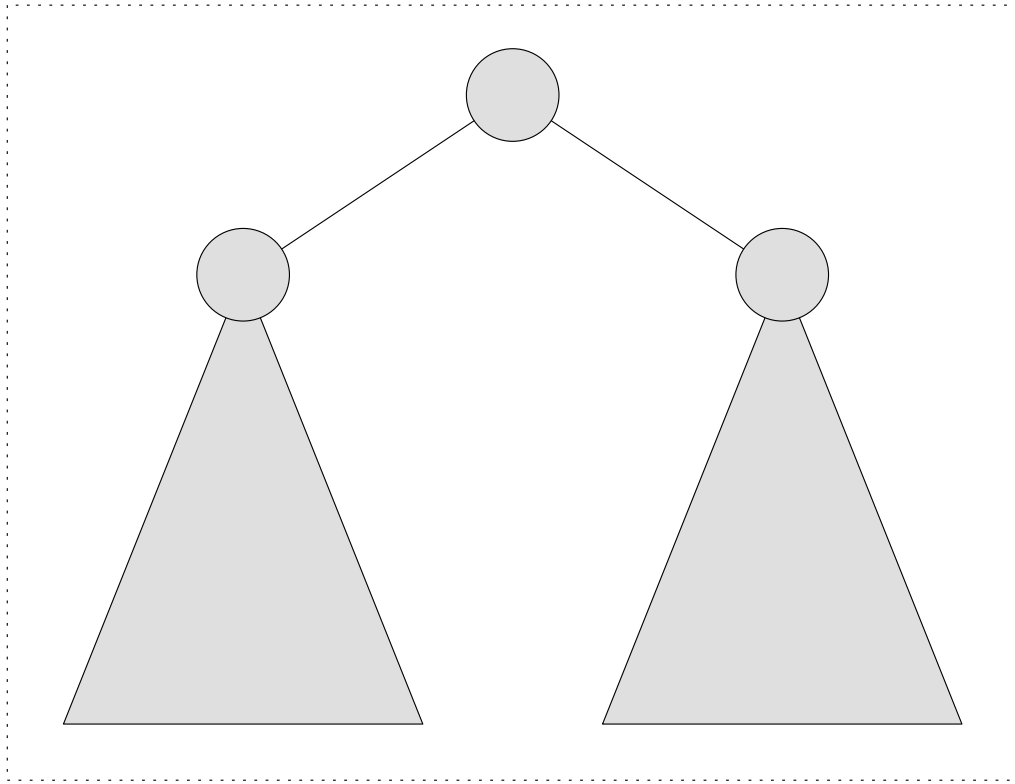
Υπολογισμός του επόμενου κλειδιού της ρίζας

Χρήσιμη ρουτίνα για την διαγραφή

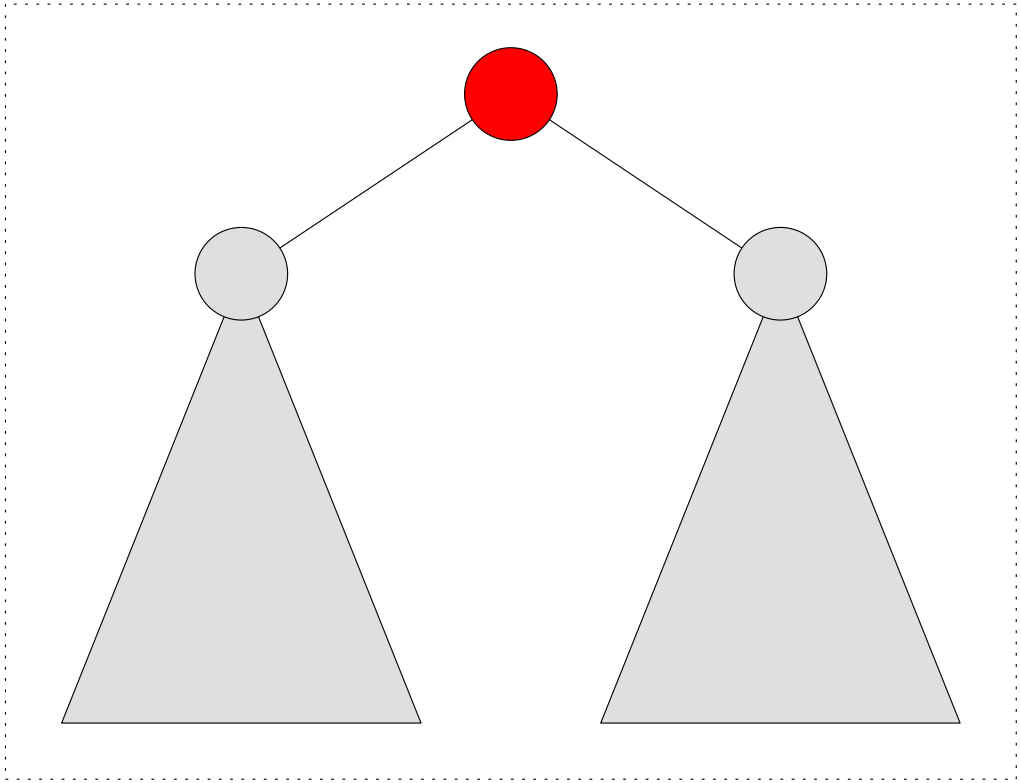
Το πρόβλημα

Έχοντας ένα υποδέντρο θέλουμε να βρούμε τον κόμβο με το αμέσως επόμενο κλειδί από τη ρίζα του υποδέντρου.

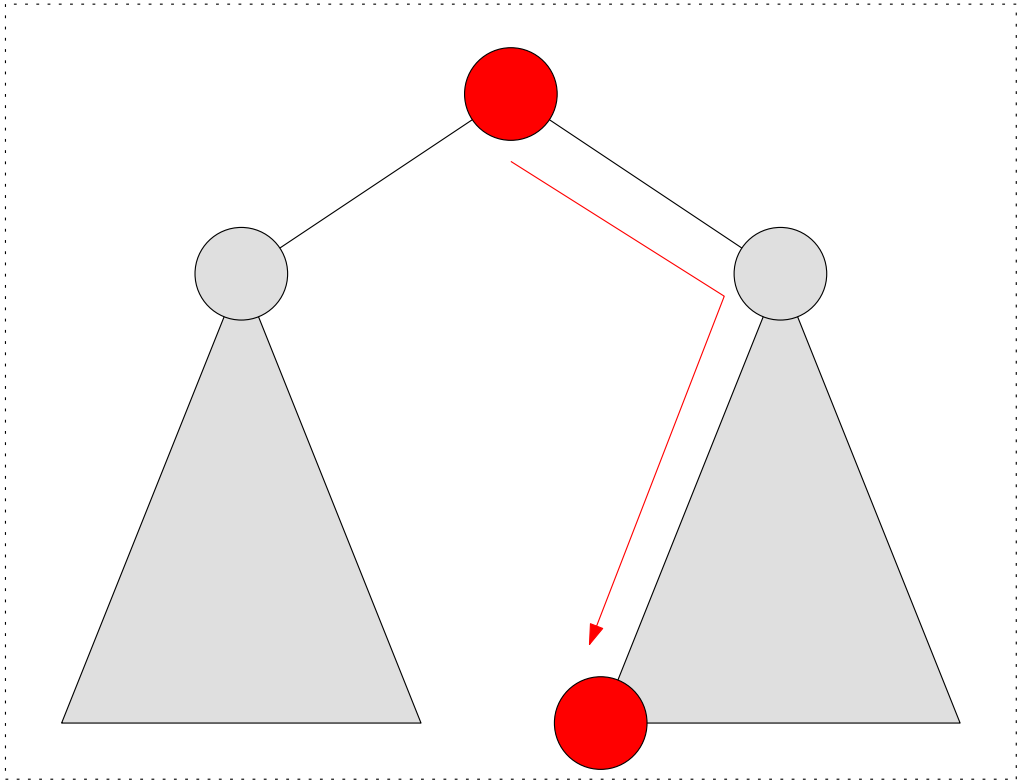
Υπολογισμός του επόμενου κλειδιού της ρίζας



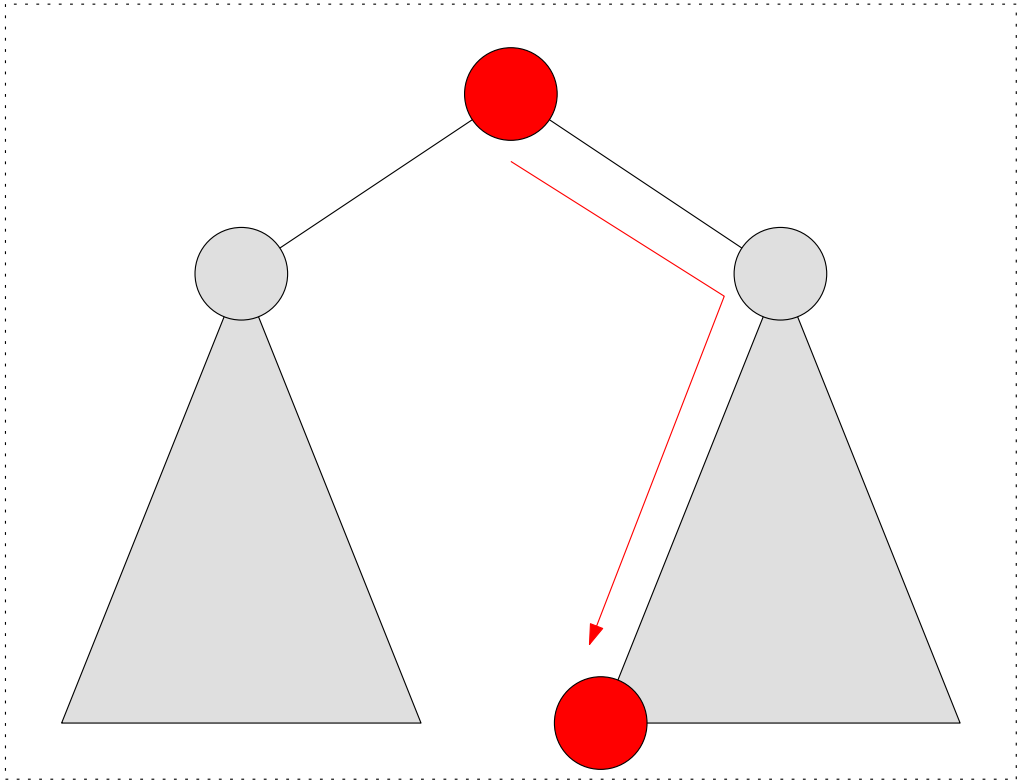
Υπολογισμός του επόμενου κλειδιού της ρίζας



Υπολογισμός του επόμενου κλειδιού της ρίζας



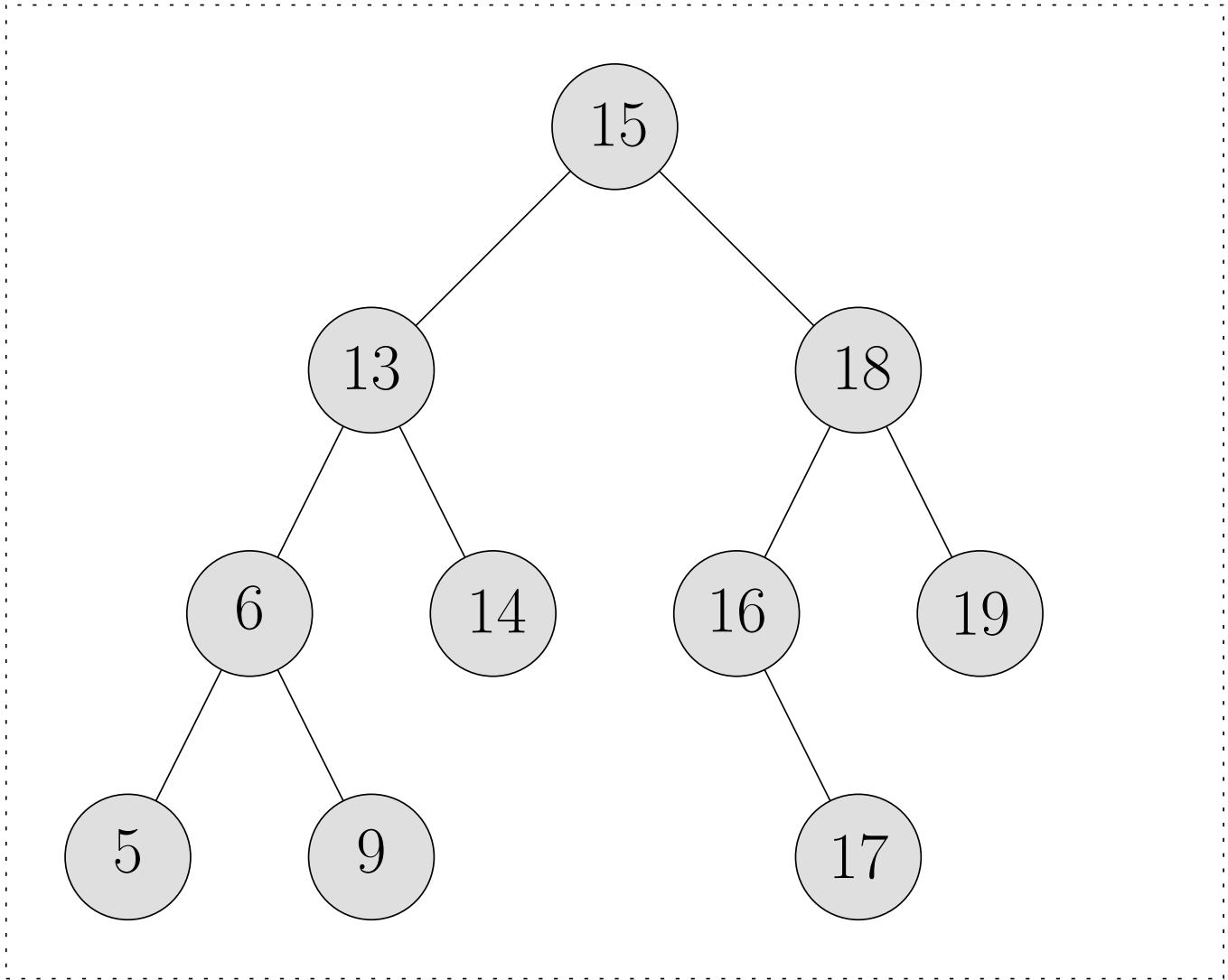
Υπολογισμός του επόμενου κλειδιού της ρίζας



Ο επόμενος κόμβος της ρίζας έχει το πολύ ένα παιδί.

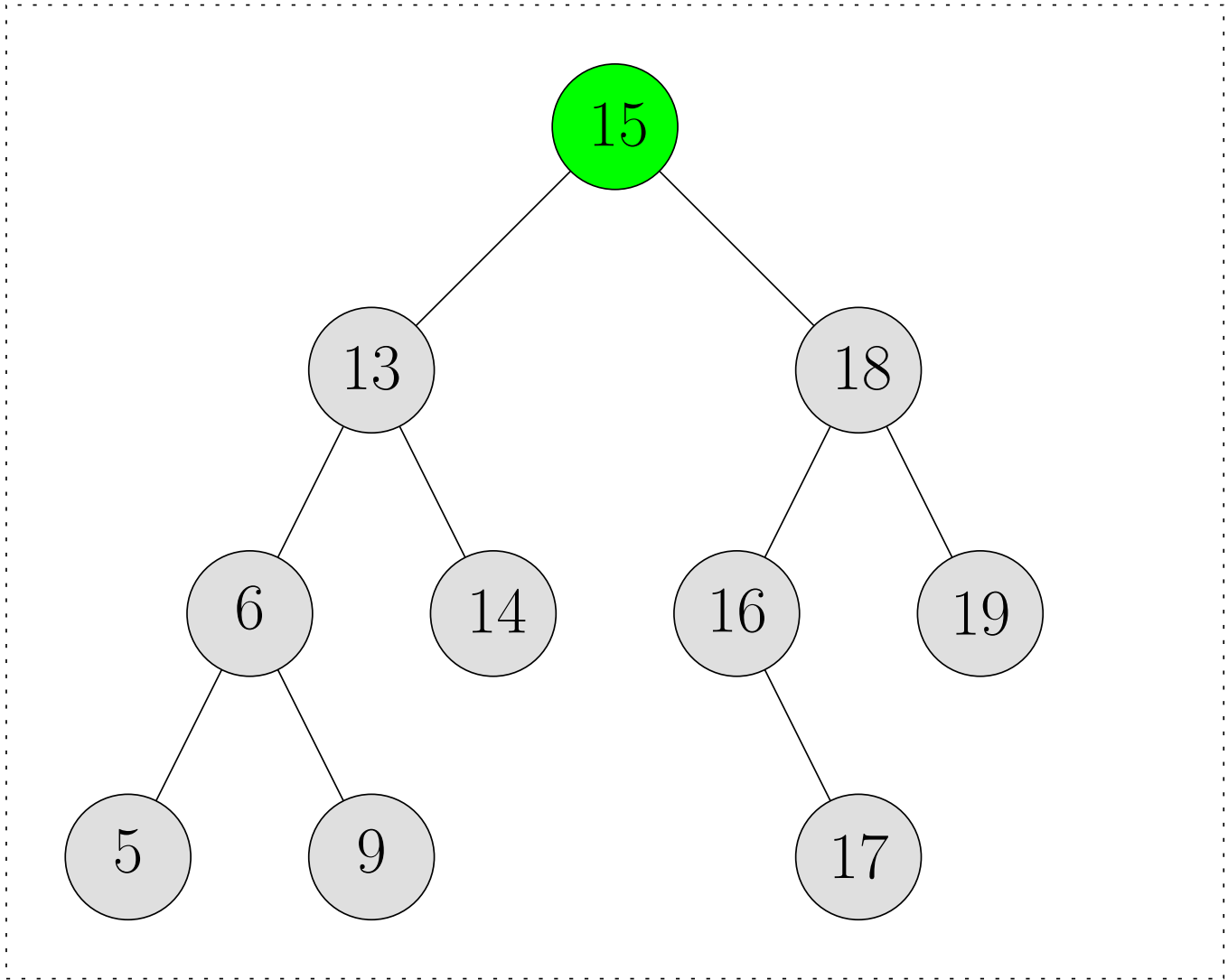
Διαγραφή σε ΔΔΑ

Περίπτωση 3: Διαγραφή του 15 (δύο παιδιά)



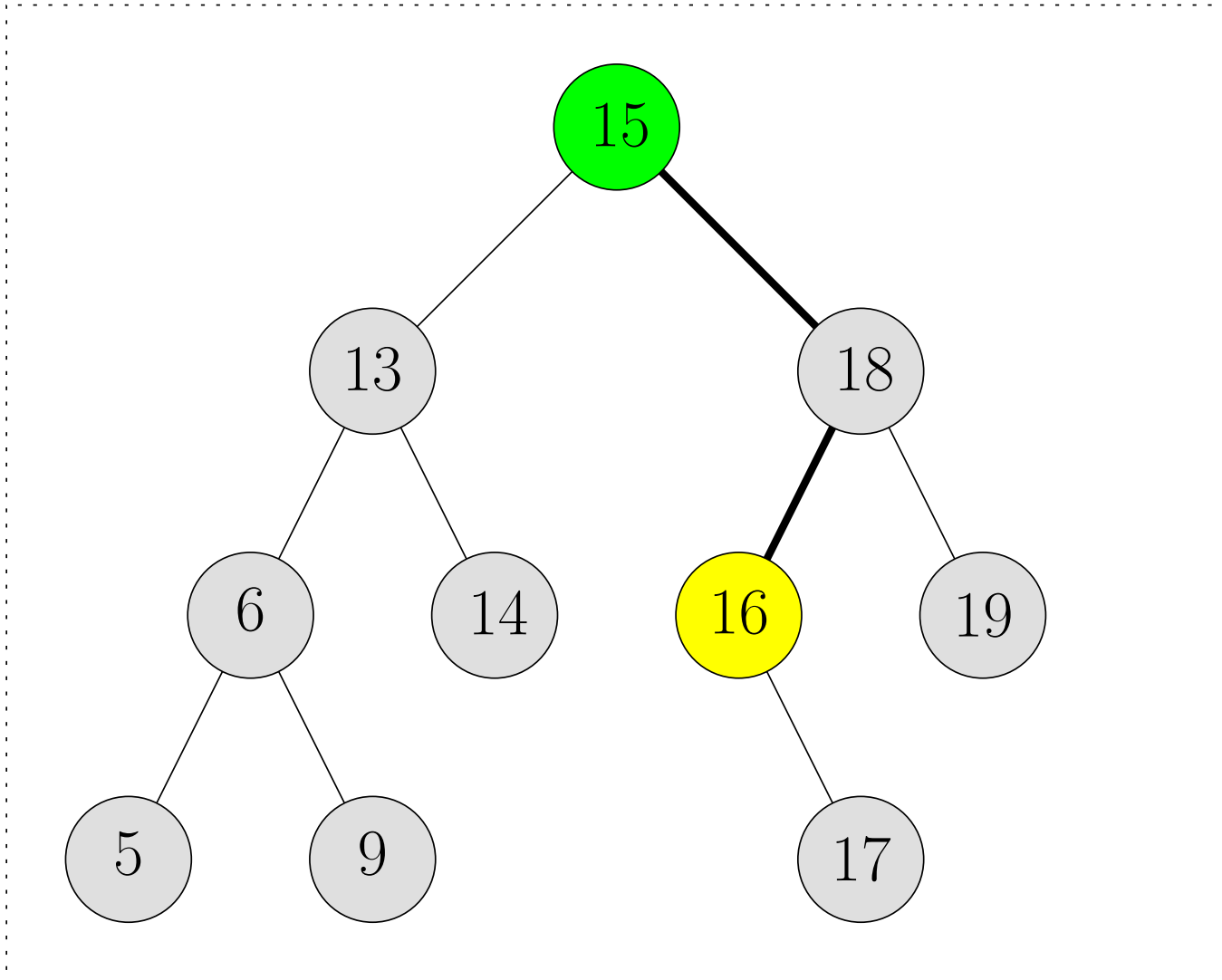
Διαγραφή σε ΔΔΑ

Περίπτωση 3: Διαγραφή του 15 (δύο παιδιά)



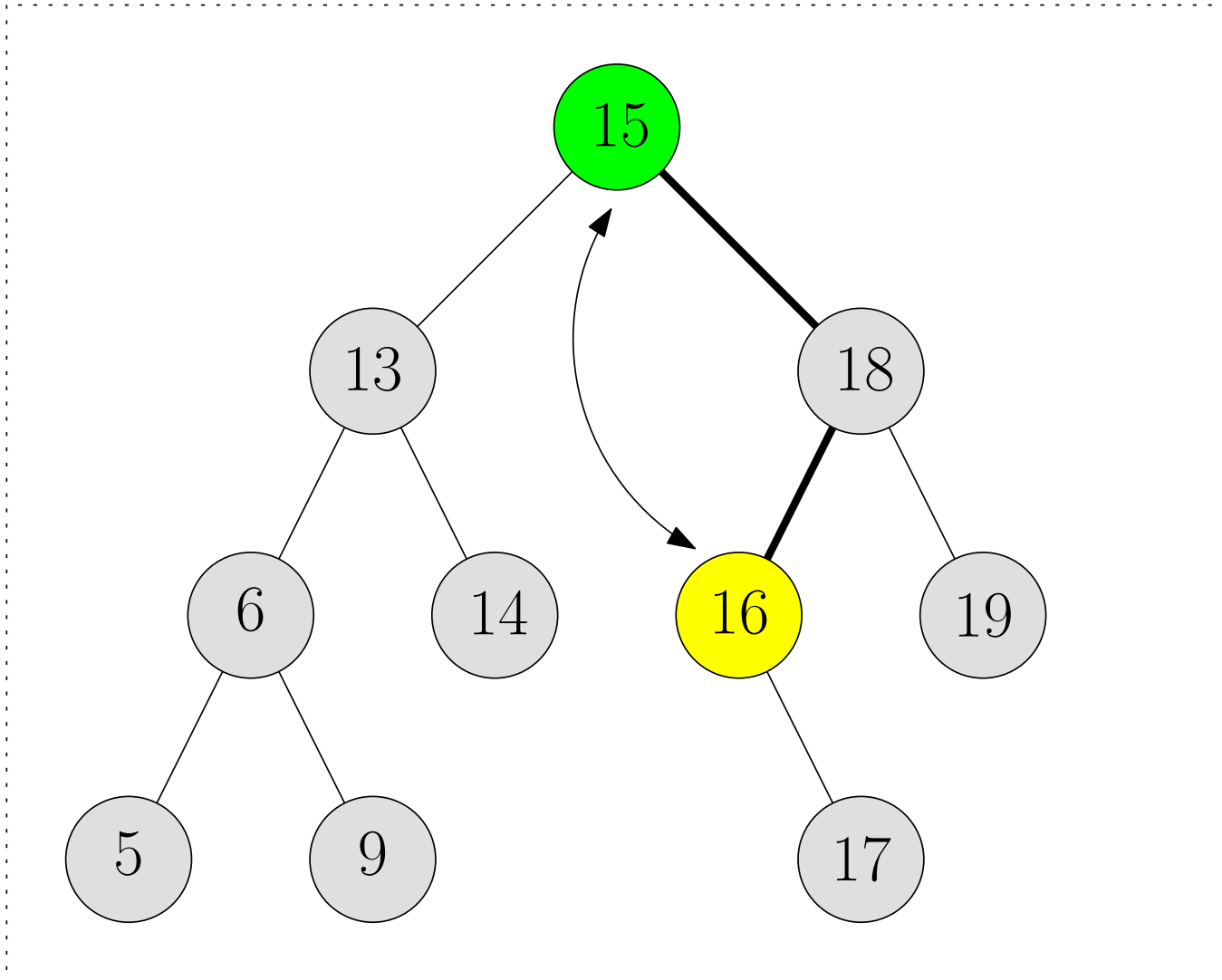
Διαγραφή σε ΔΔΑ

Περίπτωση 3: Διαγραφή του 15 (δύο παιδιά)



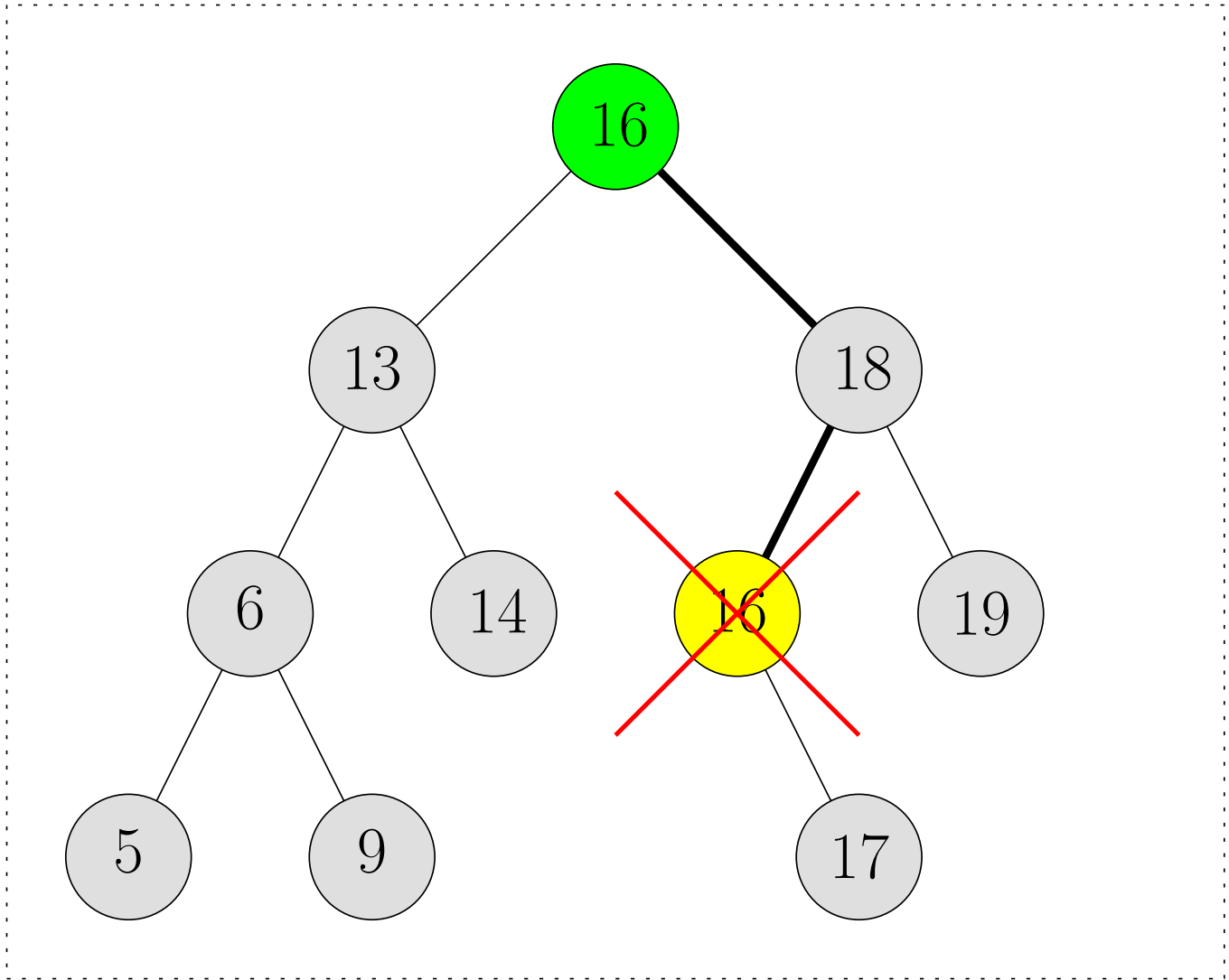
Διαγραφή σε ΔΔΑ

Περίπτωση 3: Διαγραφή του 15 (δύο παιδιά)



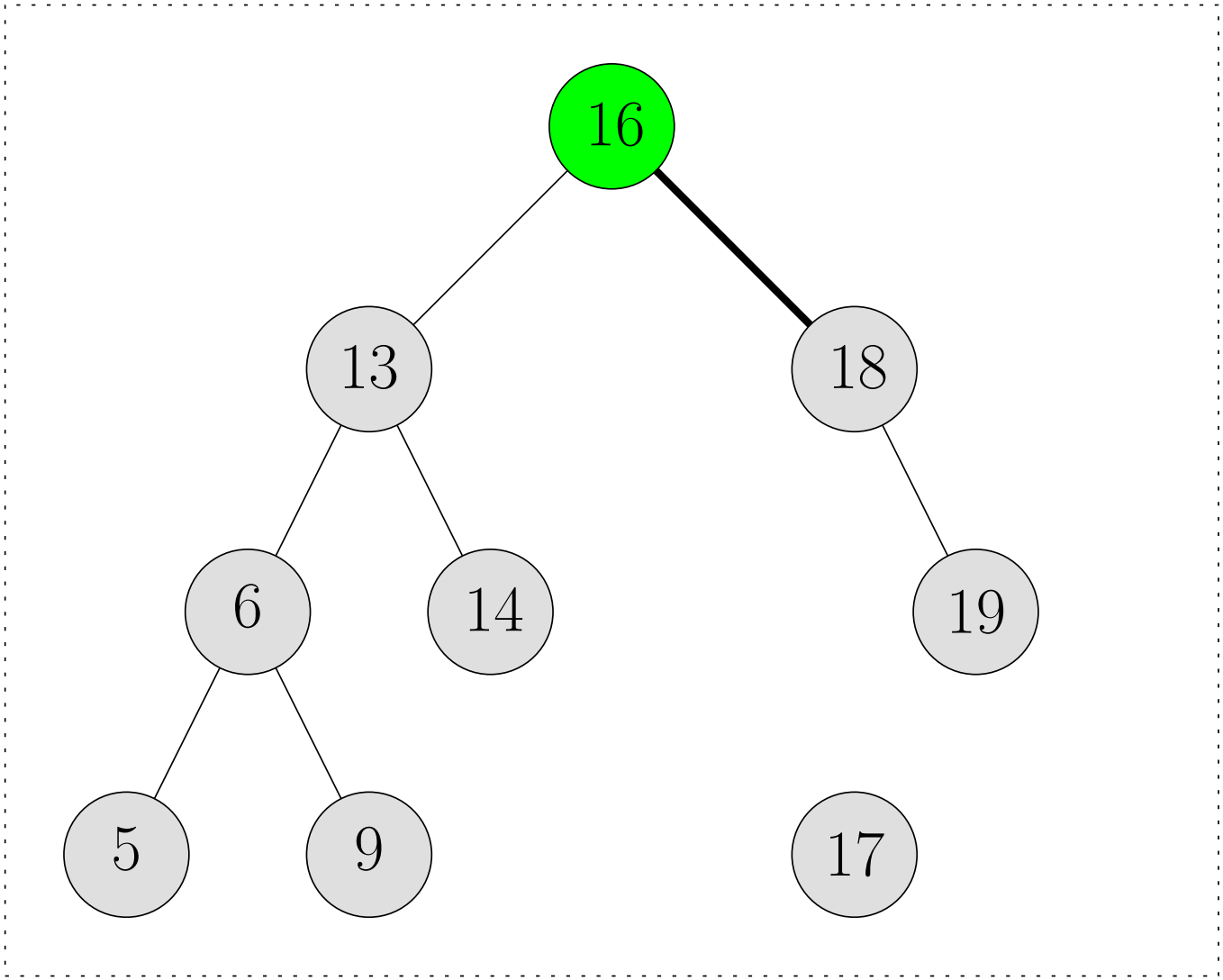
Διαγραφή σε ΔΔΑ

Περίπτωση 3: Διαγραφή του 15 (δύο παιδιά)



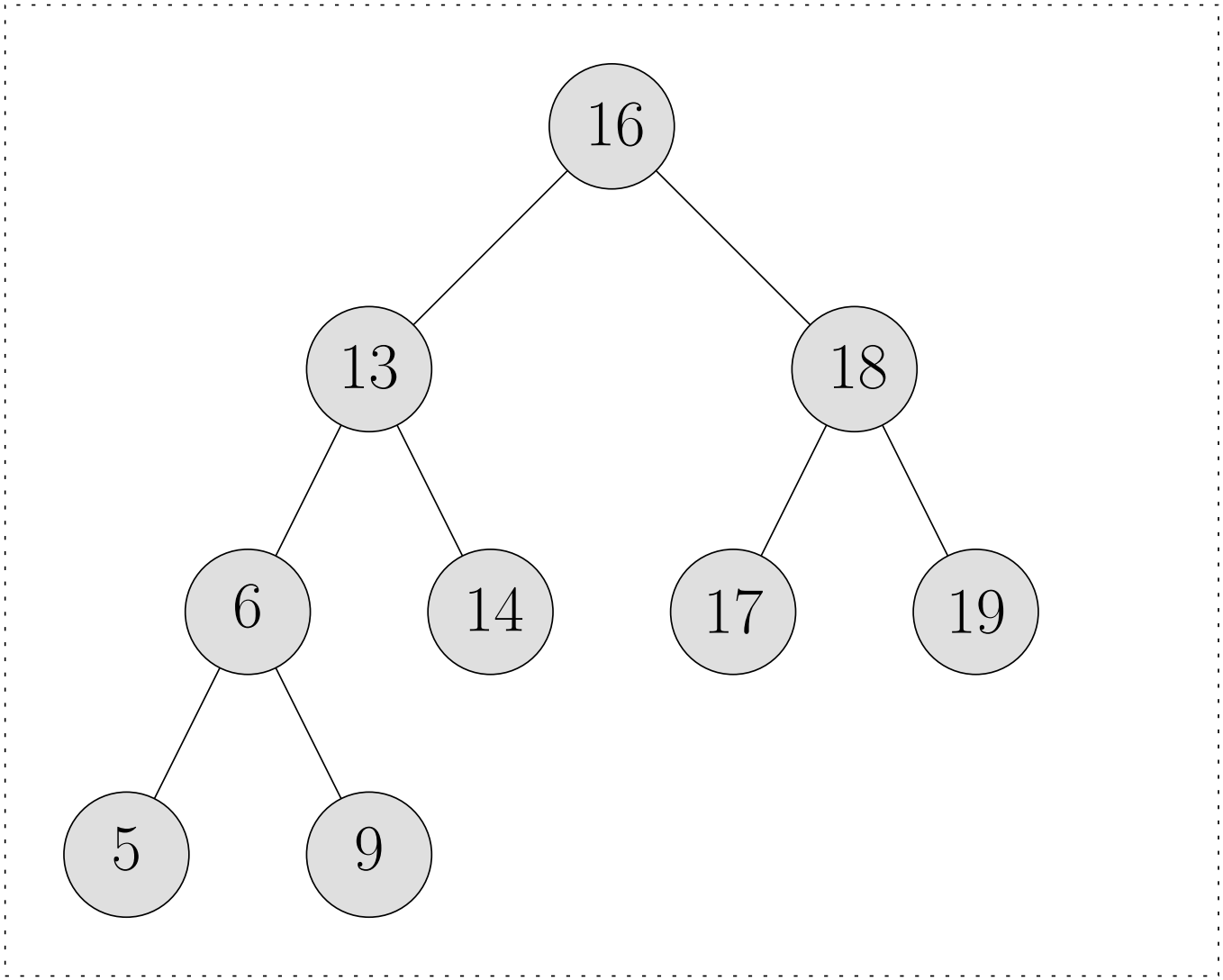
Διαγραφή σε ΔΔΑ

Περίπτωση 3: Διαγραφή του 15 (δύο παιδιά)



Διαγραφή σε ΔΔΑ

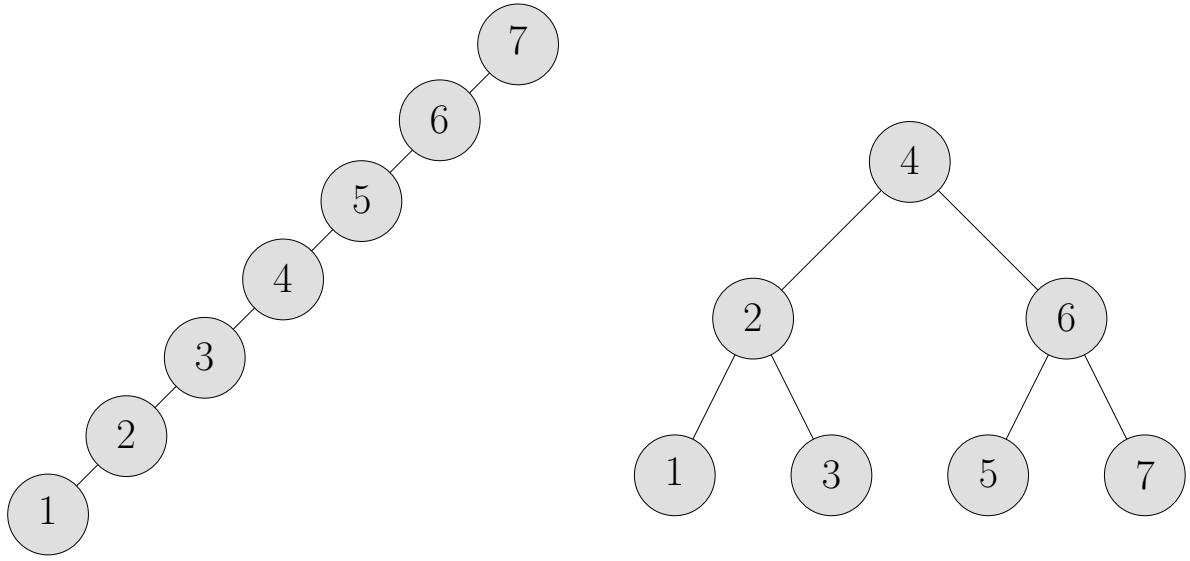
Περίπτωση 3: Διαγραφή του 15 (δύο παιδιά)



Επιδόσεις

Όλες οι λειτουργίες χρειάζονται χρόνο ανάλογο με το ύψος του δέντρου.

Προβλήματα



Υπάρχουν πολλά δέντρα για τα ίδια αντικείμενα

- αριστερά η χειρότερη περίπτωση: ύψος $h = \mathcal{O}(n)$
- δεξιά η καλύτερη περίπτωση: ύψος $h = \mathcal{O}(\log n)$

Το δέντρο εξαρτάται από την σειρά της εισαγωγής και διαγραφής των στοιχείων του δέντρο, η οποία δεν είναι γνωστή εκ των προτέρων

Ισορροπημένα Δέντρα Αναζήτησης

Balanced BSTs

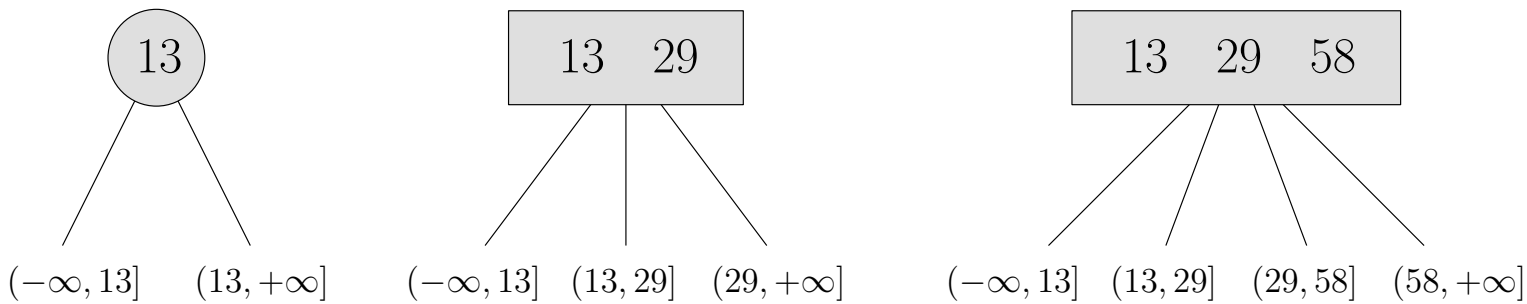
Δέντρα αναζήτησης τα οποία είναι ισορροπημένα ανά πάσα στιγμή ανεξάρτητα από την σειρά των εισαγωγών και διαγραφών στοιχείων.

$$h = \mathcal{O}(\log n)$$

2-3-4 Δέντρα Αναζήτησης

top-down

Επιτρέπουμε κόμβους με 2, 3 ή 4 παιδιά



2-3-4 Δέντρα Αναζήτησης

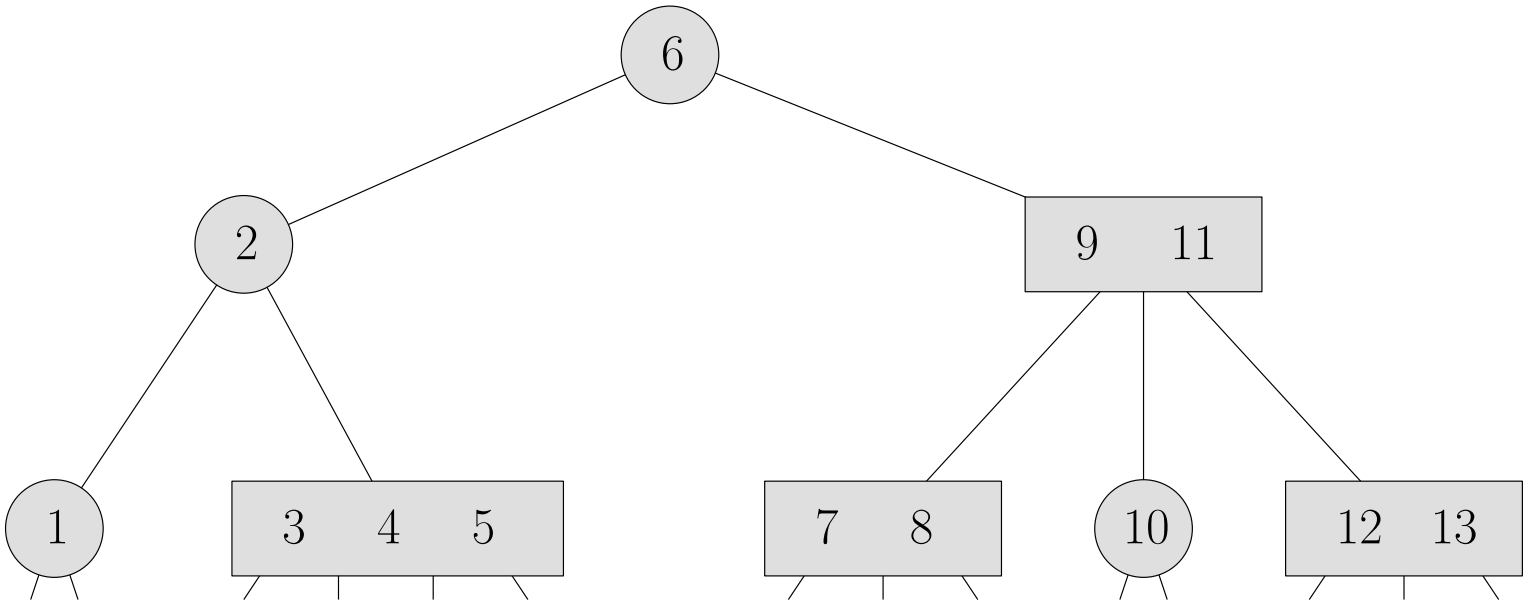
top-down

Ορισμός

Ένα **ισορροπημένο δέντρο αναζήτησης 2-3-4** είναι ένα δέντρο το οποίο:

- είτε είναι κενό
- είτε αποτελείται από τρεις τύπους κόμβων: **2-κόμβους**, **3-κόμβους** και **4-κόμβους**
- όλοι οι εξωτερικοί κόμβοι (nulls - κενοί σύνδεσμοι) ισαπέχουν από την ρίζα

Παράδειγμα Ισορροπημένου 2-3-4 Δέντρου



Ύψος Ισορροπημένου 2-3-4 Δέντρου

Θεώρημα

Ένα 2-3-4 δέντρο αναζήτησης με n κλειδιά έχει ύψος $\mathcal{O}(\log n)$.

Απόδειξη

Έστω h το ύψος του δέντρου.

Ύψος Ισορροπημένου 2-3-4 Δέντρου

Θεώρημα

Ένα 2-3-4 δέντρο αναζήτησης με n κλειδιά έχει ύψος $\mathcal{O}(\log n)$.

Απόδειξη

Έστω h το ύψος του δέντρου.

Κάθε επίπεδο i έχει τουλάχιστον 2^i κόμβους.

Ύψος Ισορροπημένου 2-3-4 Δέντρου

Θεώρημα

Ένα 2-3-4 δέντρο αναζήτησης με n κλειδιά έχει ύψος $\mathcal{O}(\log n)$.

Απόδειξη

Έστω h το ύψος του δέντρου.

Κάθε επίπεδο i έχει τουλάχιστον 2^i κόμβους.

Άρα $n \geq 1 + 2 + 4 + \ldots 2^{h-1} = 2^h - 1$.

Ύψος Ισορροπημένου 2-3-4 Δέντρου

Θεώρημα

Ένα 2-3-4 δέντρο αναζήτησης με n κλειδιά έχει ύψος $\mathcal{O}(\log n)$.

Απόδειξη

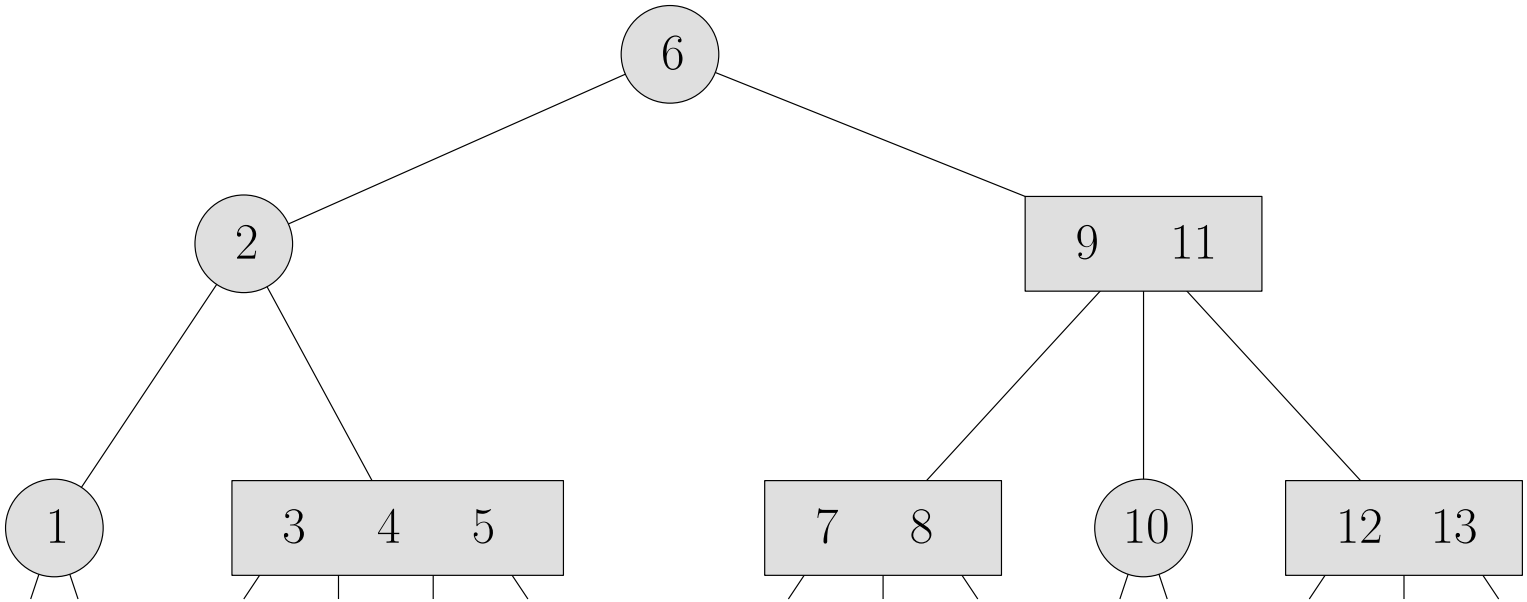
Έστω h το ύψος του δέντρου.

Κάθε επίπεδο i έχει τουλάχιστον 2^i κόμβους.

Άρα $n \geq 1 + 2 + 4 + \ldots 2^{h-1} = 2^h - 1$.

Λογαριθμίζοντας πέρνουμε πως $\log_2(n+1) \geq h$. □

Αναζήτηση σε Ισορροπημένο 2-3-4 Δέντρο



Αναζήτηση σε 2-3-4 Δέντρο: Γενίκευση του αλγόριθμου για ΔΔΑ

Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

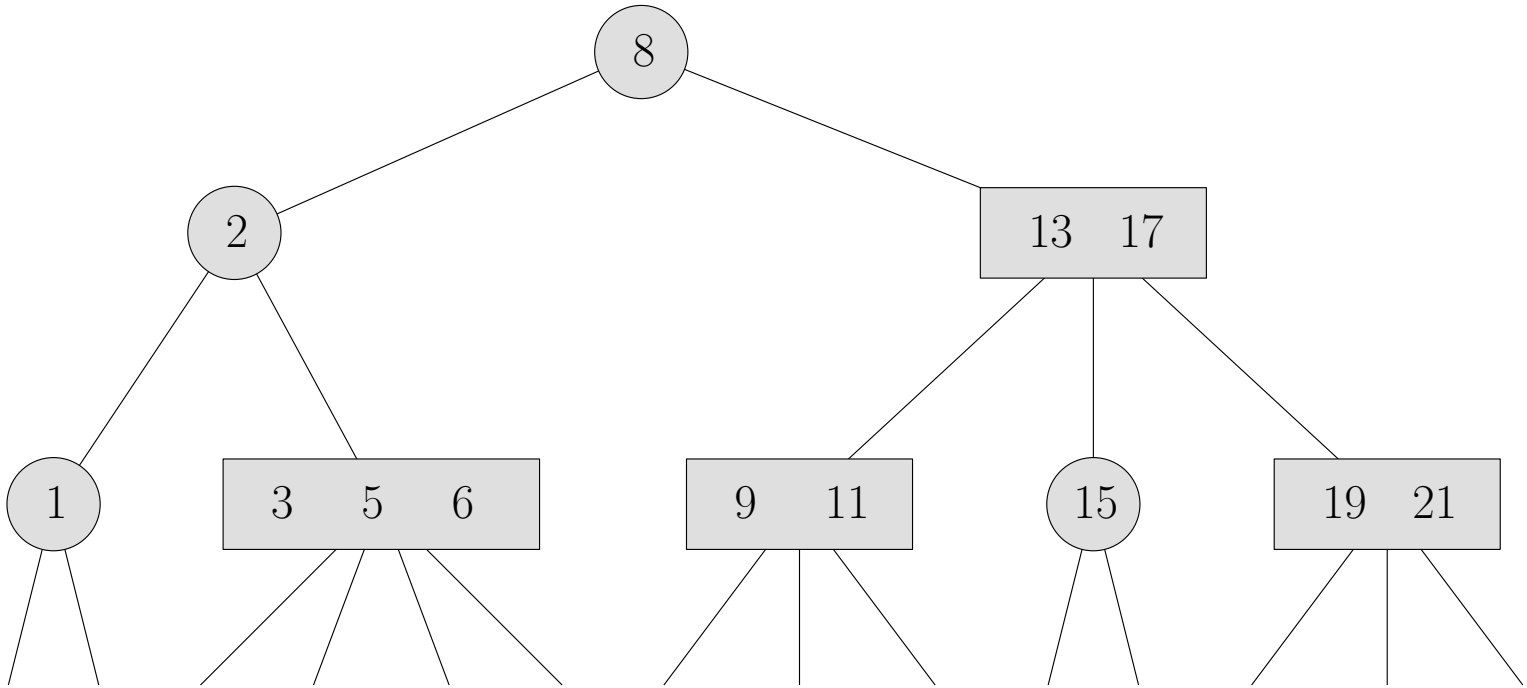
Διατηρώντας την ισορροπία!

Τρέχουμε αναζήτηση για να δούμε που πρέπει να προστεθεί το νέο κλειδί:

- εάν η αναζήτηση τερματιστεί σε 2-κόμβο τον κάνουμε 3-κόμβο
- εάν η αναζήτηση τερματιστεί σε 3-κόμβο τον κάνουμε 4-κόμβο
- εάν η αναζήτηση τερματιστεί σε 4-κόμβο
  - σπάμε τον 4-κόμβο σε δύο 2-κόμβους μεταβιβάζοντας το μεσαίο κλειδί προς τα επάνω και μετά
  - προσθέτουμε το καινούριο κλειδί σε έναν από τους 2-κόμβους (μπορεί να χρειαστεί να κάνουμε το ίδιο και στον πατέρα, μέχρι την ρίζα)

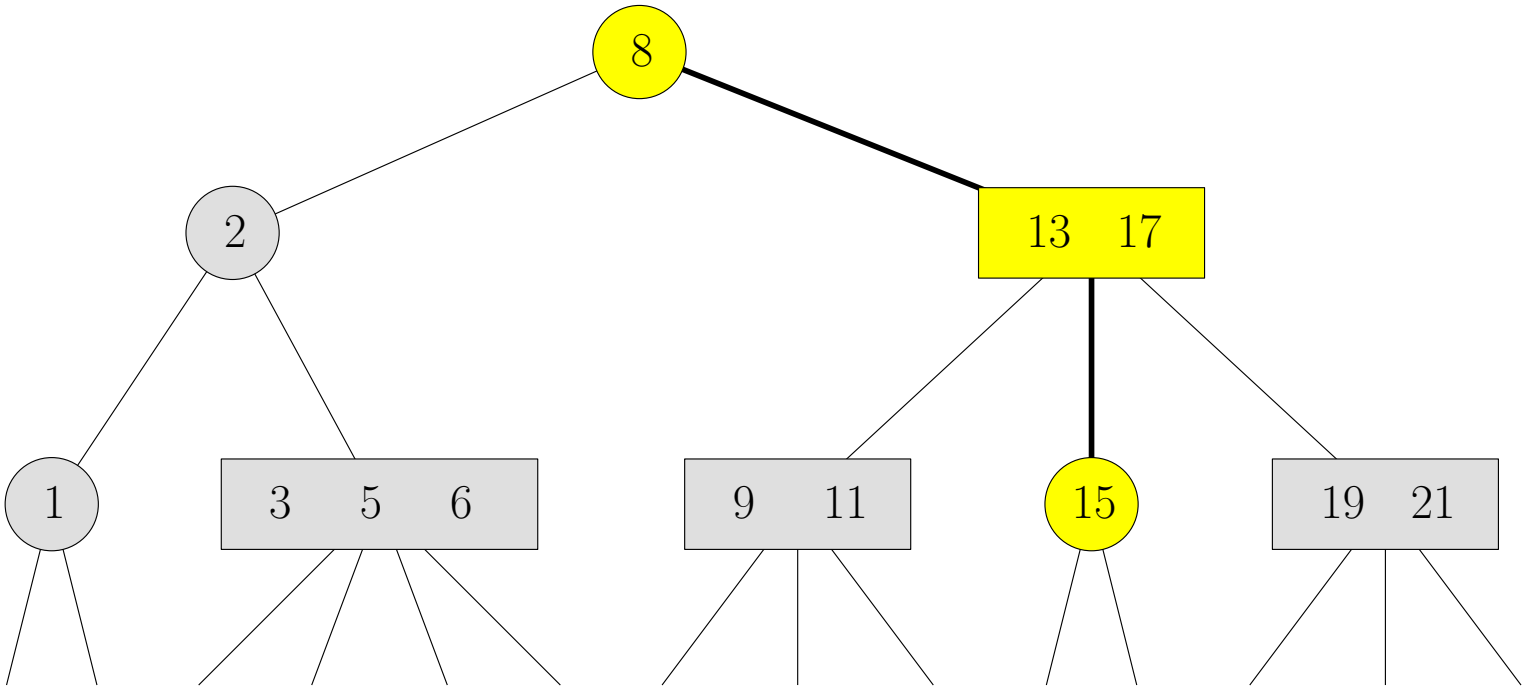
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Τερματισμός σε 2-κόμβο (εισαγωγή του 14)



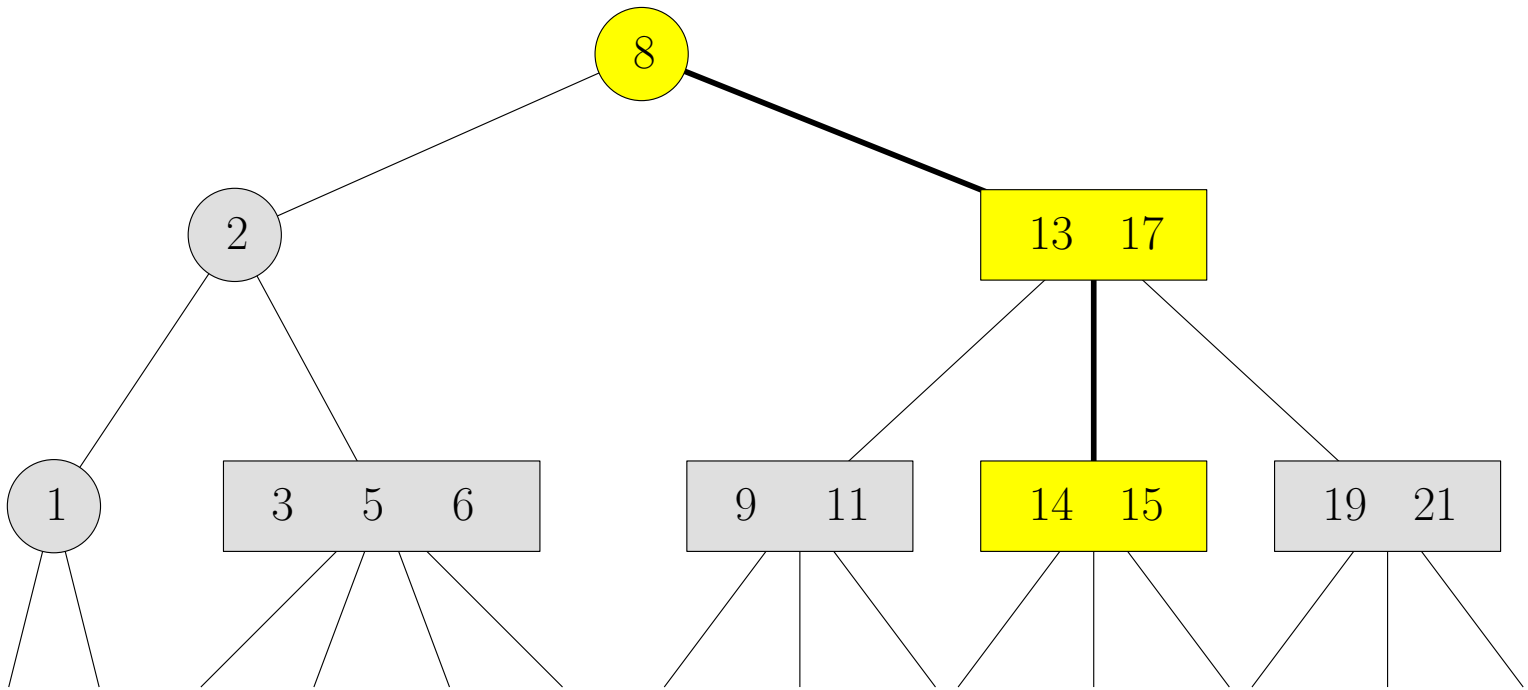
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Τερματισμός σε 2-κόμβο (εισαγωγή του 14)



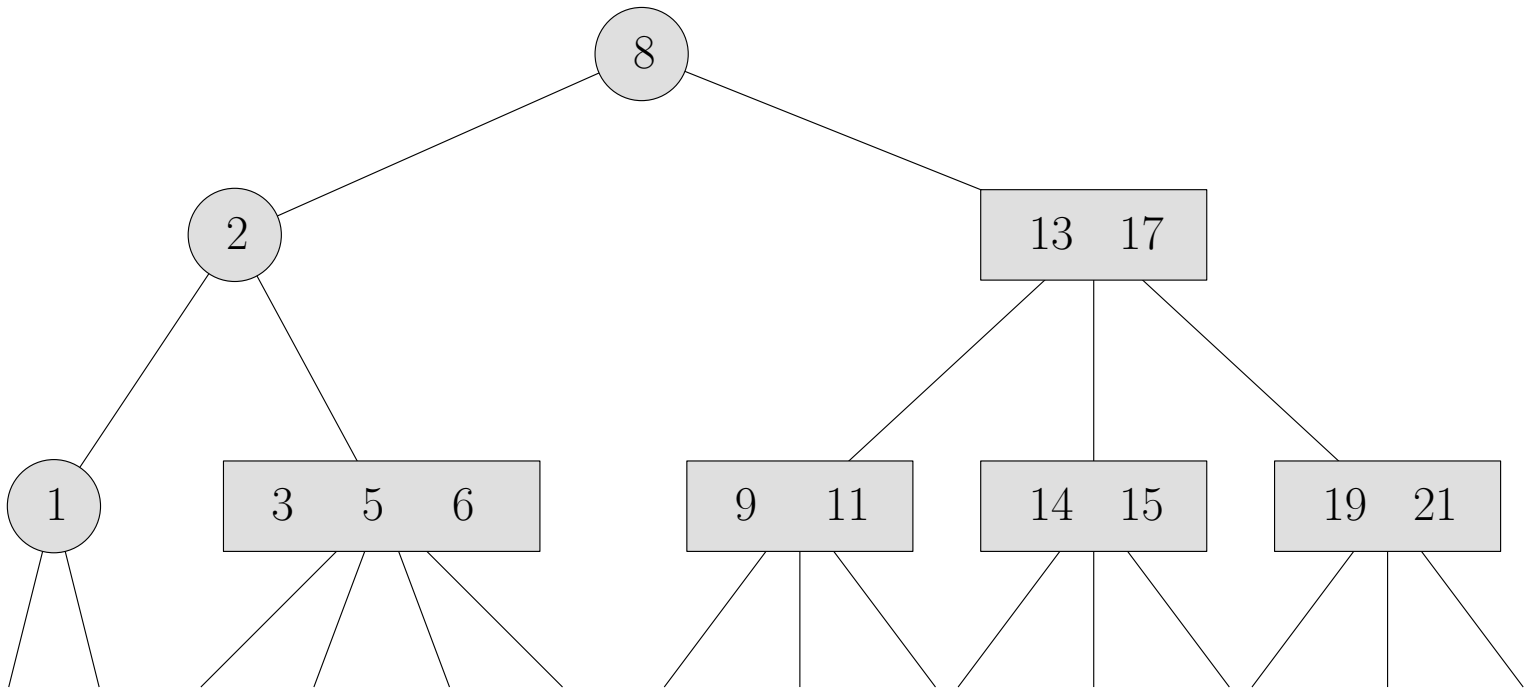
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Τερματισμός σε 2-κόμβο (εισαγωγή του 14)



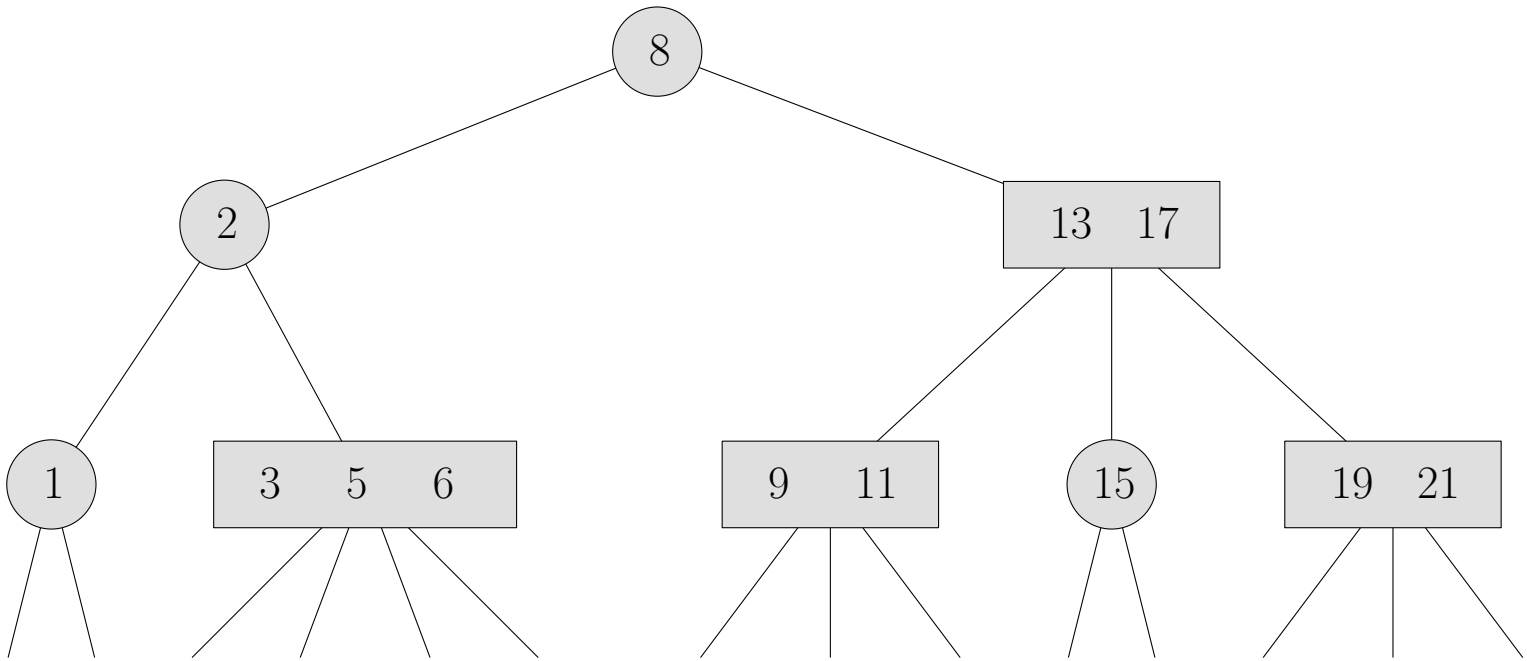
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Τερματισμός σε 2-κόμβο (εισαγωγή του 14)



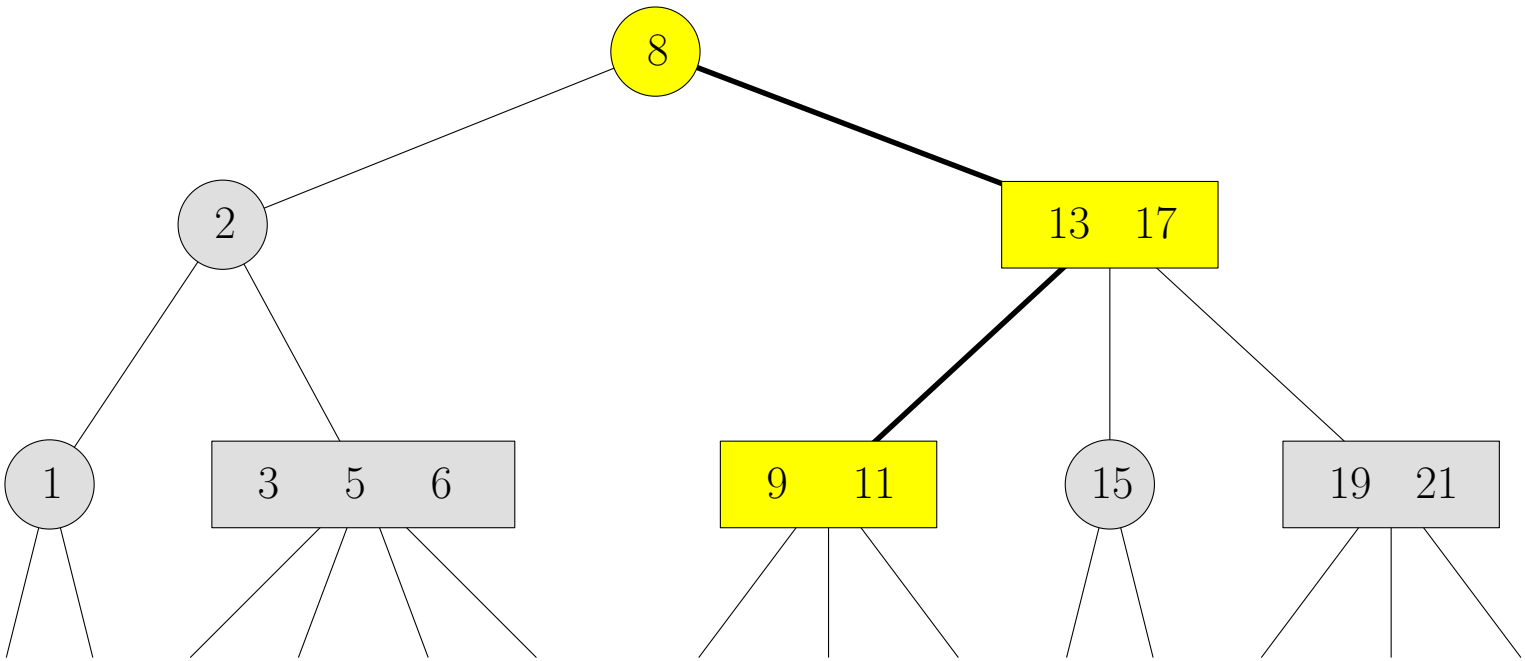
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Τερματισμός σε 3-κόμβο (εισαγωγή του 10)



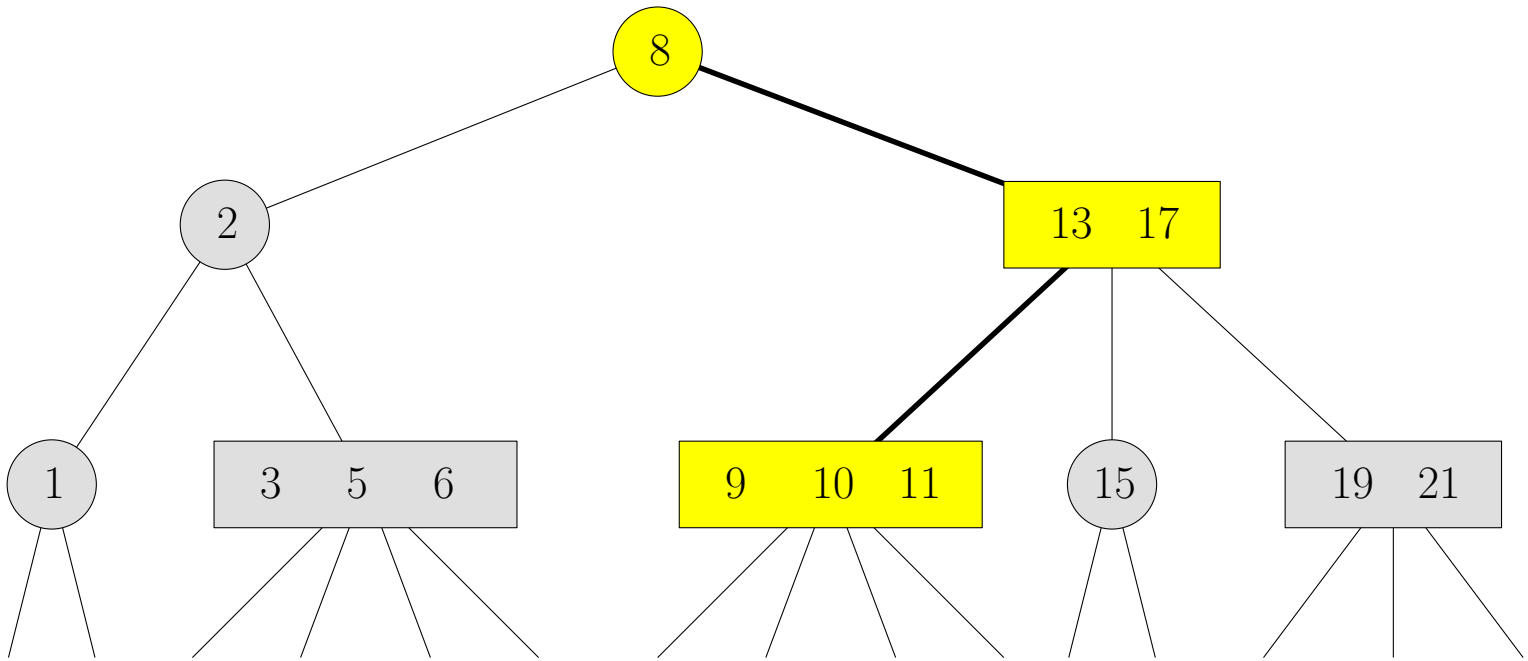
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Τερματισμός σε 3-κόμβο (εισαγωγή του 10)



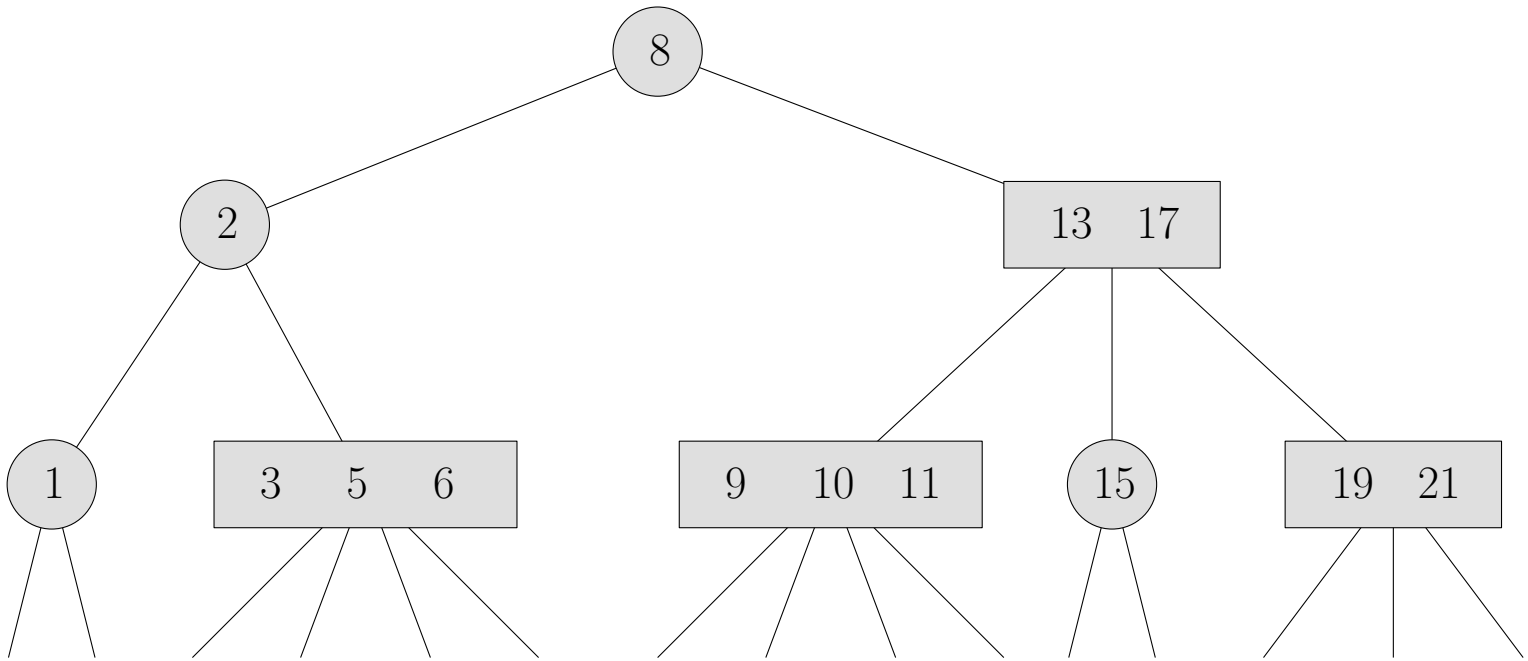
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Τερματισμός σε 3-κόμβο (εισαγωγή του 10)



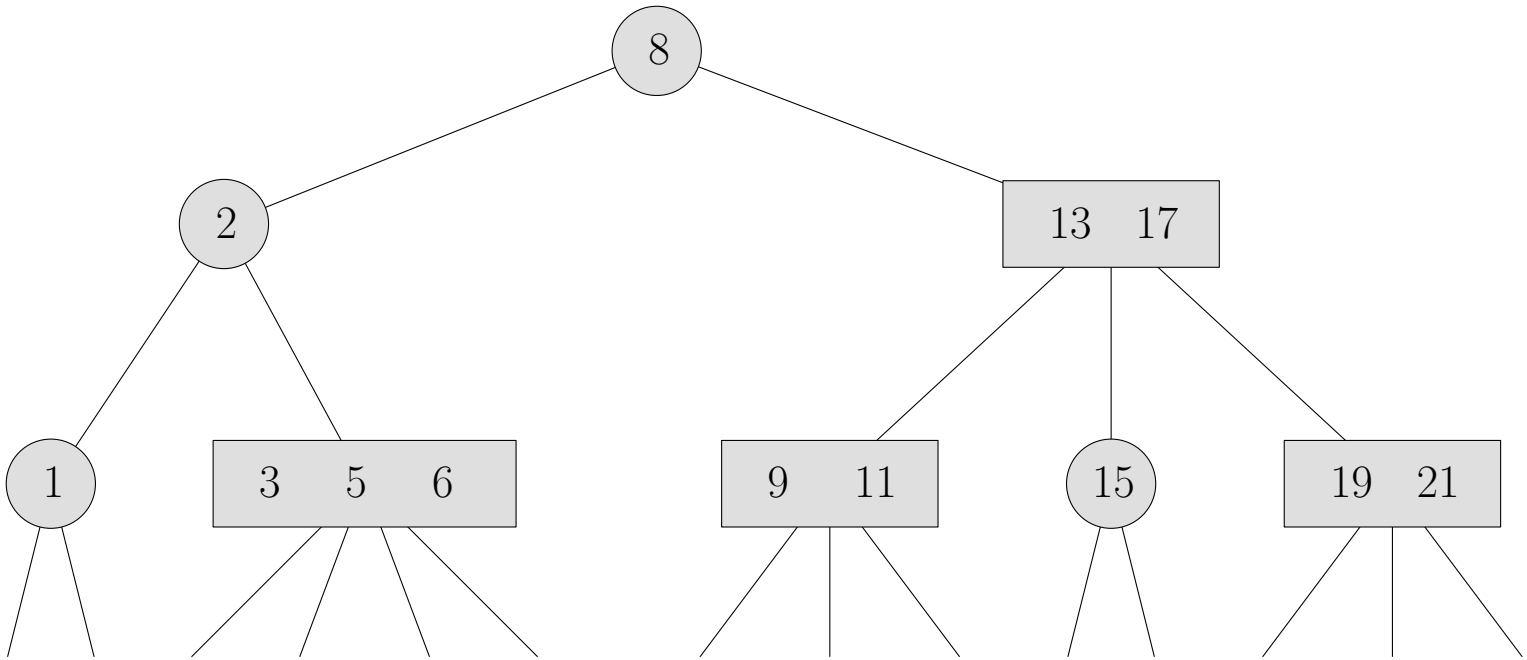
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Τερματισμός σε 3-κόμβο (εισαγωγή του 10)



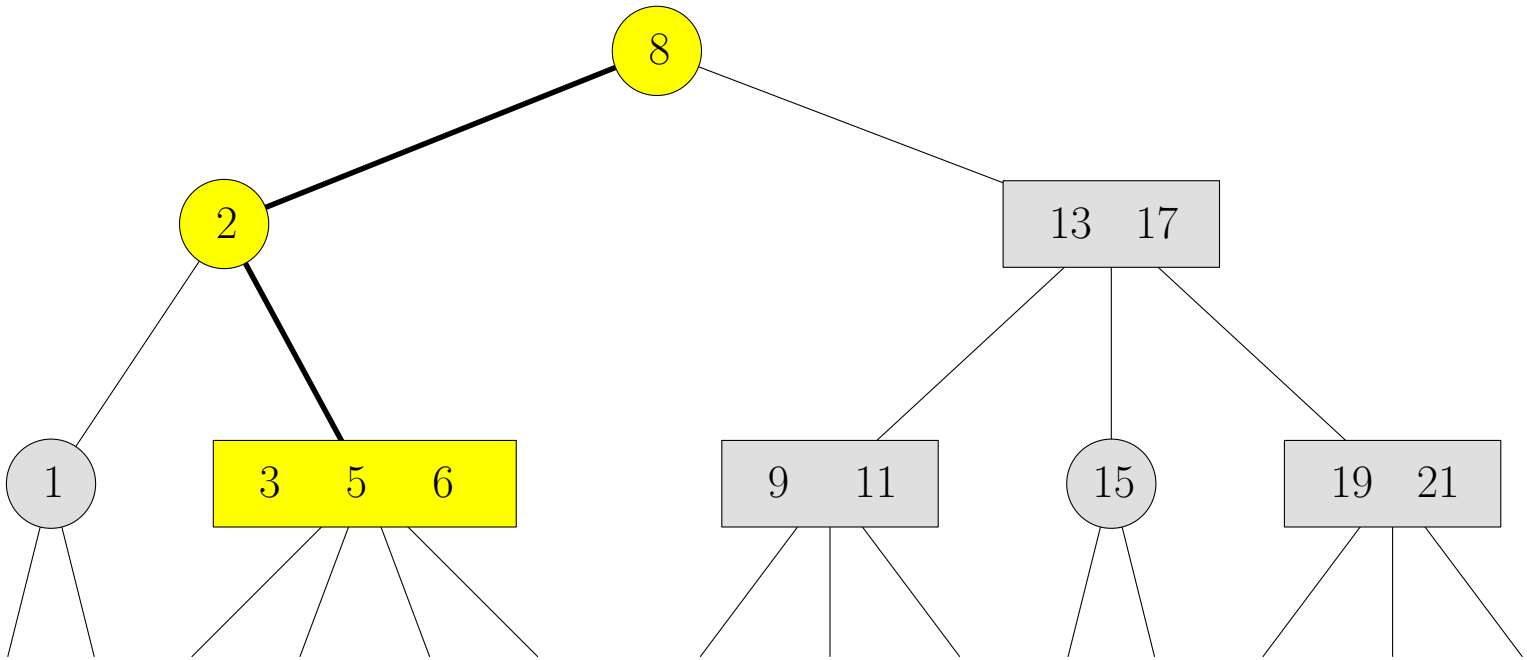
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Τερματισμός σε 4-κόμβο (εισαγωγή του 4)



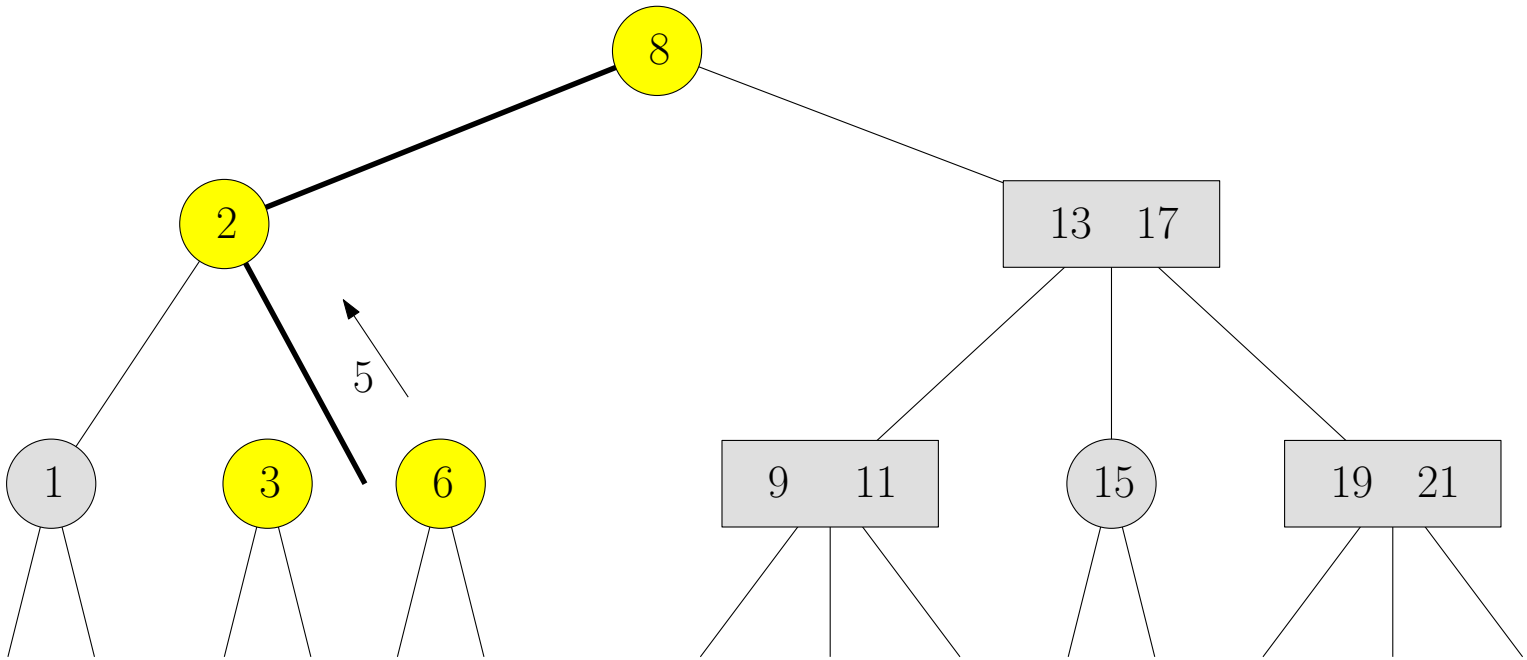
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Τερματισμός σε 4-κόμβο (εισαγωγή του 4)



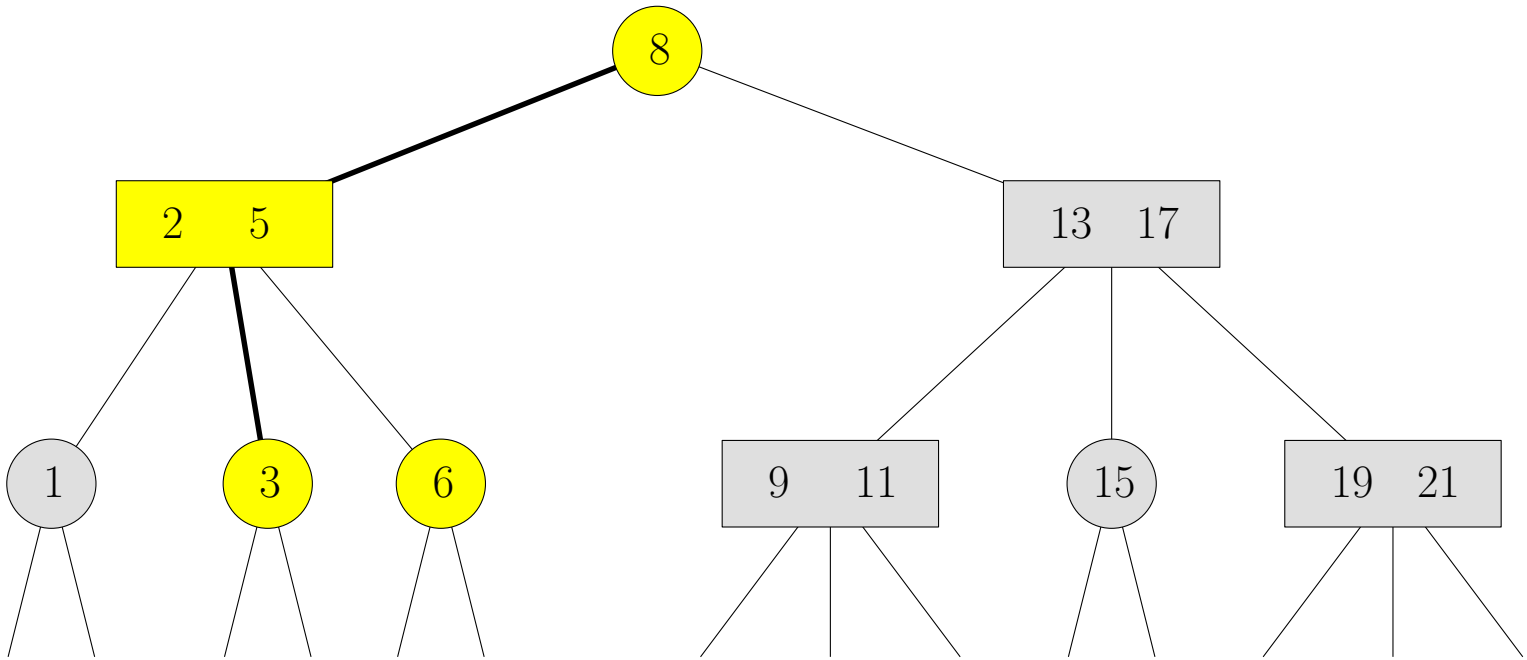
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Τερματισμός σε 4-κόμβο (εισαγωγή του 4)



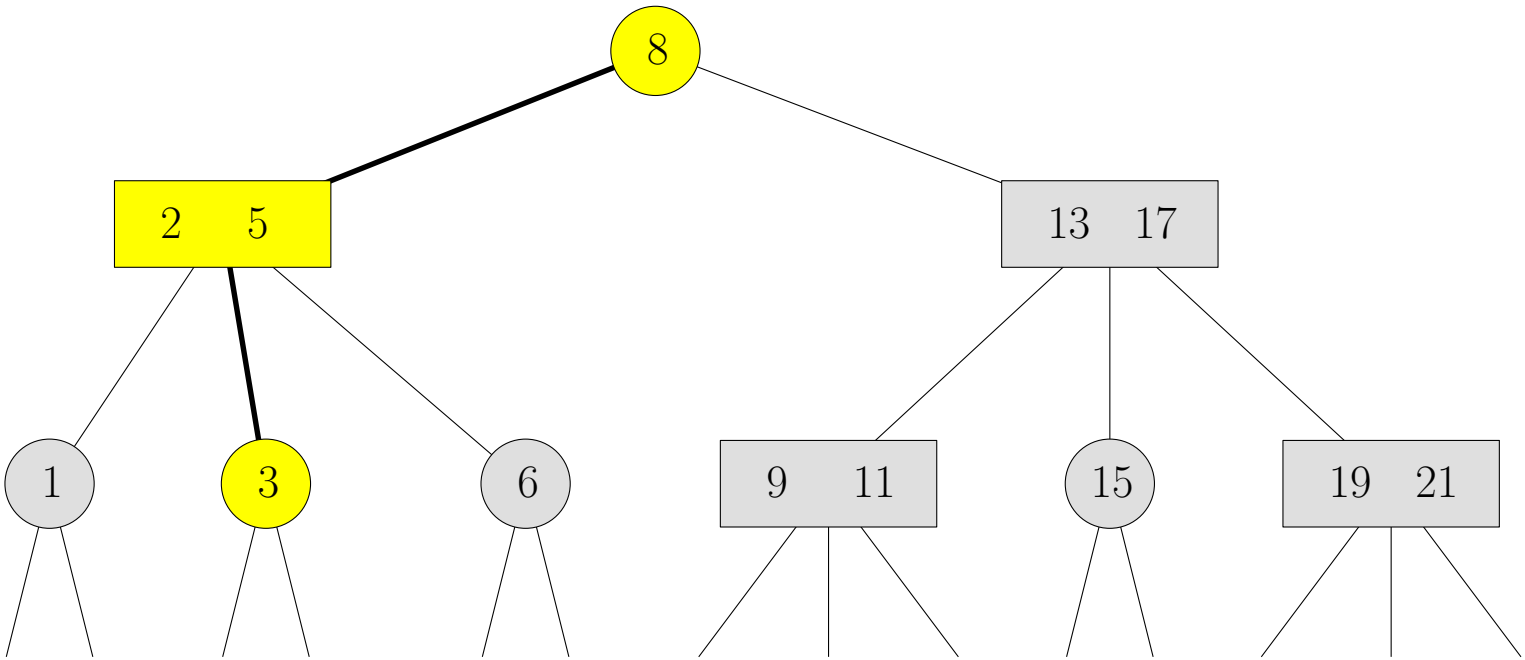
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Τερματισμός σε 4-κόμβο (εισαγωγή του 4)



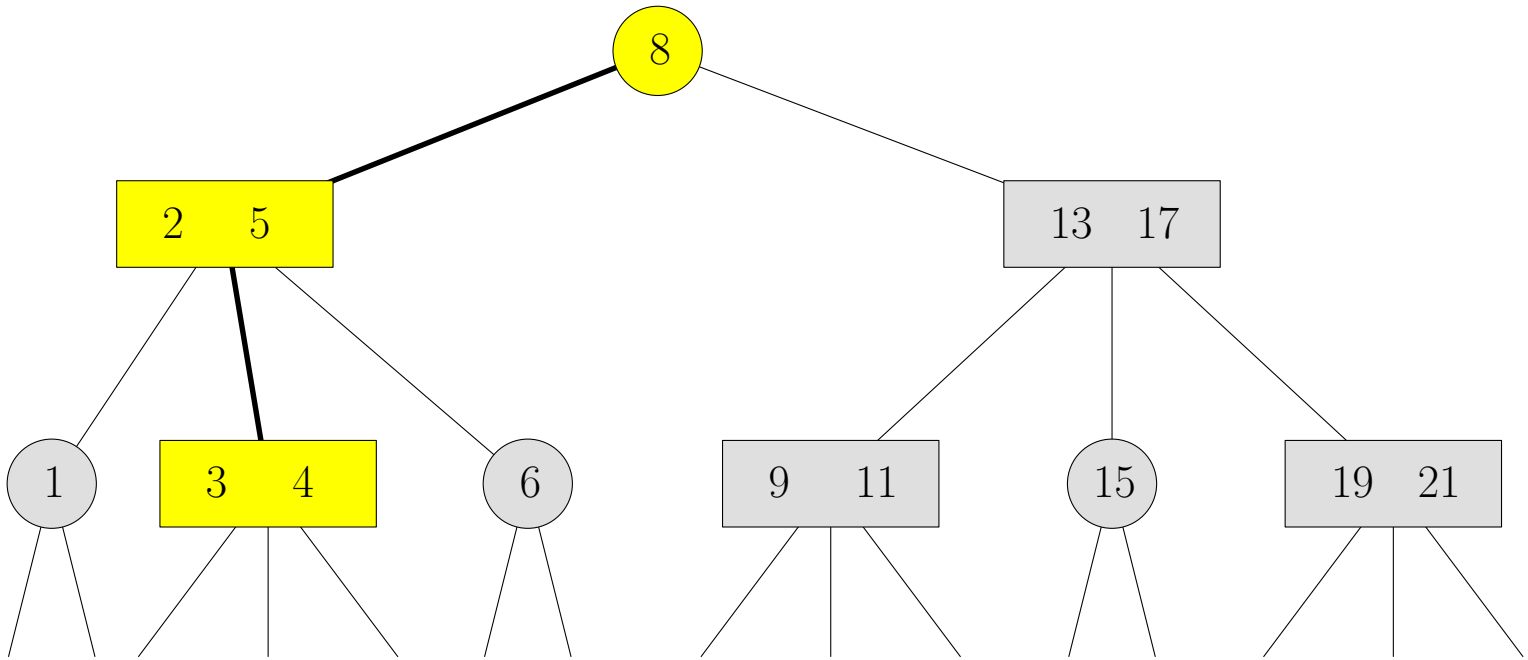
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Τερματισμός σε 4-κόμβο (εισαγωγή του 4)



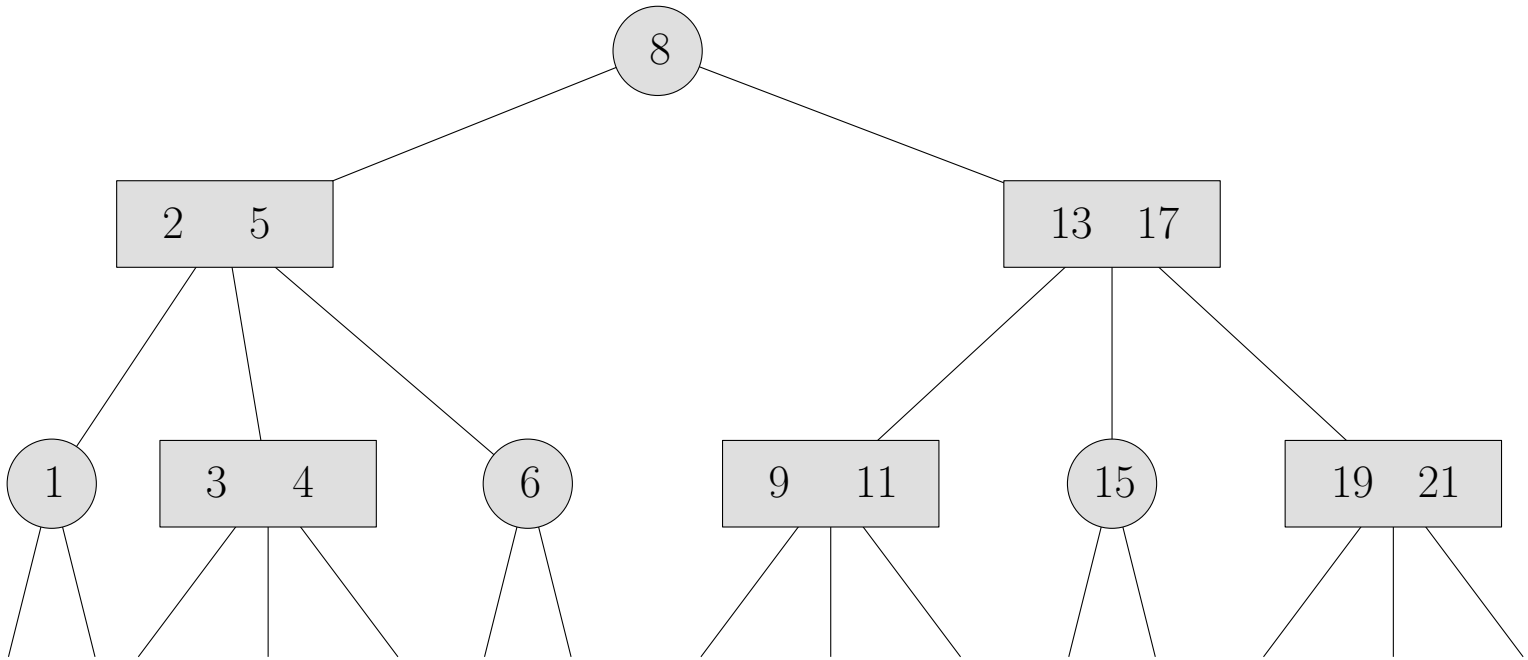
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Τερματισμός σε 4-κόμβο (εισαγωγή του 4)



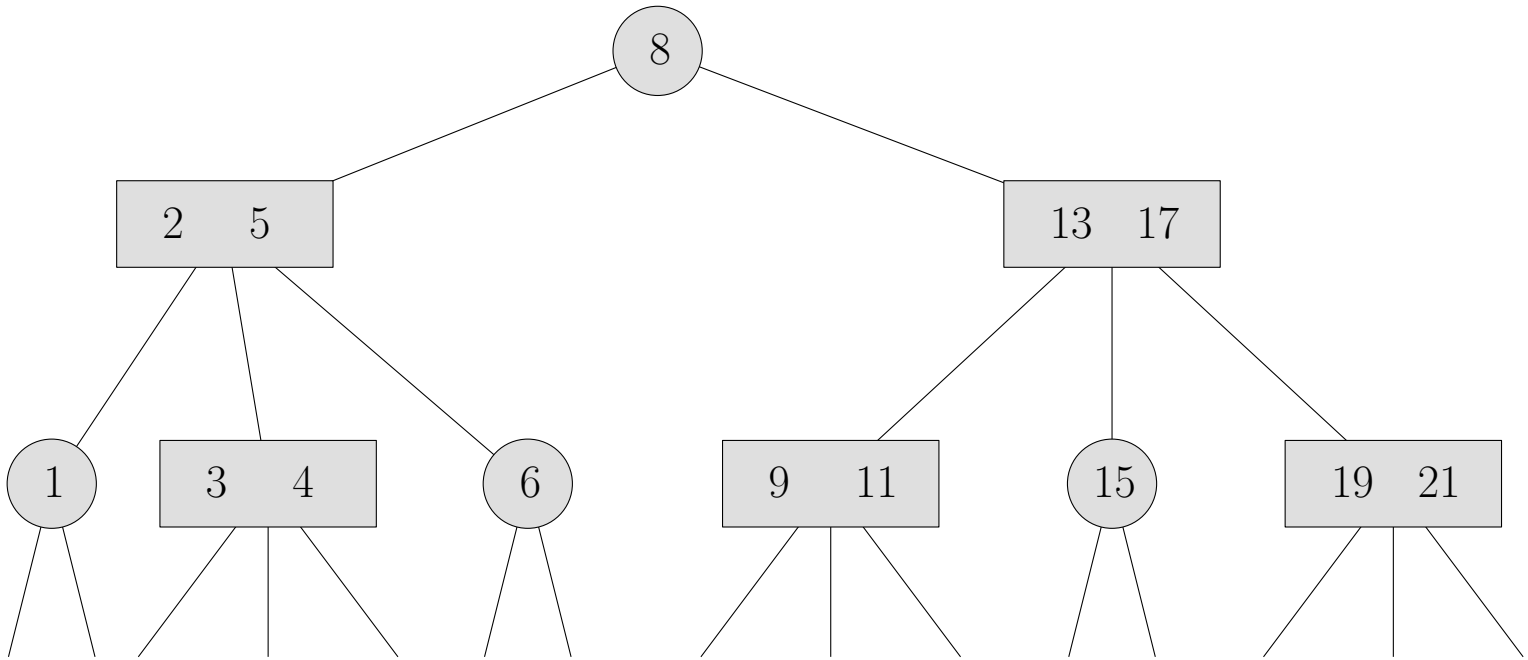
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Τερματισμός σε 4-κόμβο (εισαγωγή του 4)



Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Τερματισμός σε 4-κόμβο (εισαγωγή του 4)



Υπάρχει όμως και η περίπτωση όπου και ο πατέρας είναι 4-κόμβος.

Σε αυτήν την περίπτωση κάνουμε την ίδια διάσπαση ξανά, πιθανώς μέχρι την ρίζα

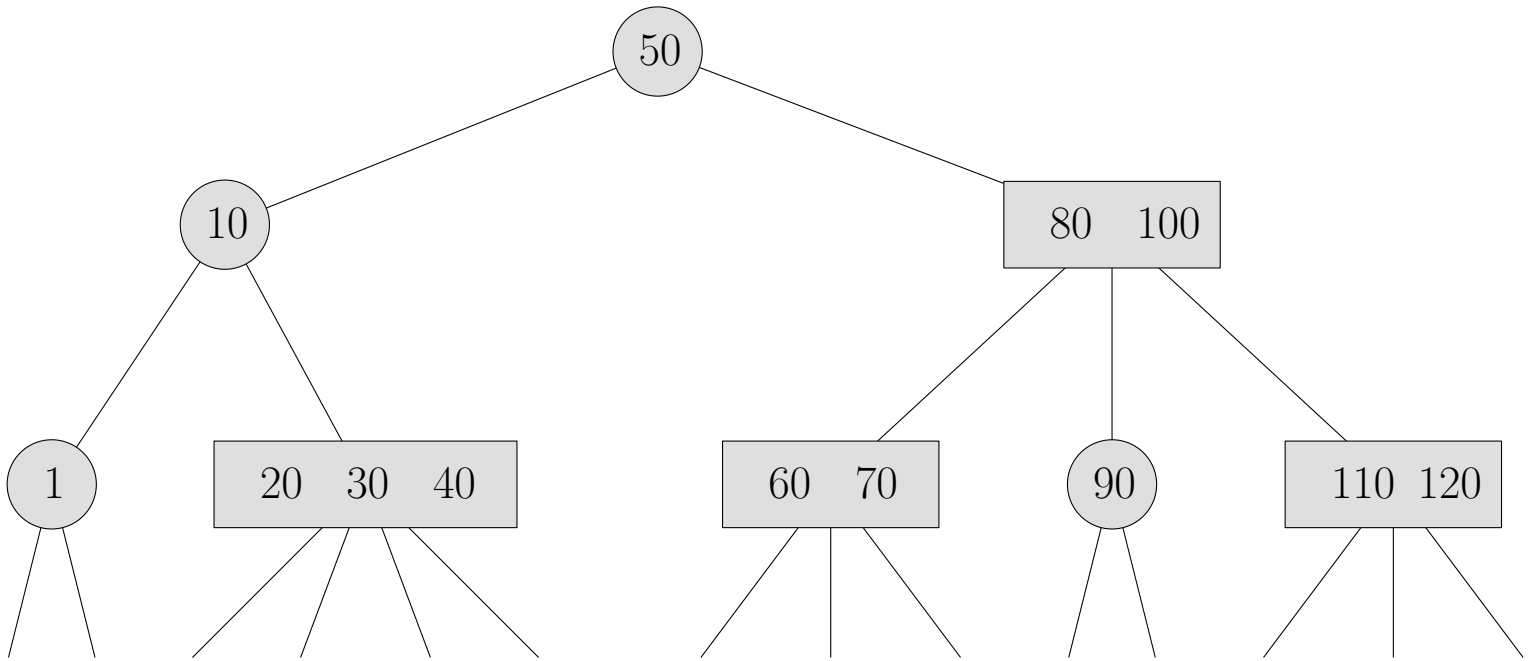
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Για να απλουστεύσουμε την διαδικασία εισαγωγής σε περίπτωση διάσπασης πολλών κόμβων, χρησιμοποιούμε την εξής τεχνική:

- φροντίζουμε η διαδρομή αναζήτησης να μη τερματίσει σε 4-κόμβο, χωρίζοντας κάθε 4-κόμβο που βρίσκουμε κατά την **κάθοδο** στο δέντρο

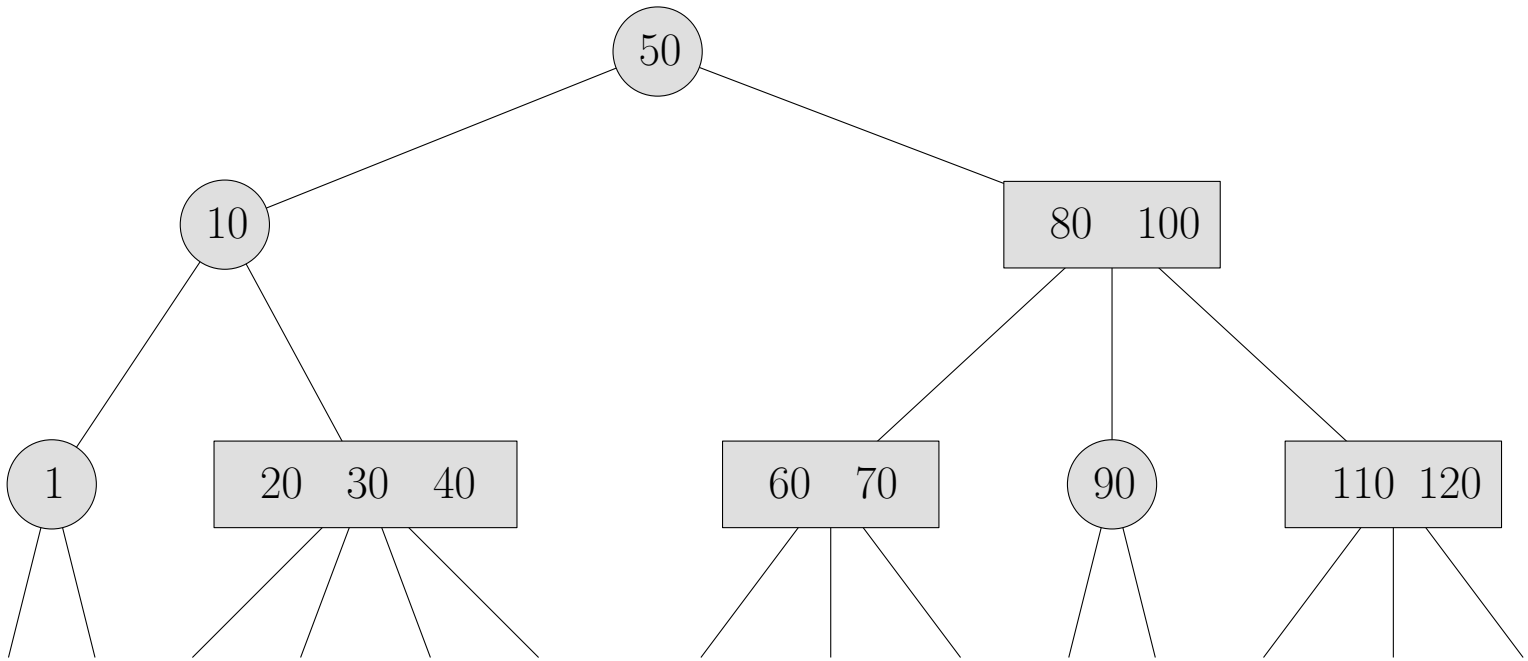
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 25



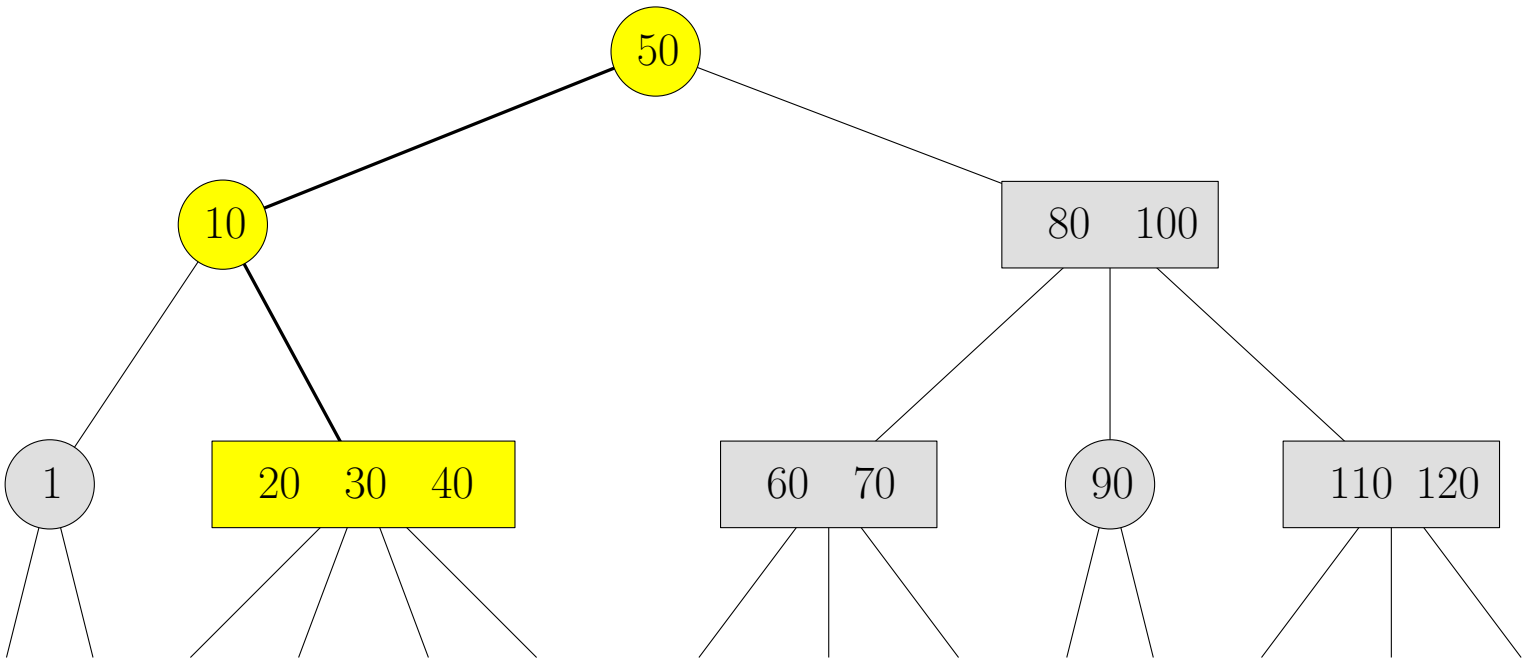
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 25



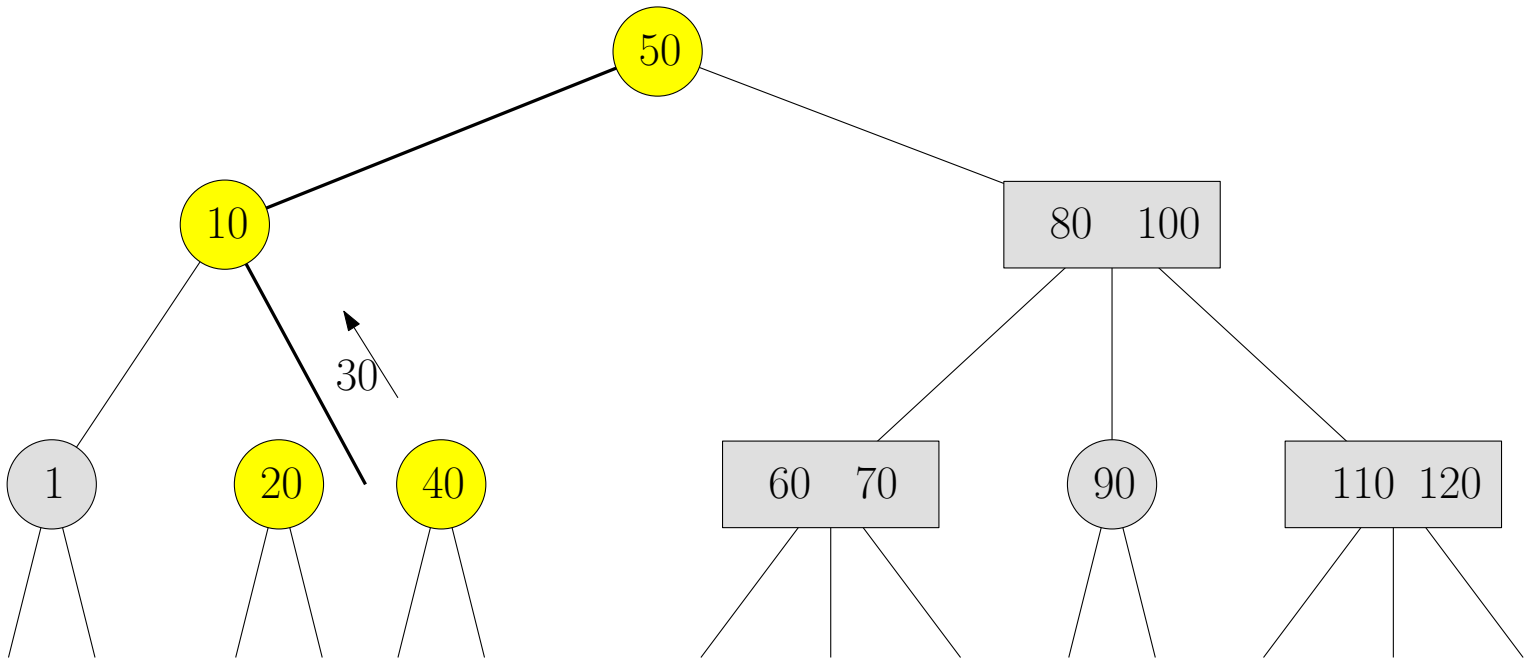
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 25



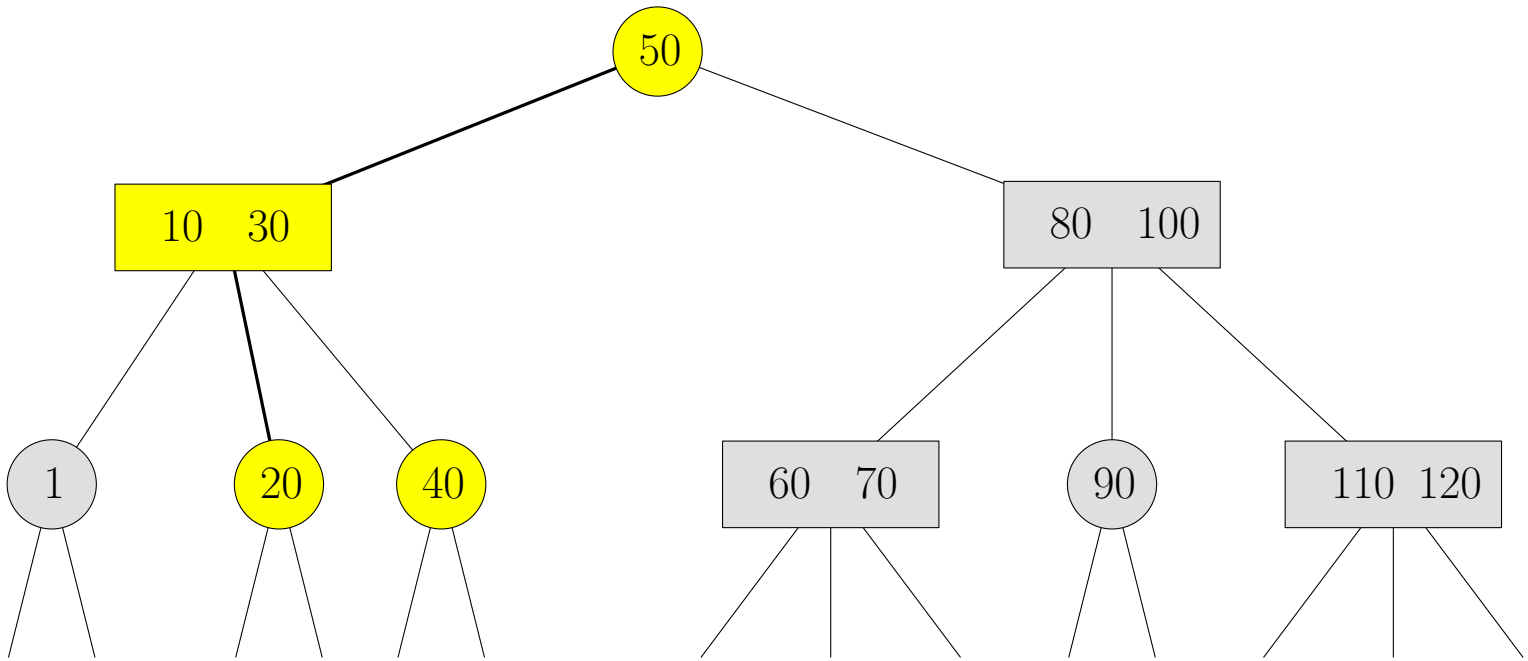
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 25



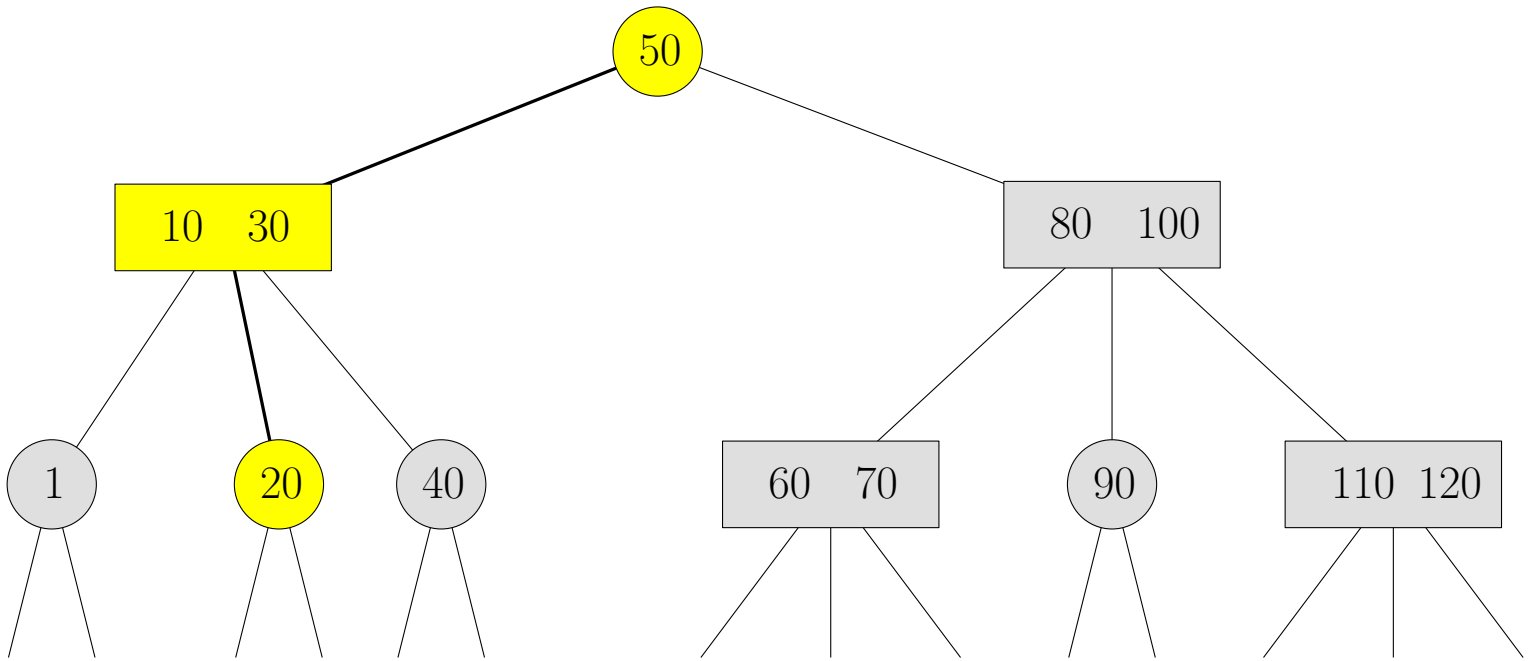
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 25



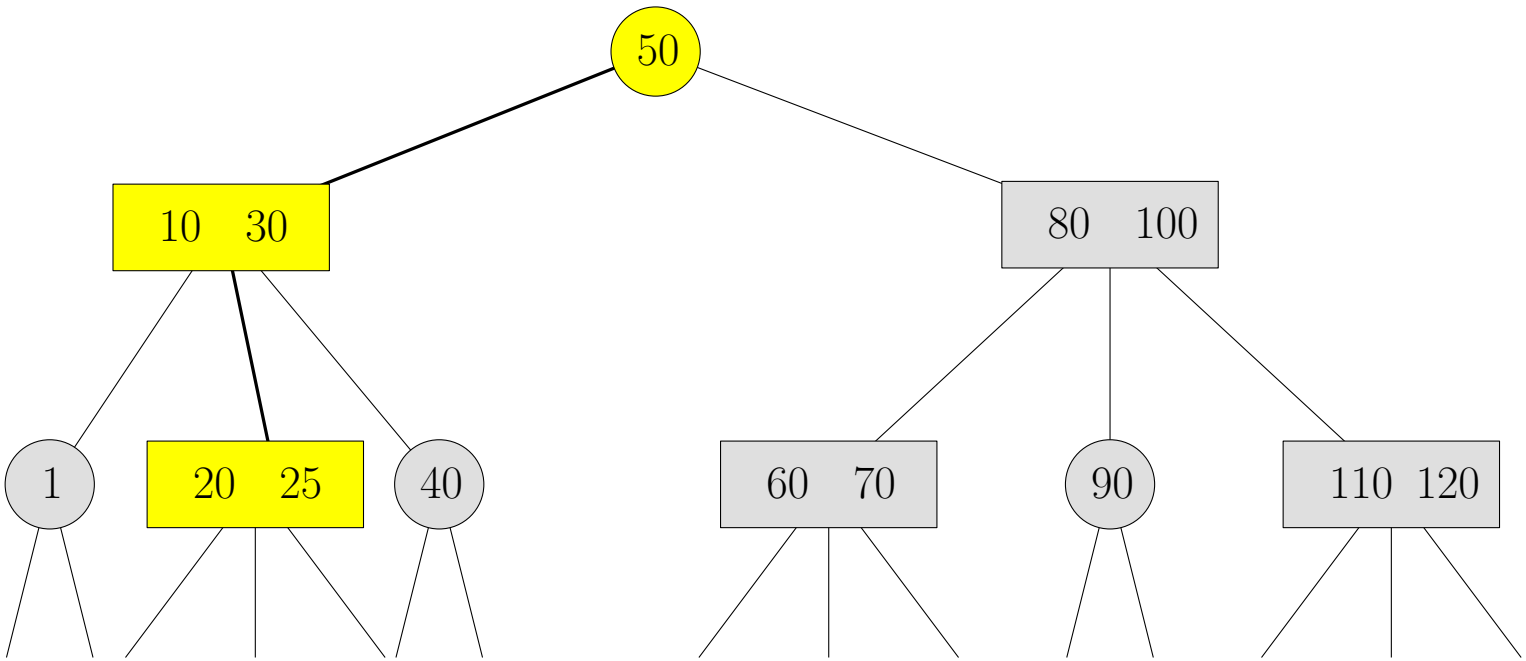
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 25



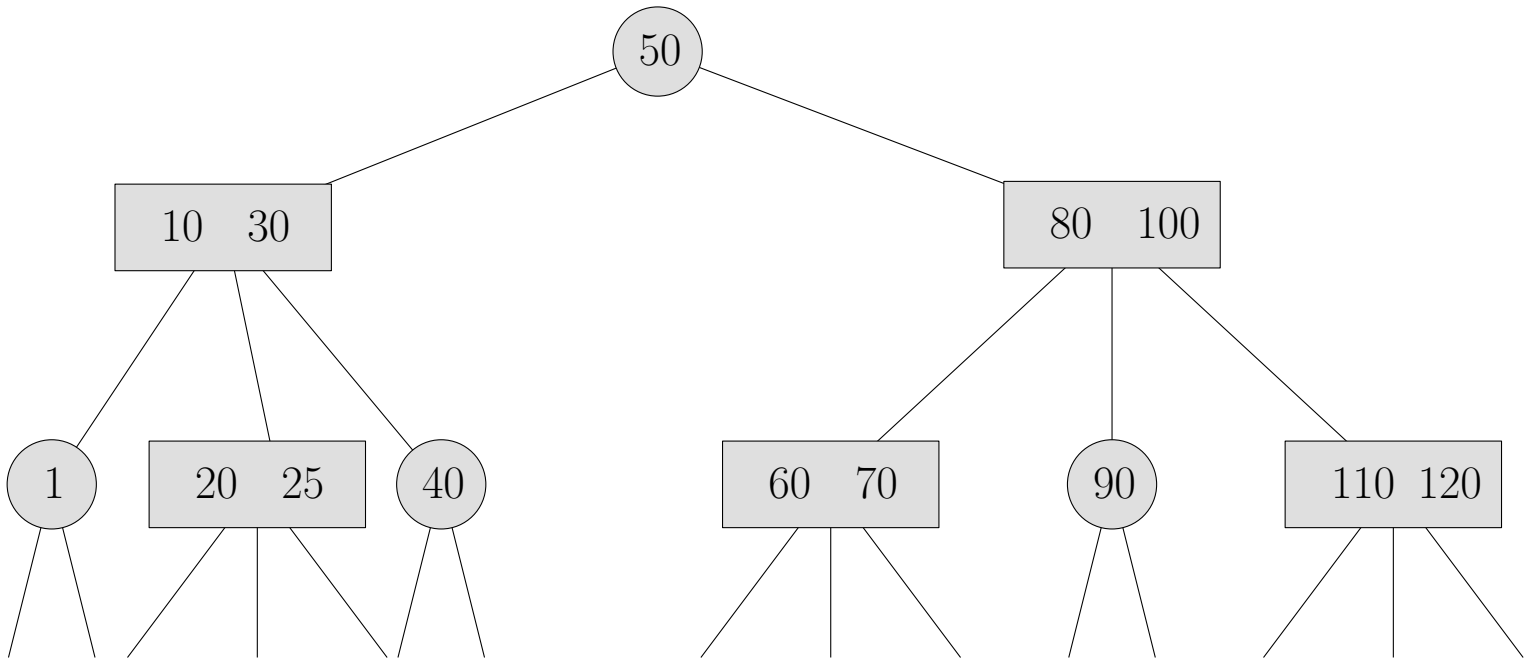
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 25



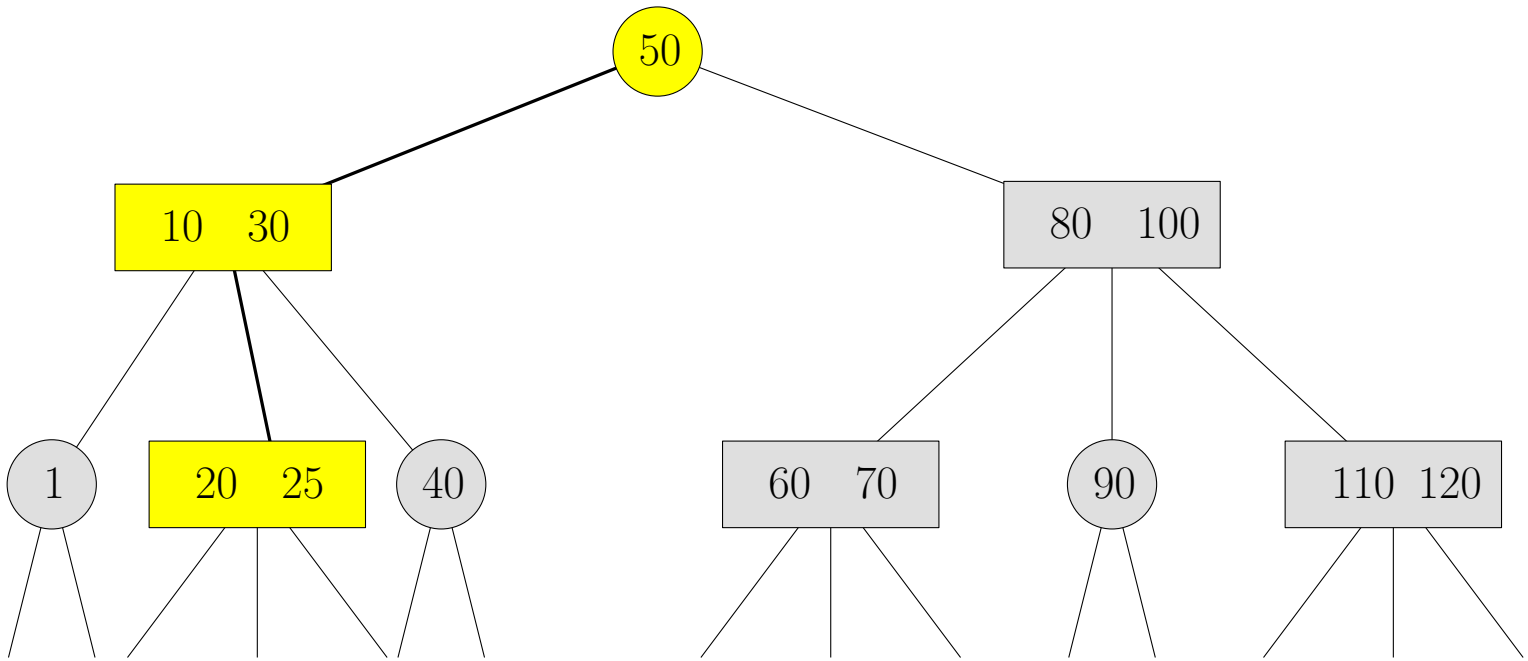
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 29



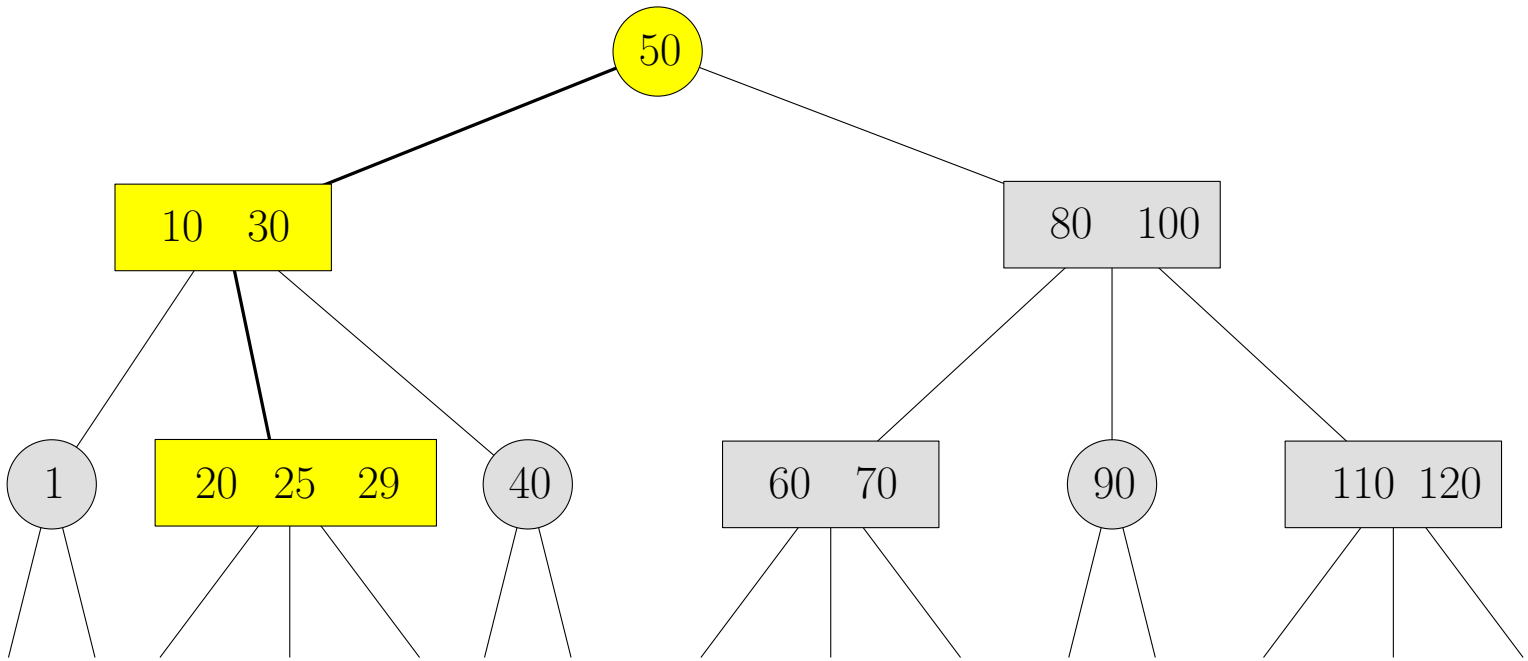
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 29



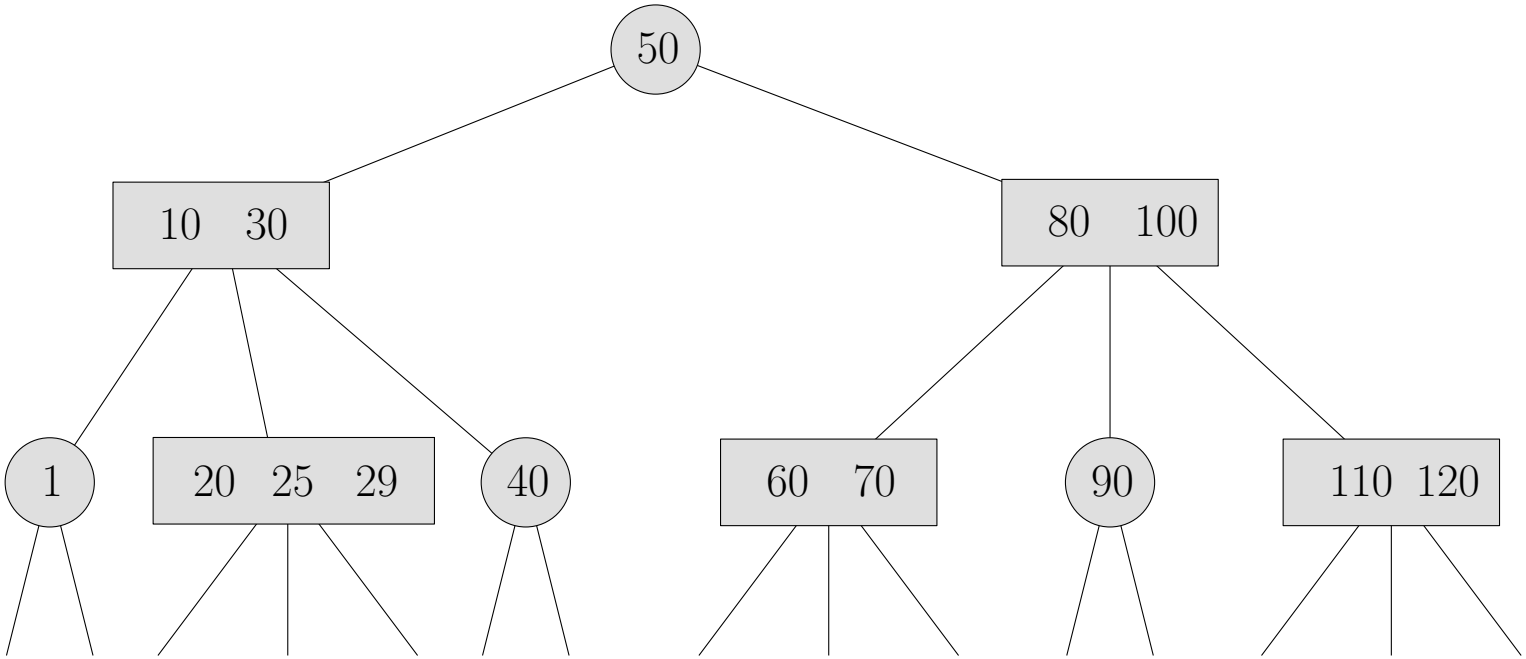
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 29



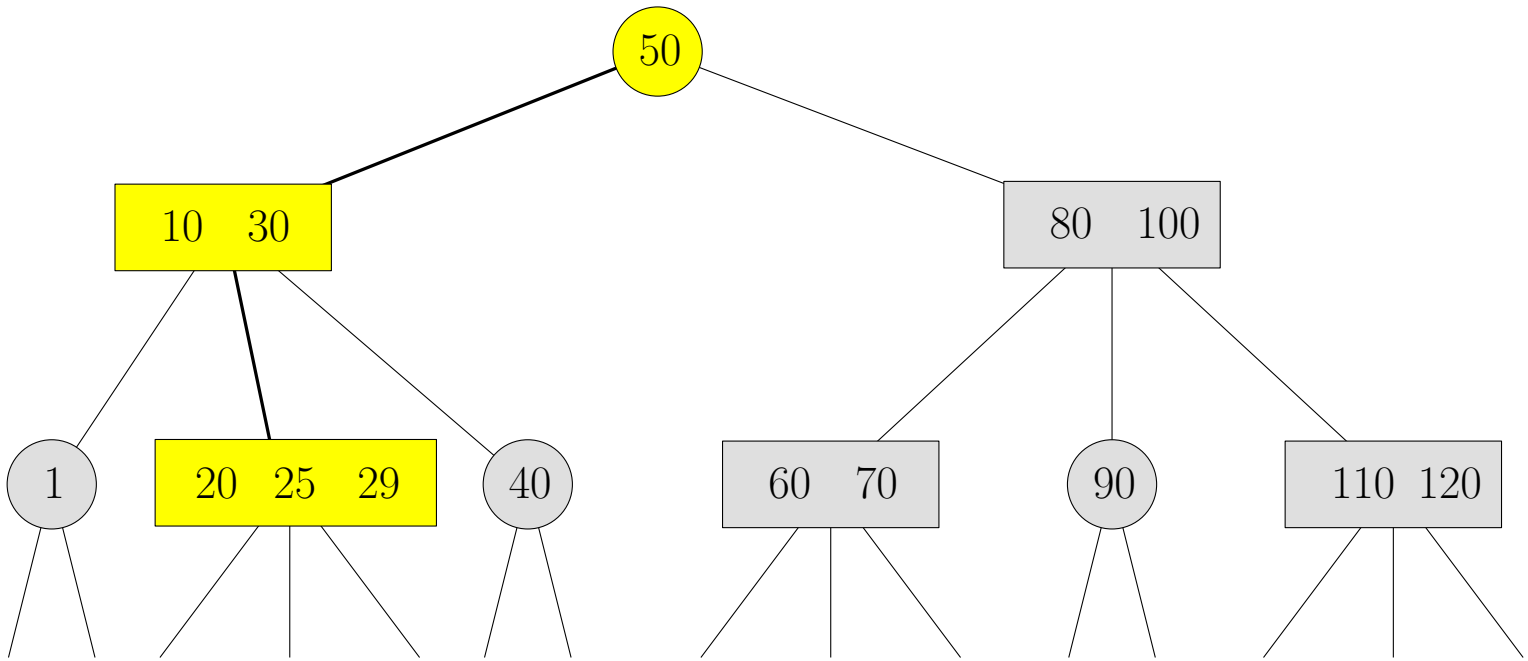
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 29



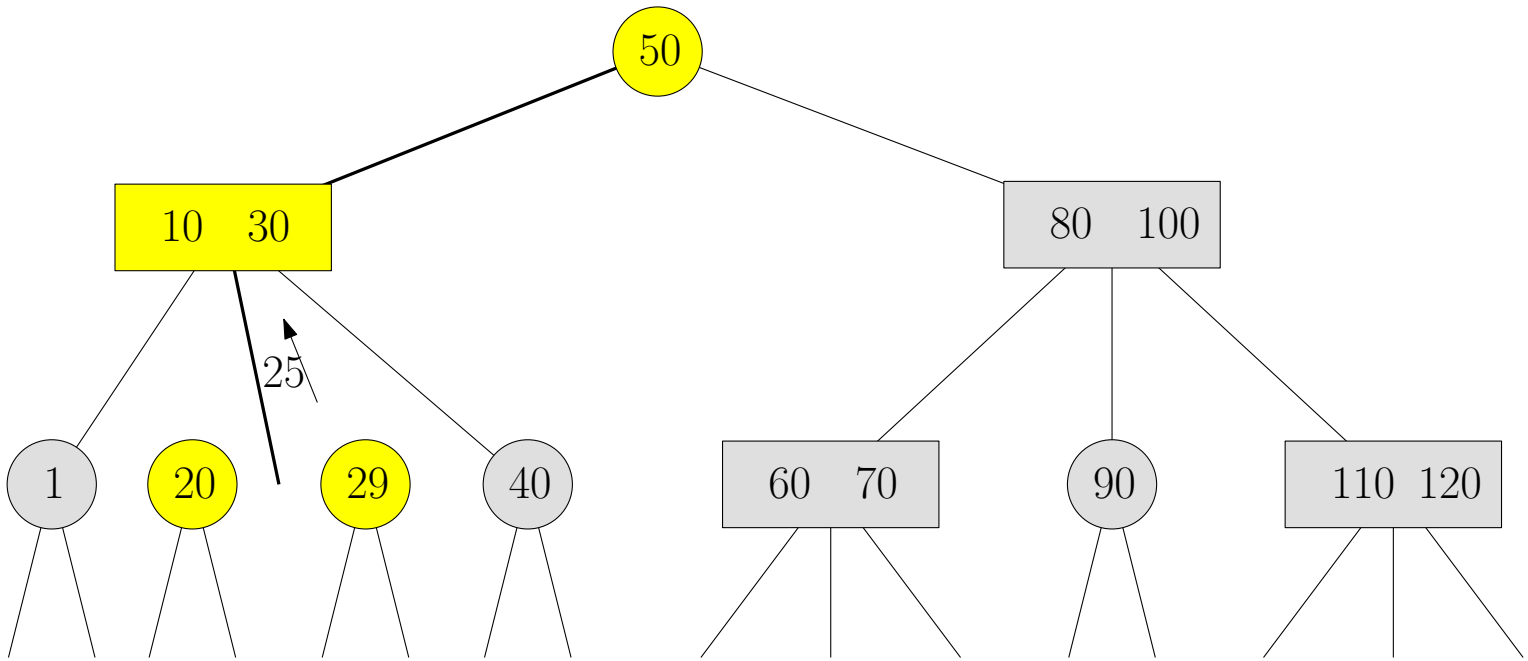
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 28



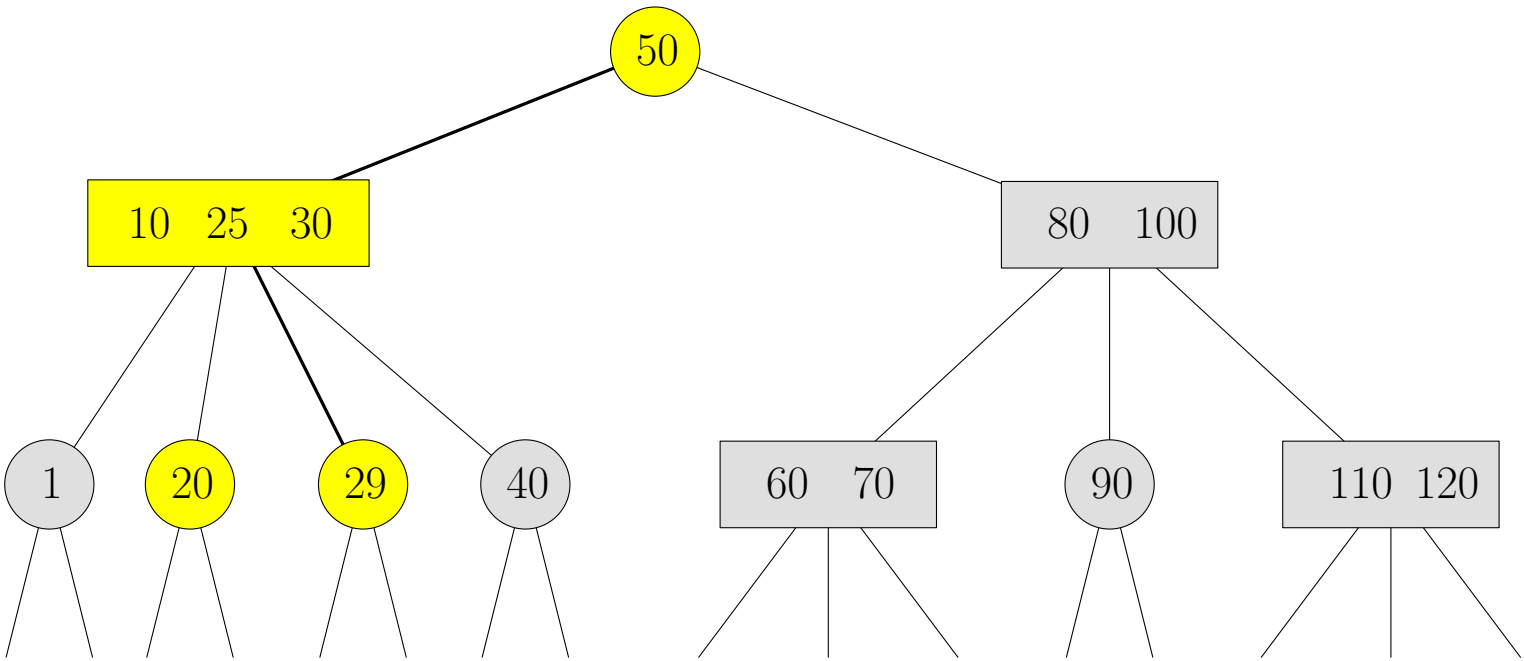
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 28



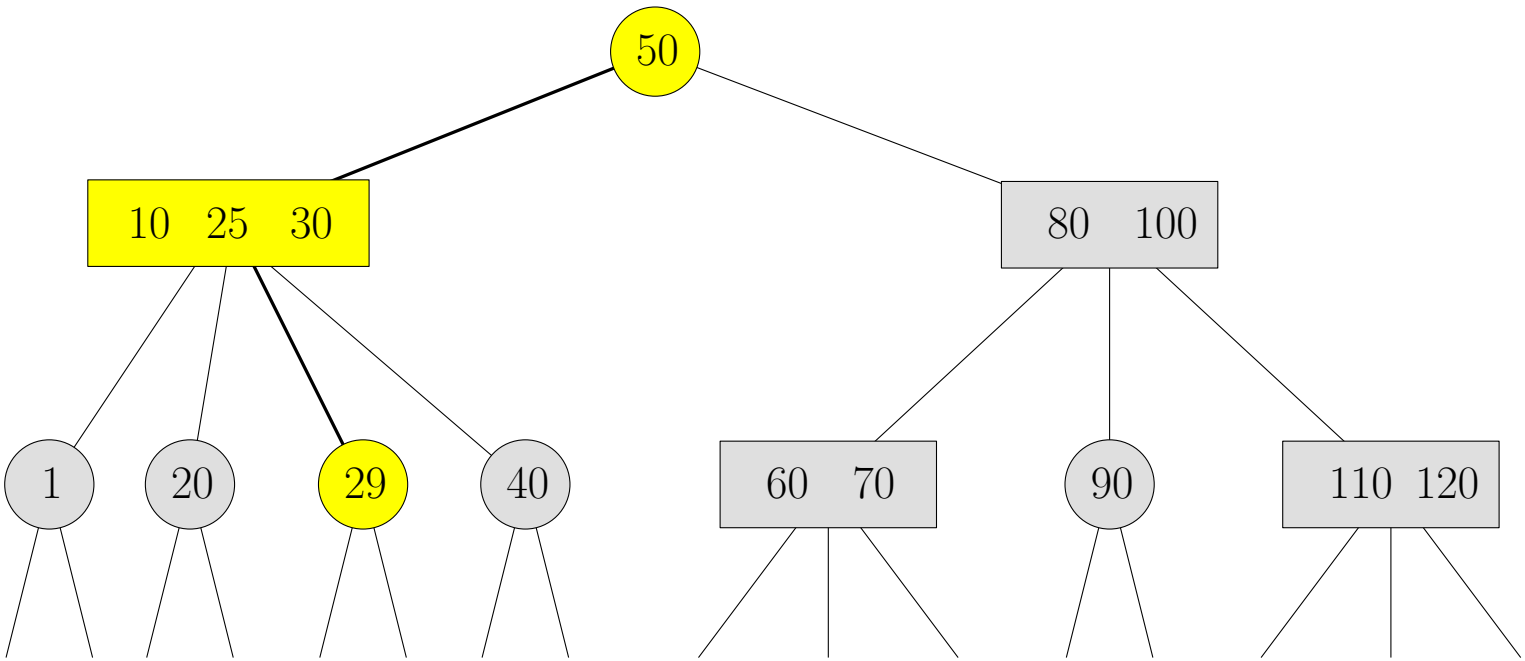
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 28



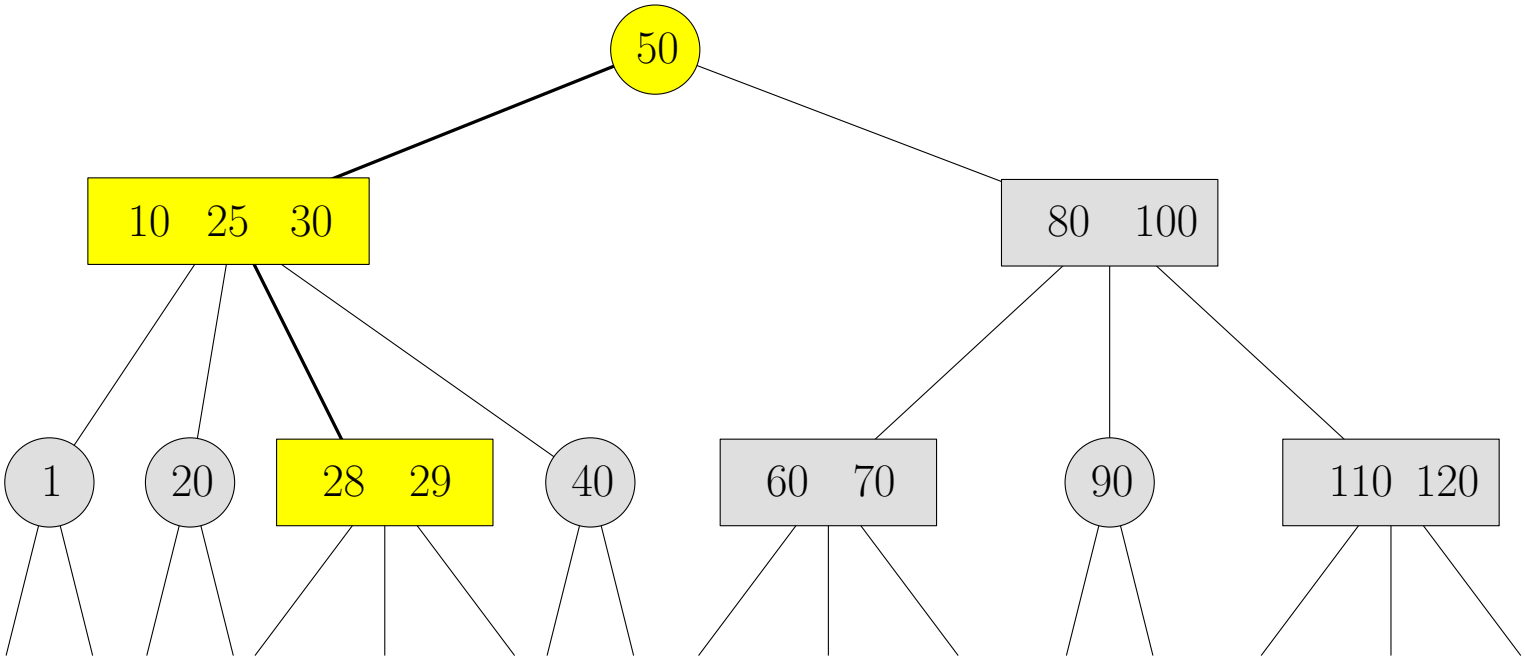
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 28



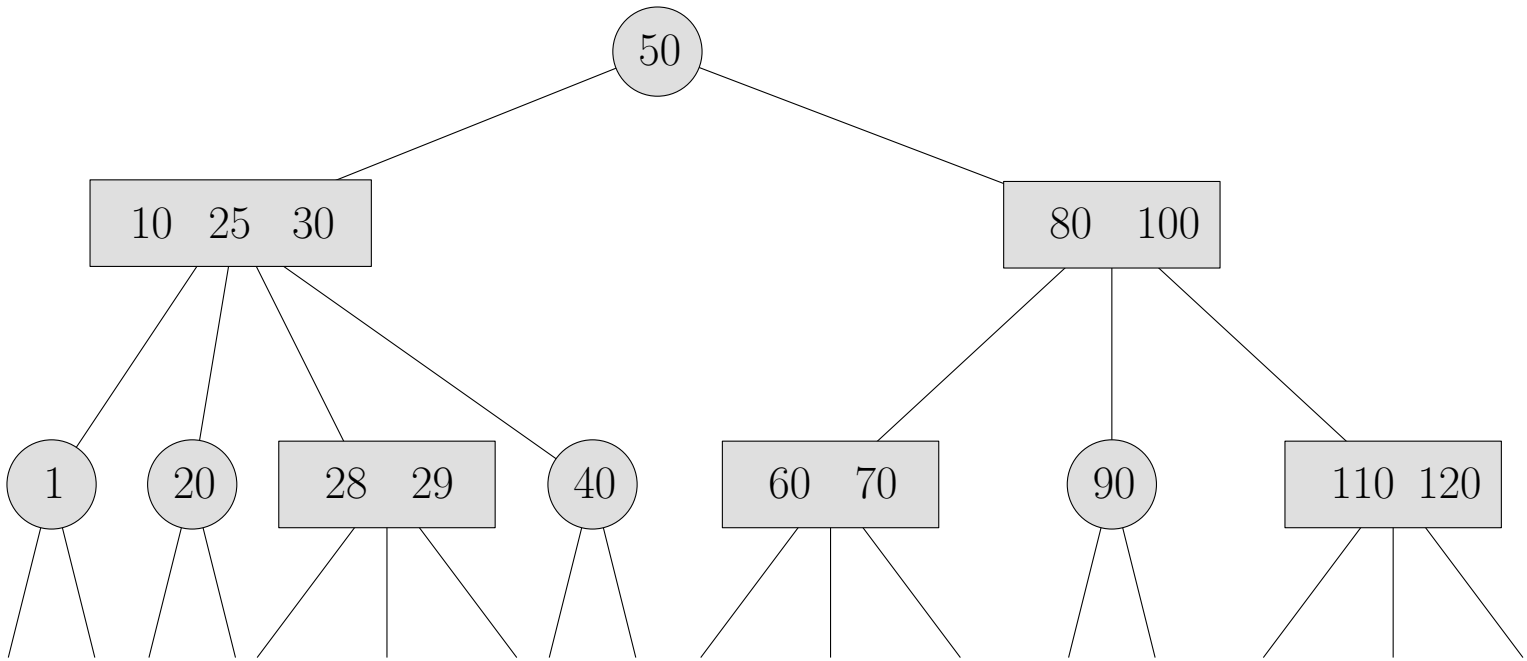
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 28



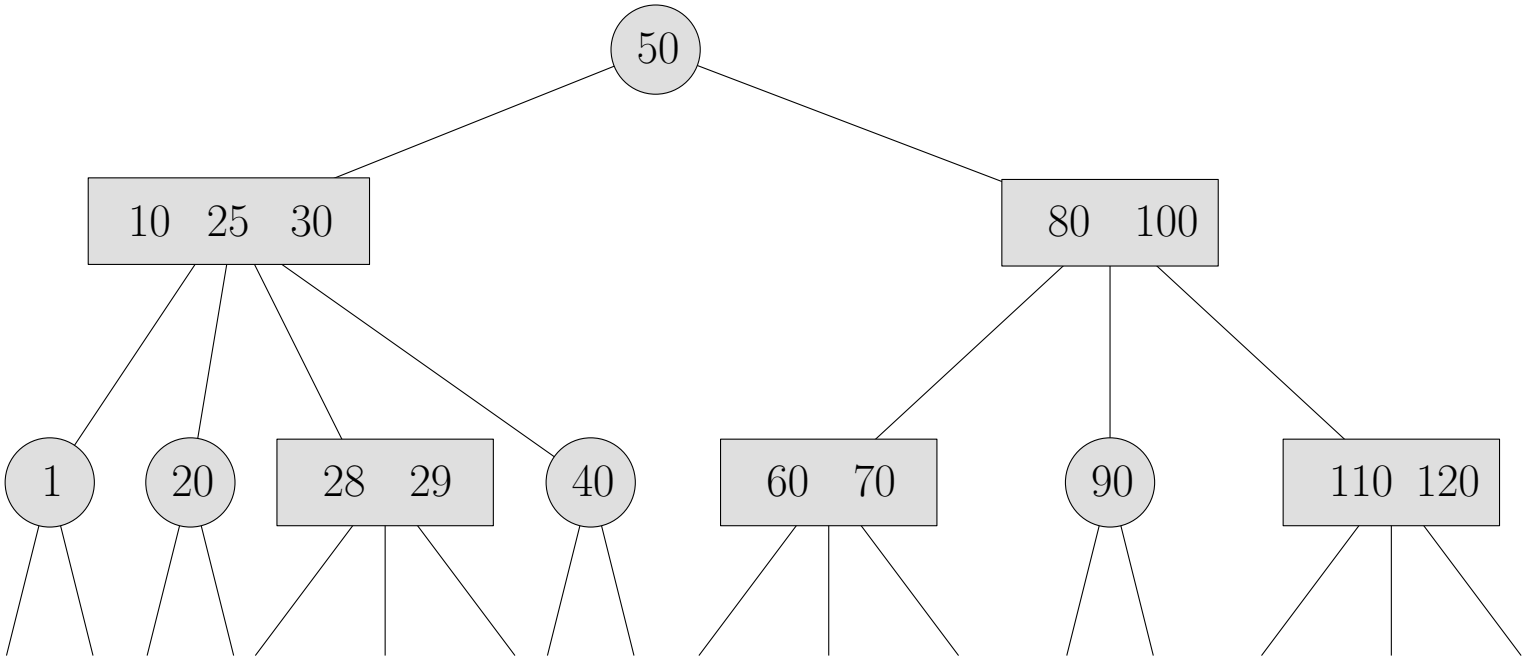
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 28



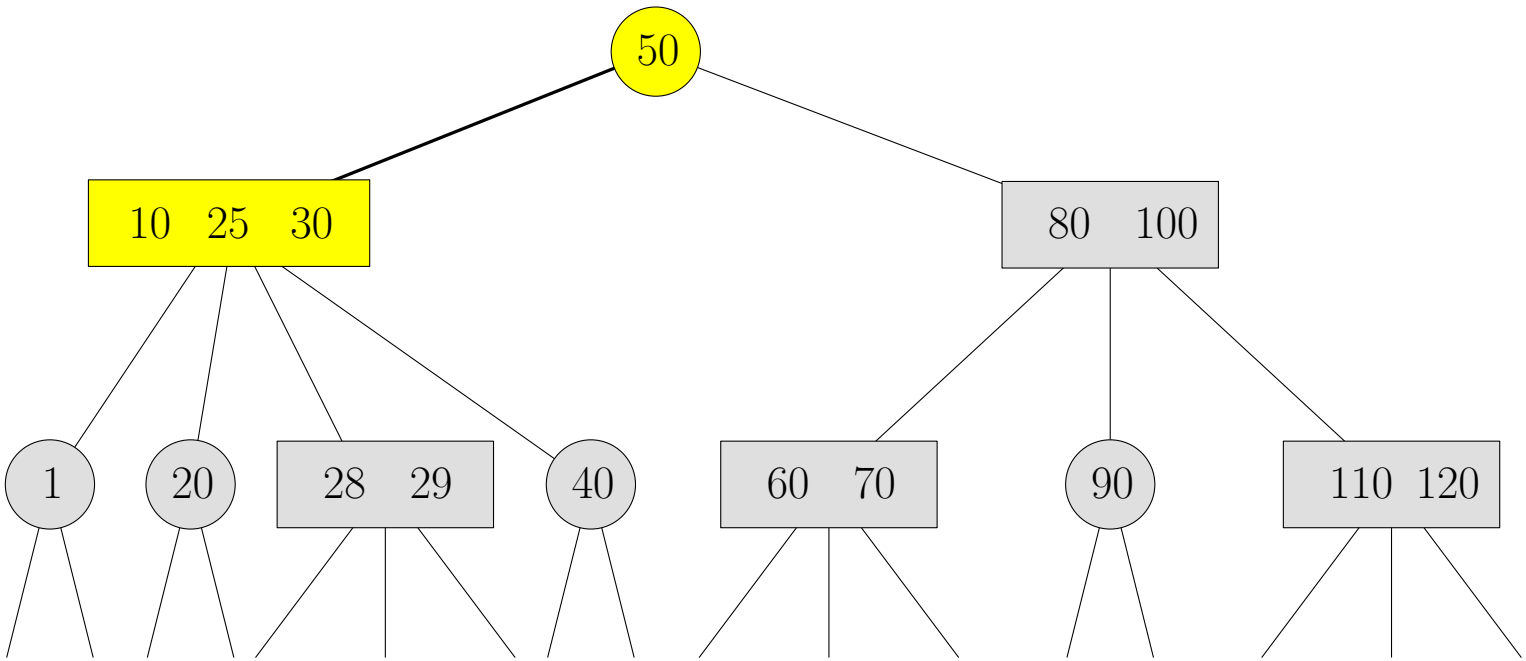
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 27



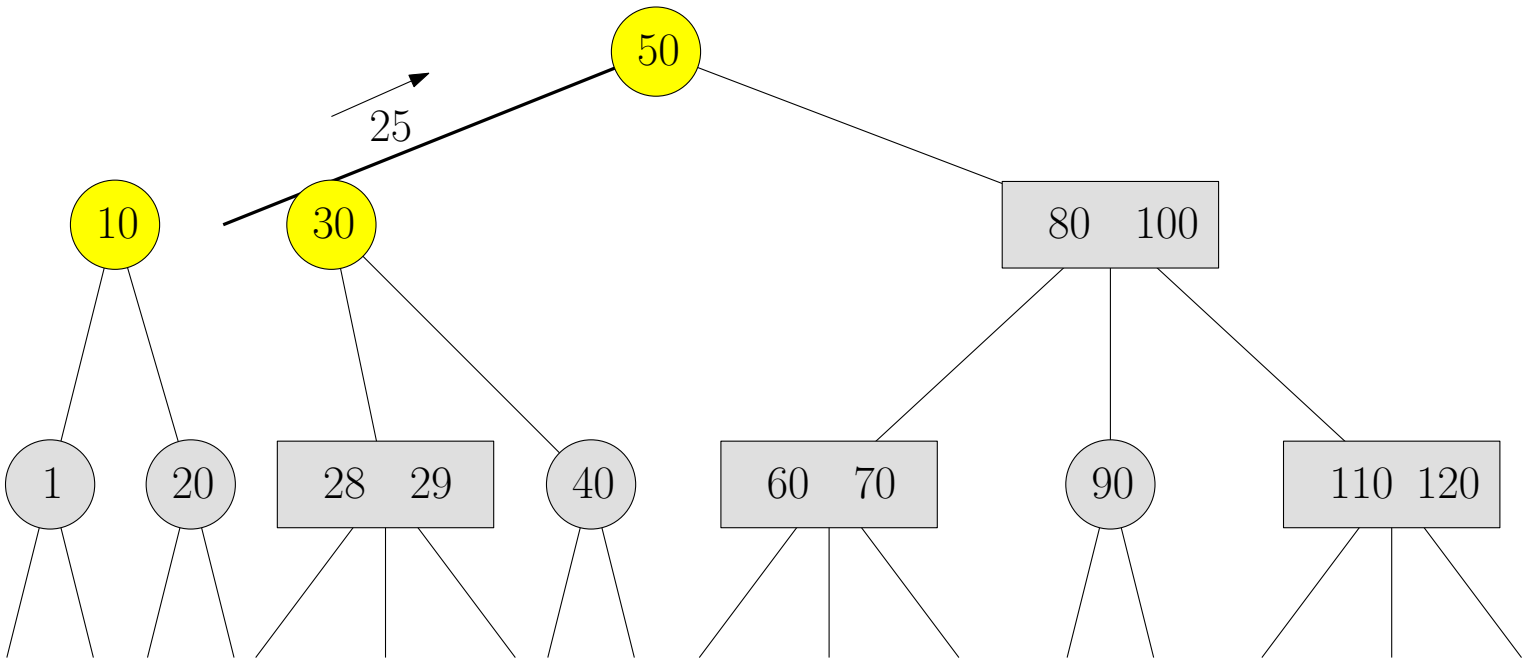
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 27



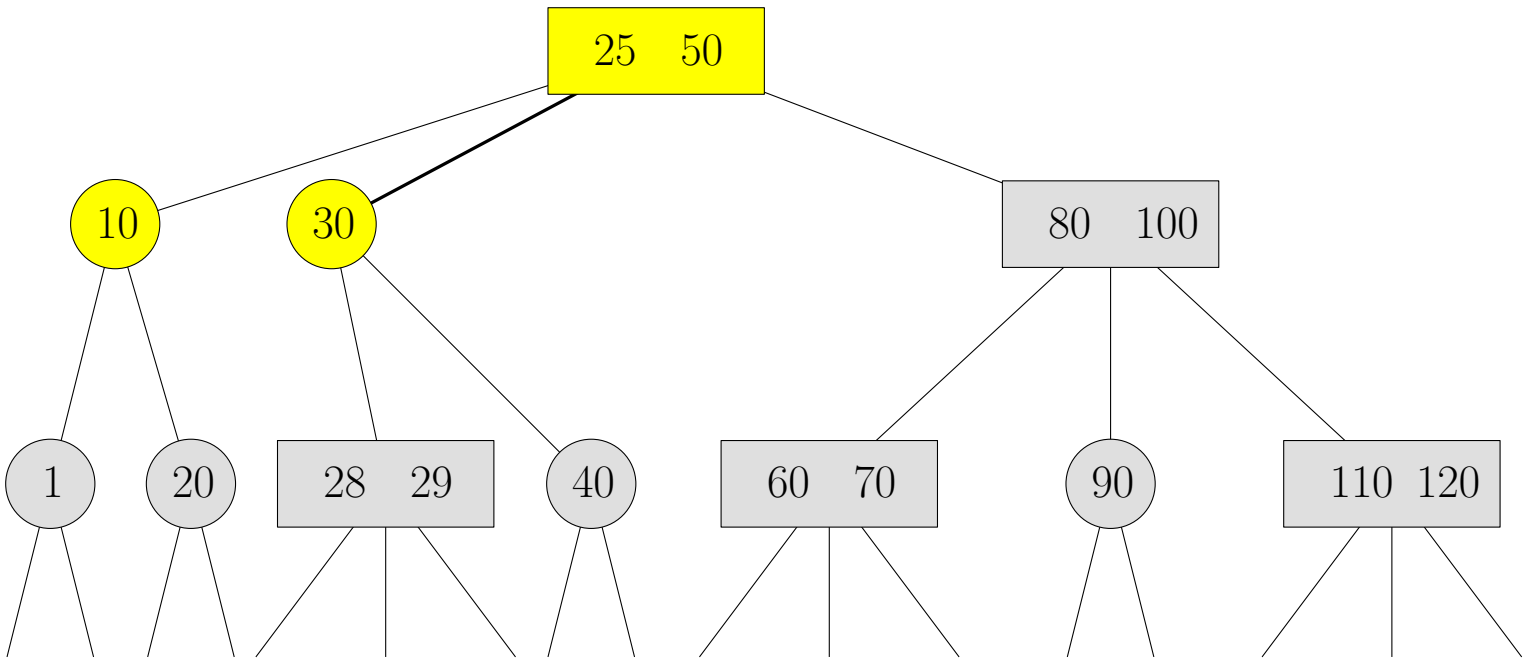
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 27



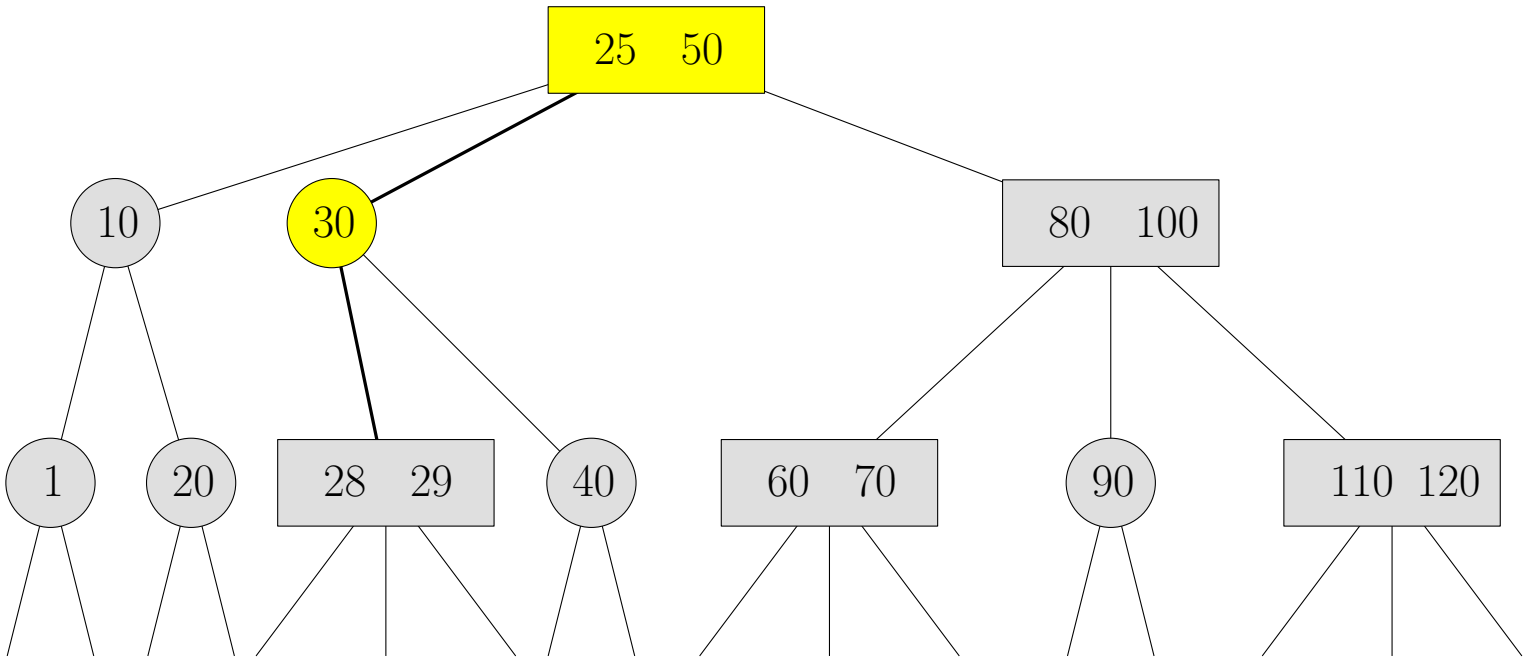
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 27



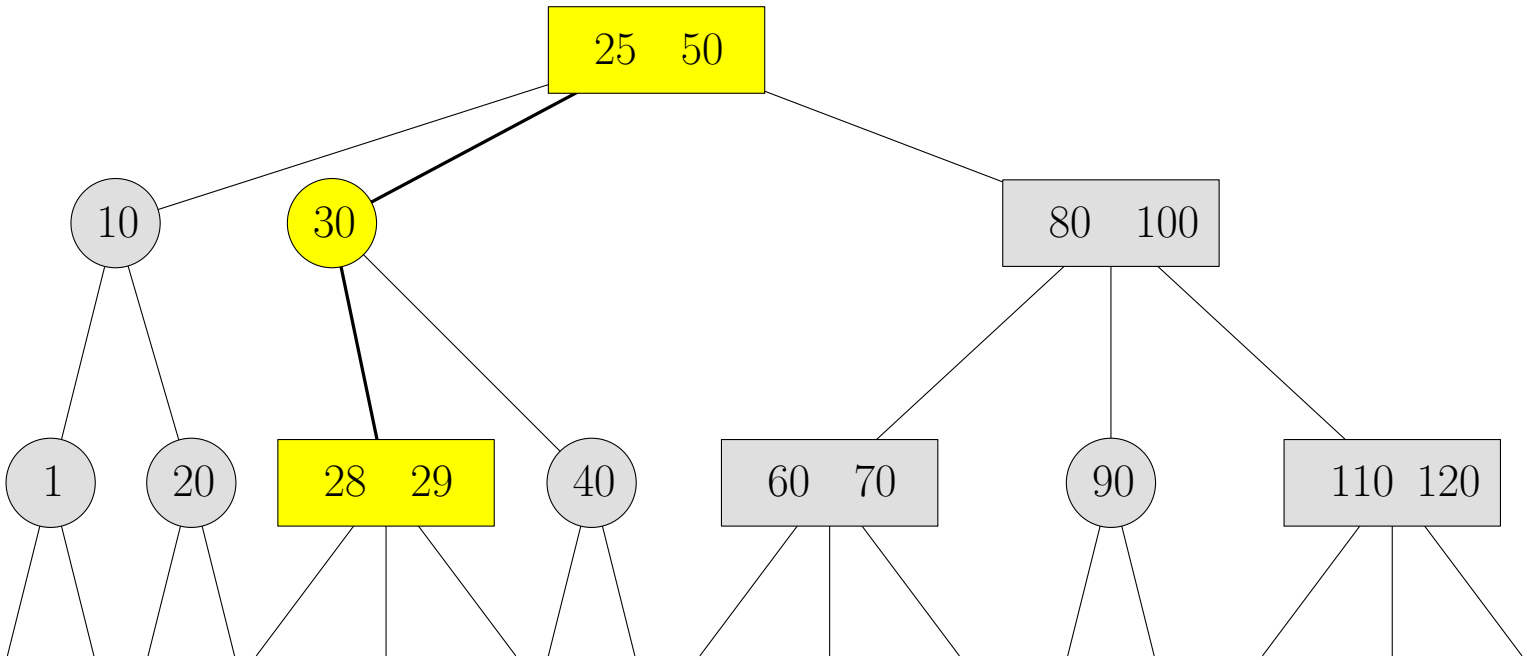
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 27



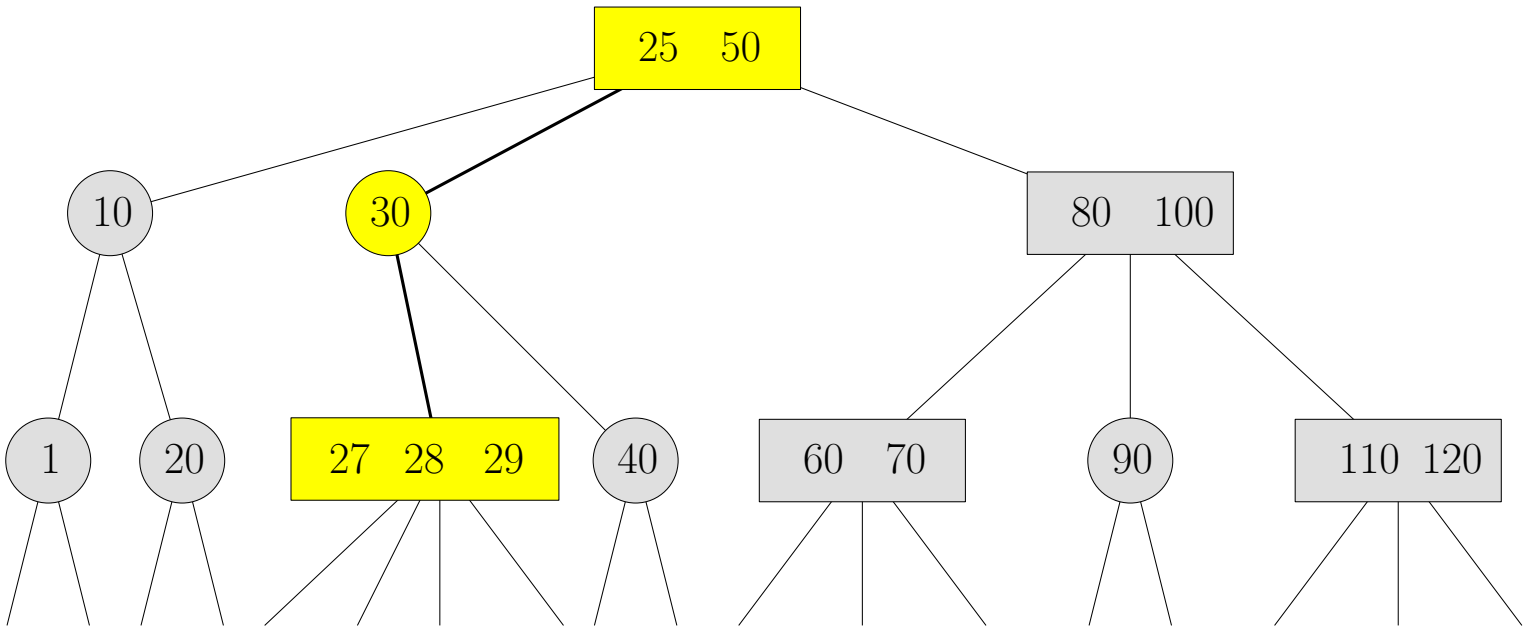
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 27



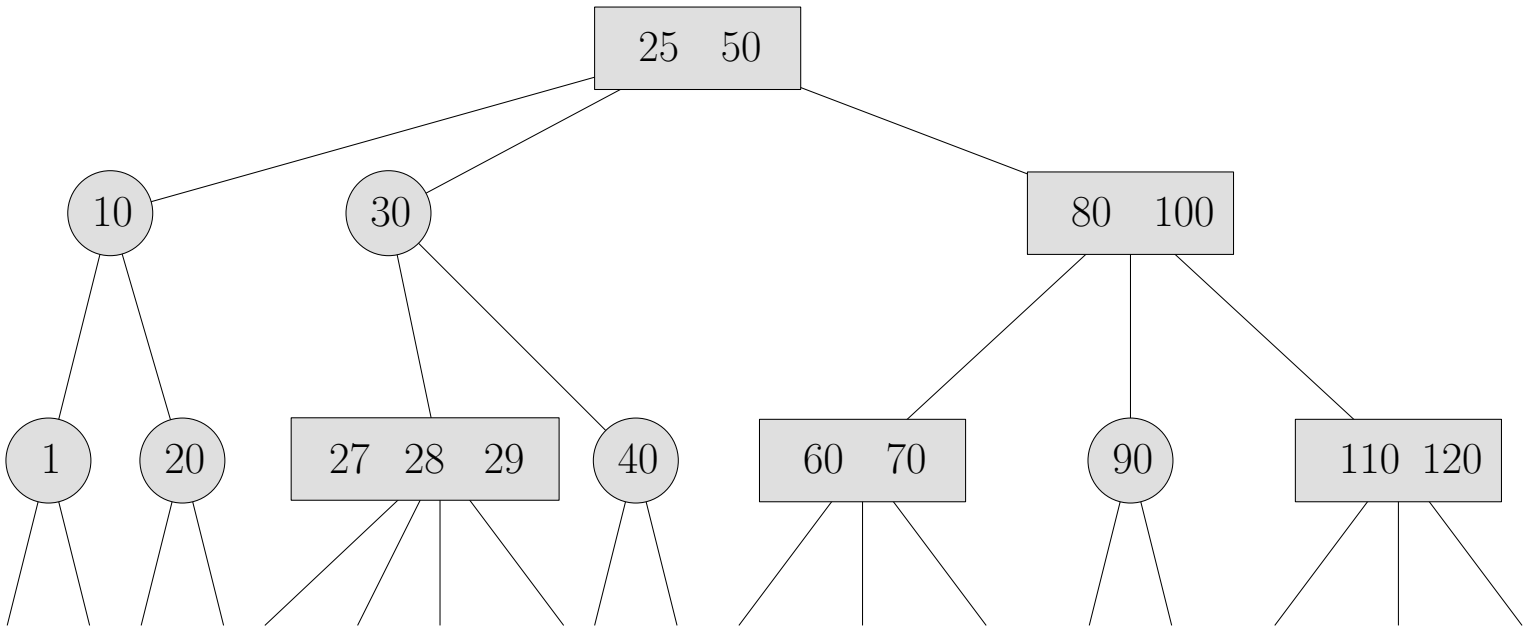
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 27



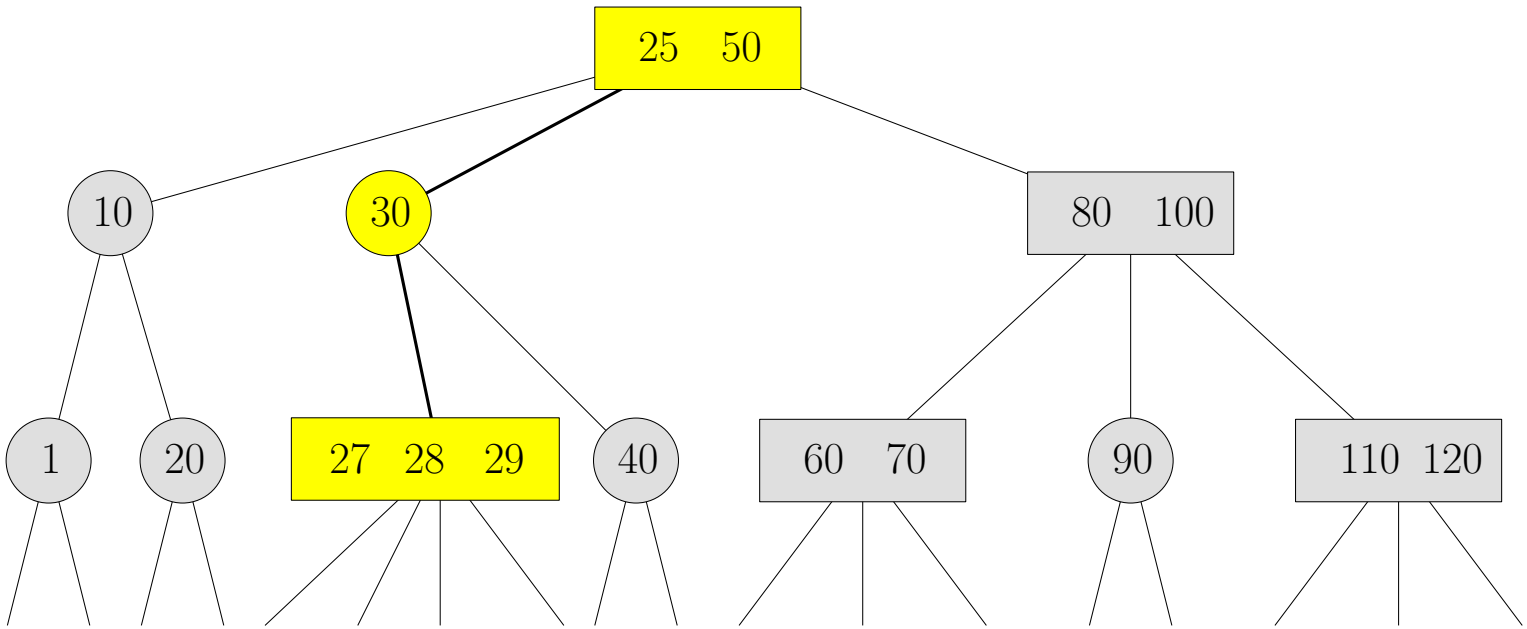
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 26



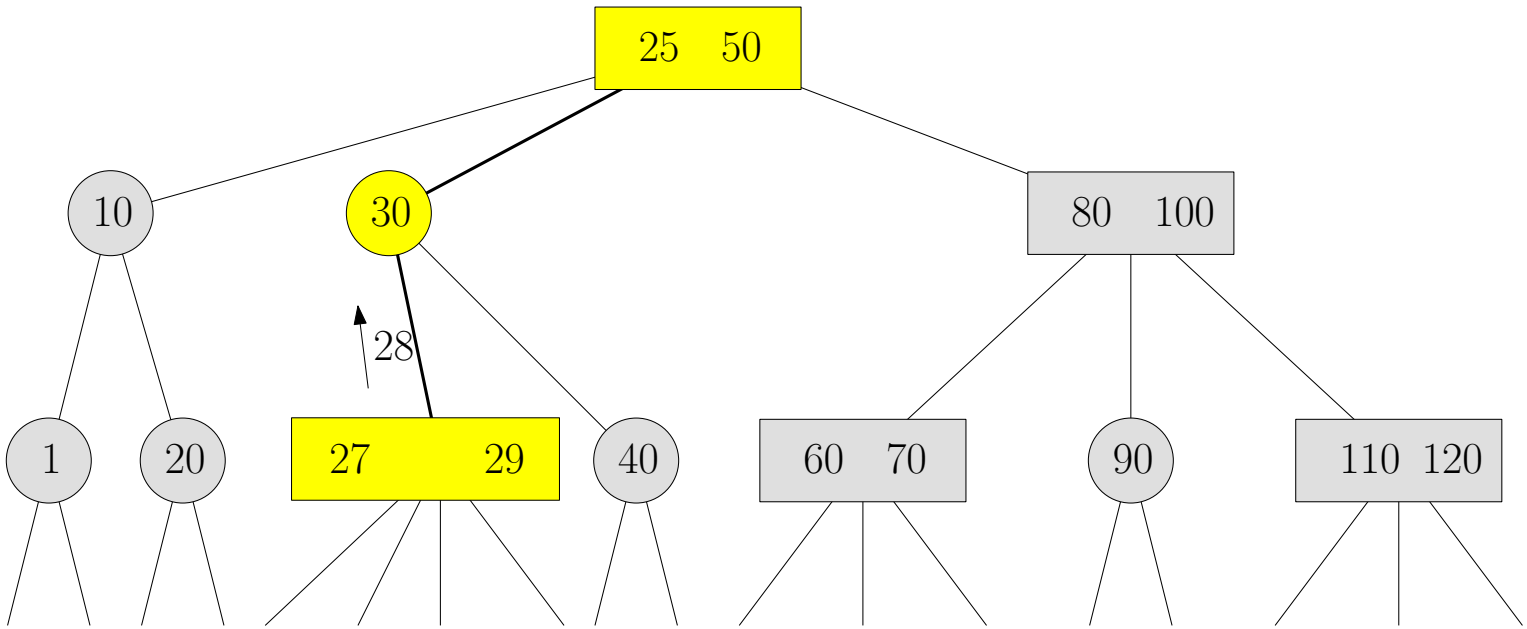
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 26



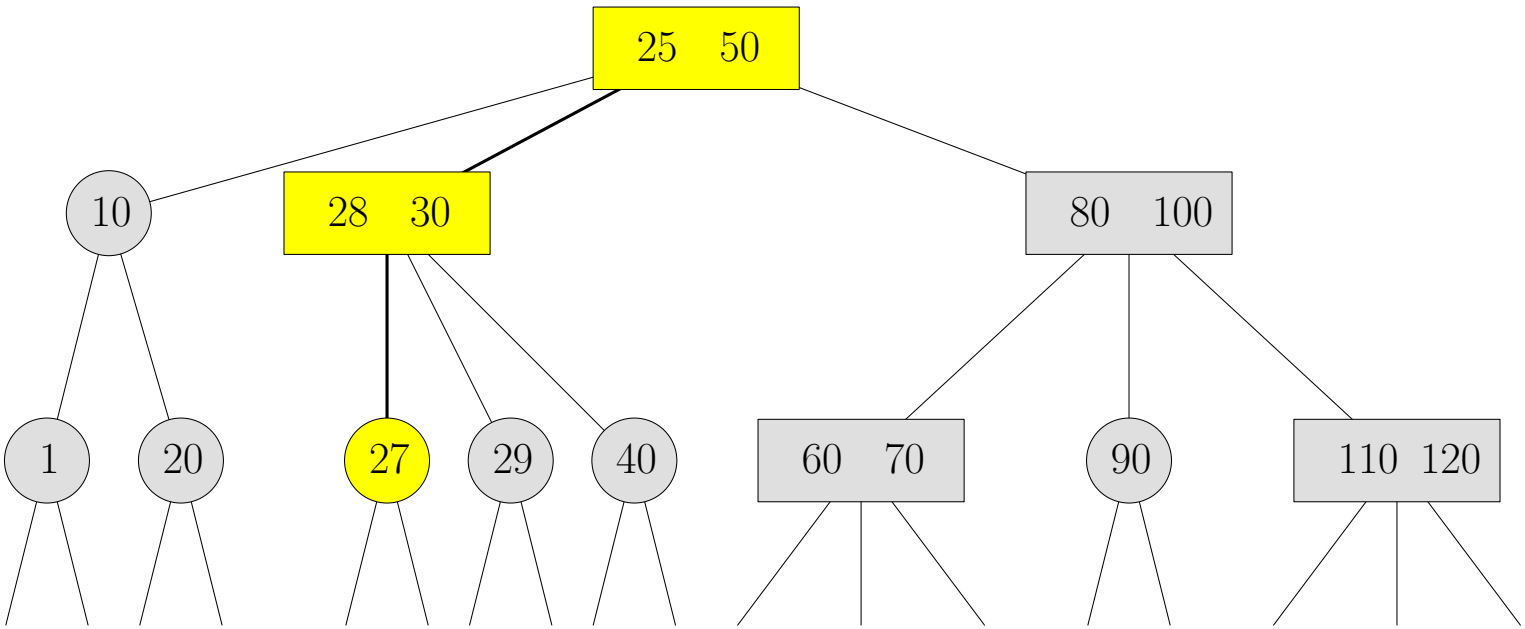
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 26



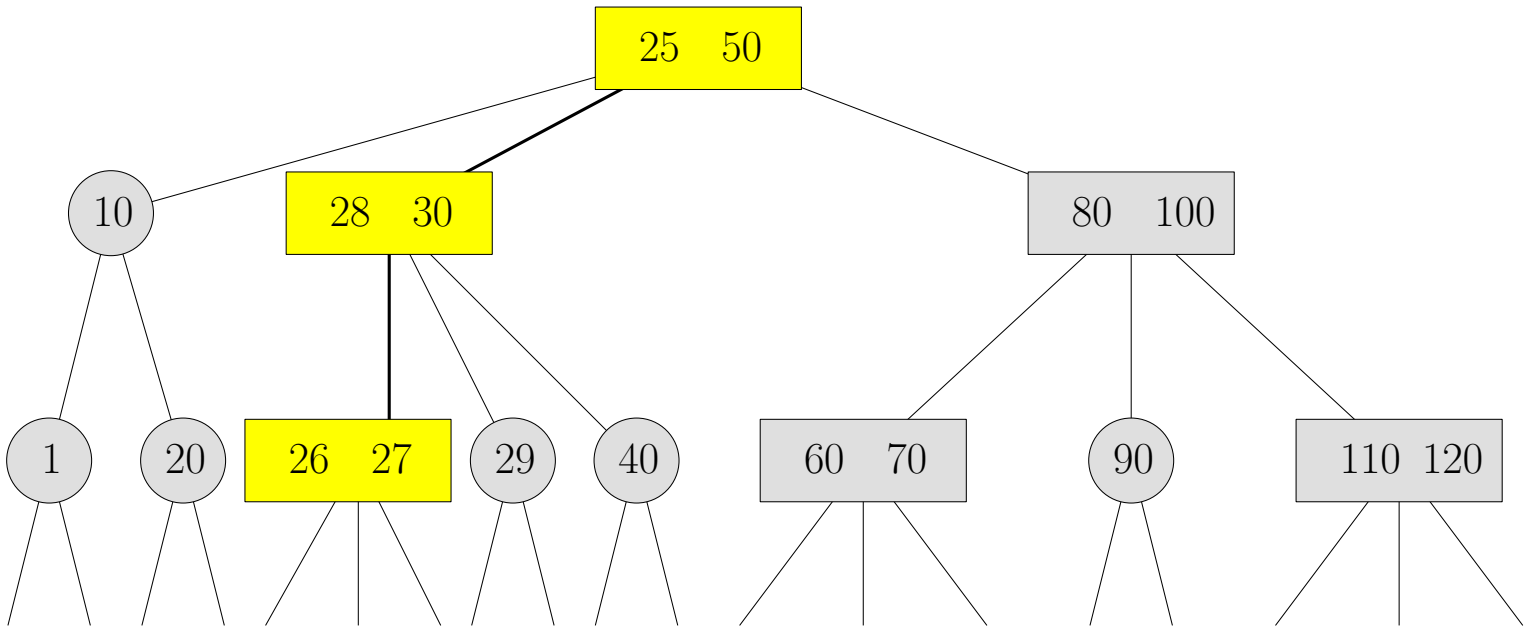
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 26



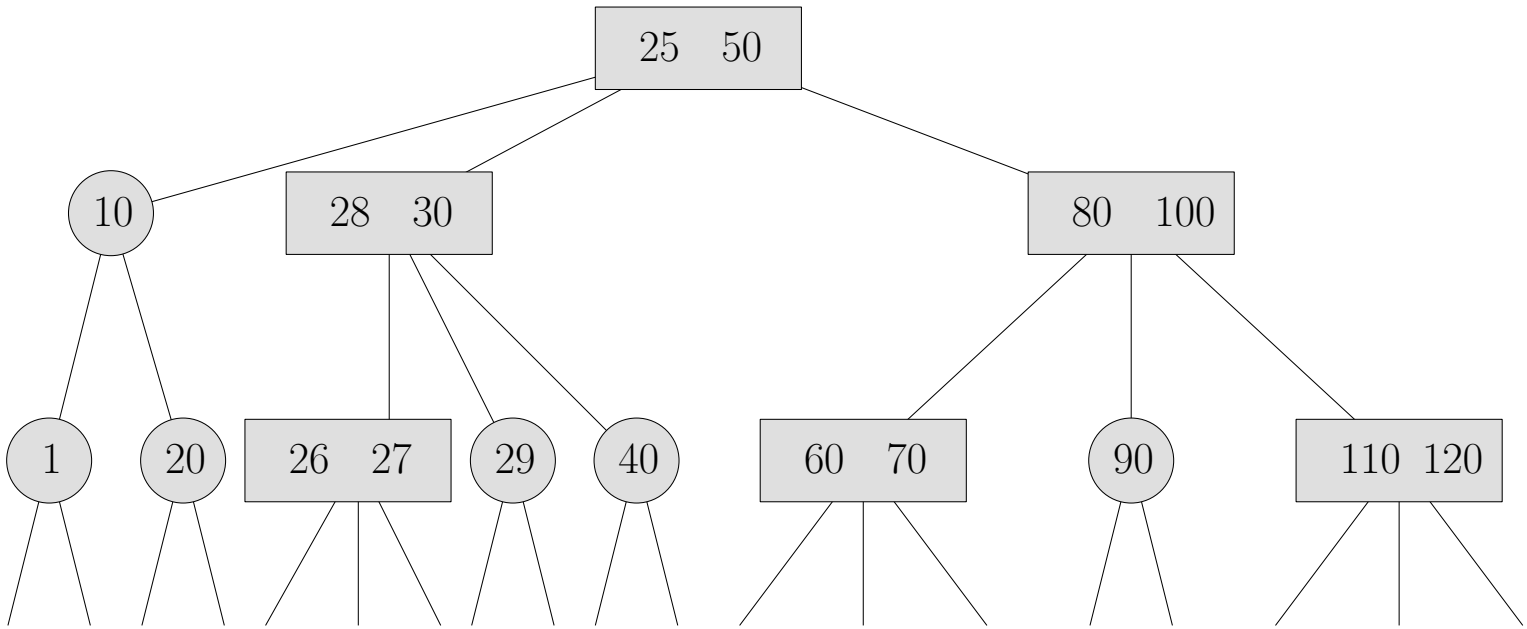
Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 26



Εισαγωγή σε Ισορροπημένο 2-3-4 Δέντρο

Εισαγωγή του 26

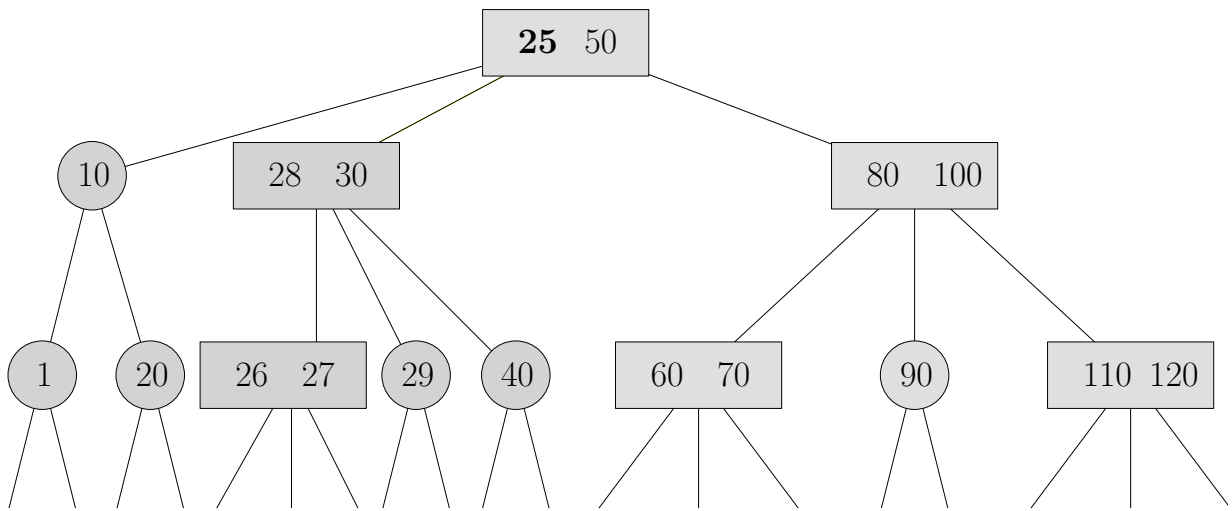


Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

Διαγραφή από μη-Φύλλο

- Κάνουμε αναγωγή του προβλήματος στο πρόβλημα διαγραφής από φύλλο
- Χρησιμοποιούμε την τεχνική των ΔΔΑ

π.χ για διαγραφή του 25, το αντικαθιστούμε πρώτα με τον *inorder successor (26)* ή με τον *inorder predecessor (20)*.

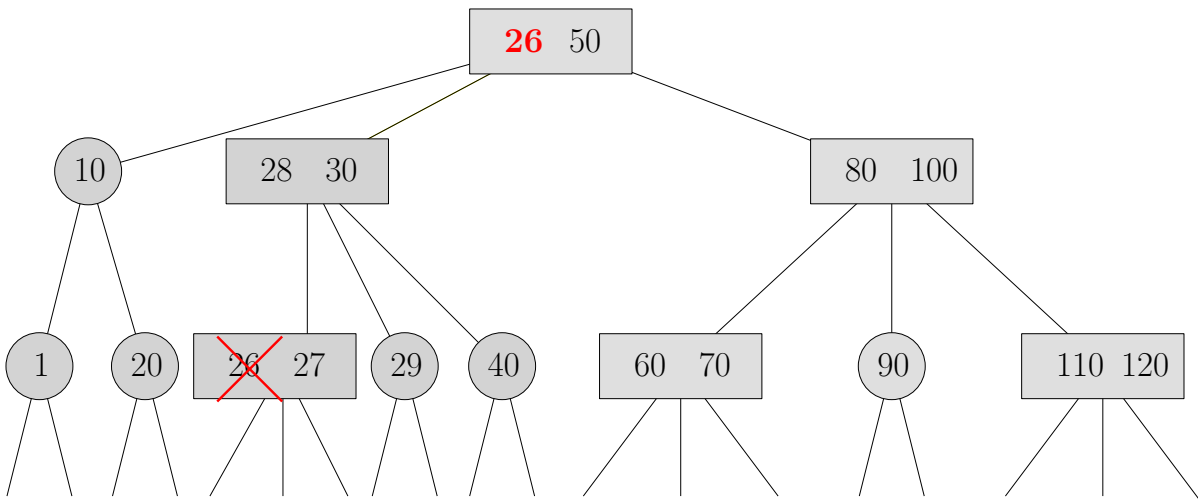


Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

Διαγραφή από μη-Φύλλο

- Κάνουμε αναγωγή του προβλήματος στο πρόβλημα διαγραφής από φύλλο
- Χρησιμοποιούμε την τεχνική των ΔΔΑ

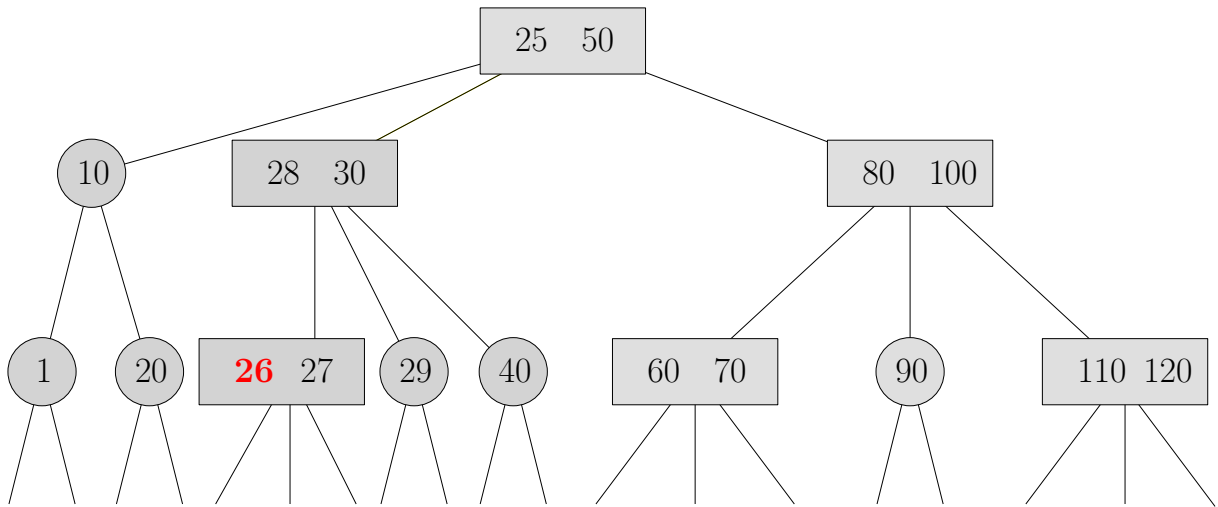
π.χ για διαγραφή του 25, το αντικαθιστούμε πρώτα με τον *inorder successor (26)* ή με τον *inorder predecessor (20)*.



Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

Διαγραφή από Φύλλο

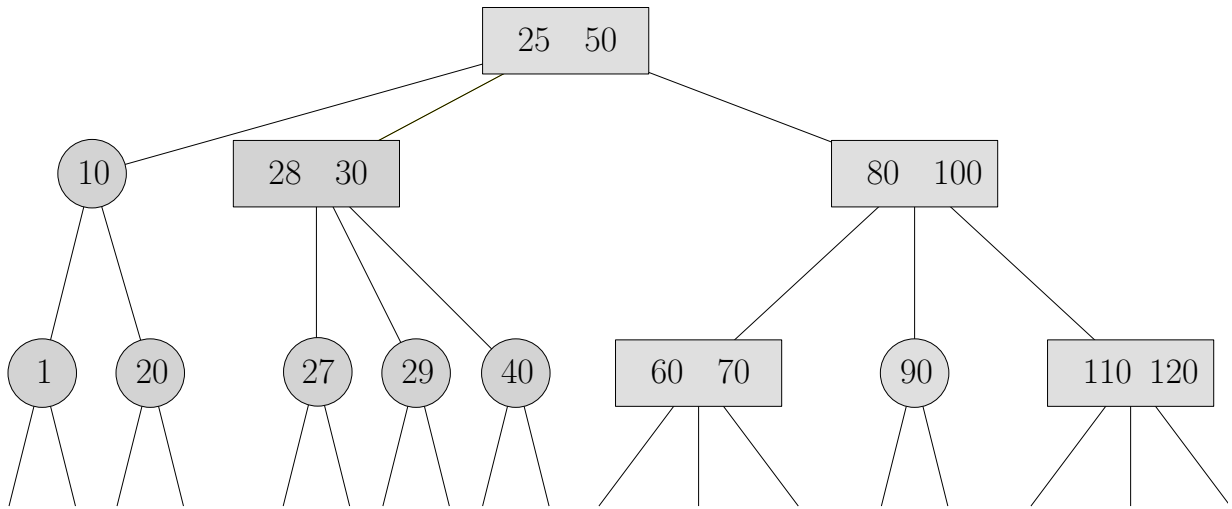
- Το φύλλο έχει ≥ 2 κλειδιά (π.χ διαγραφή του 26 παρακάτω), δηλαδή είναι είτε 3-κόμβος, είτε 4-κόμβος.
- Έυκολη περίπτωση, απλά το διαγράφουμε.



Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

Διαγραφή από Φύλλο

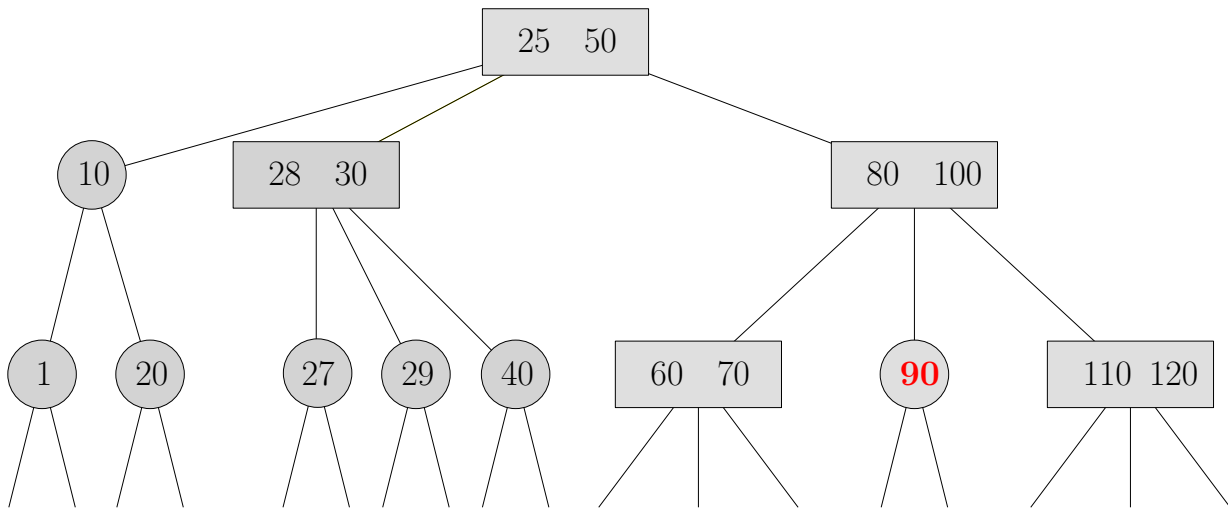
- Το φύλλο έχει ≥ 2 κλειδιά (π.χ διαγραφή του 26 παρακάτω), δηλαδή είναι είτε 3-κόμβος, είτε 4-κόμβος.
- Έυκολη περίπτωση, απλά το διαγράφουμε.



Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

Διαγραφή από Φύλλο

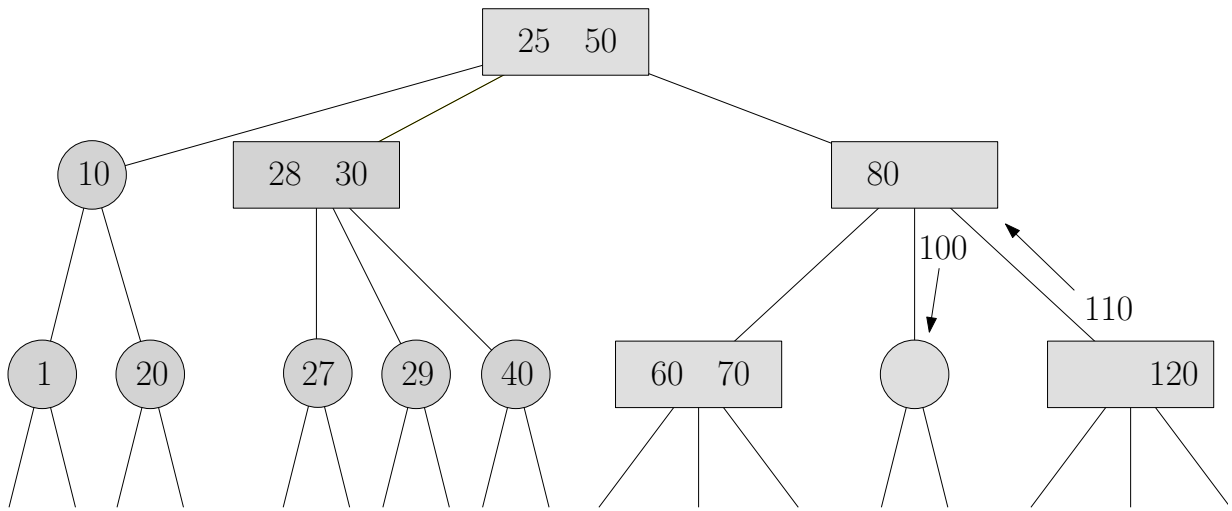
- Το φύλλο έχει 1 κλειδί (underflow), δηλαδή είναι 2-κόμβος.
- κάποιος αδελφός κόμβος (sibling node) έχει ≥ 2 κλειδιά
- **balance**: π.χ διαγραφή του 90



Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

Διαγραφή από Φύλλο

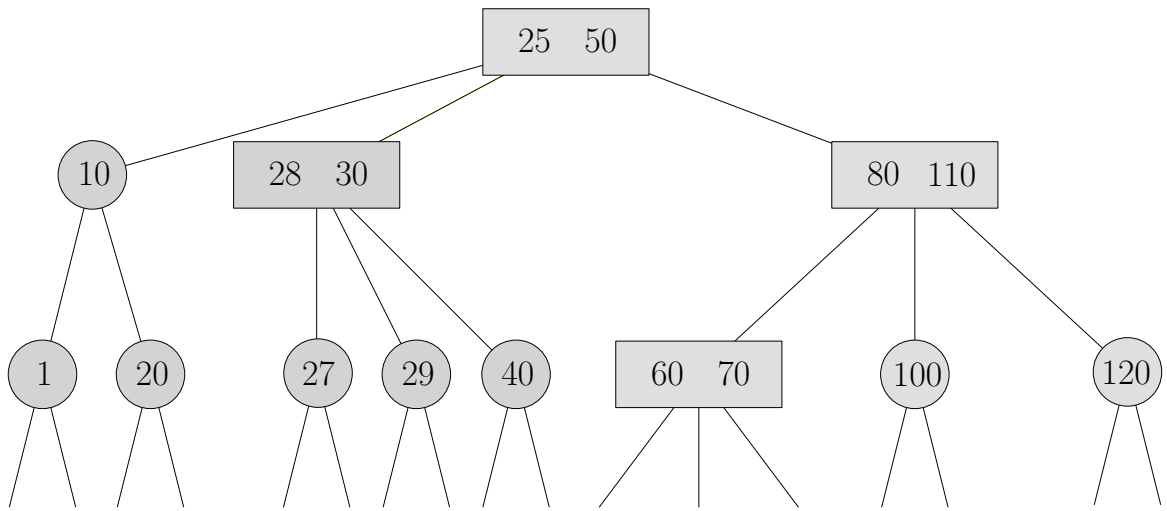
- Το φύλλο έχει 1 κλειδί (underflow), δηλαδή είναι 2-κόμβος.
- κάποιος αδελφός κόμβος (sibling node) έχει ≥ 2 κλειδιά
- **balance**: π.χ διαγραφή του 90



Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

Διαγραφή από Φύλλο

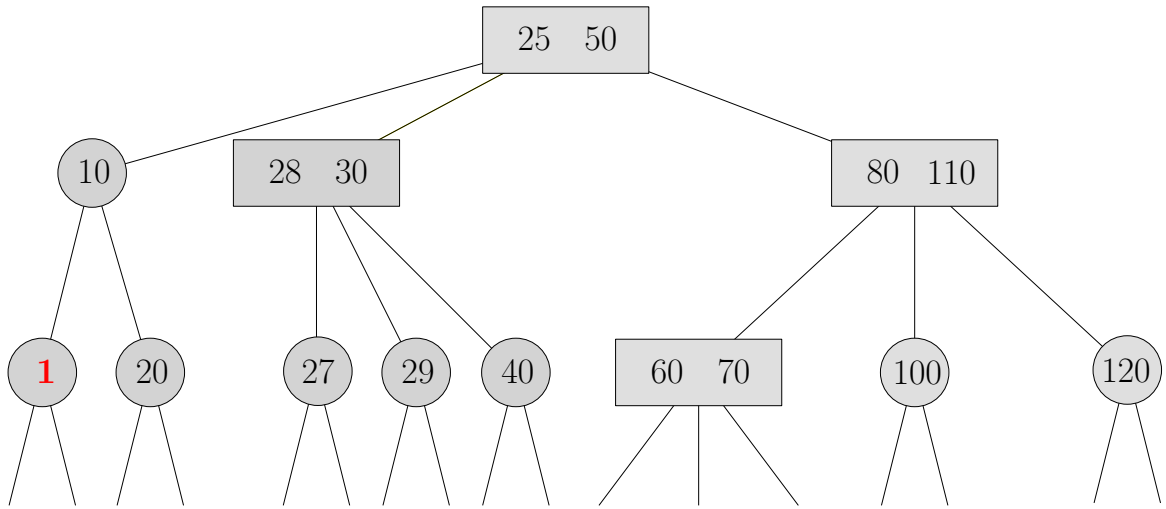
- Το φύλλο έχει 1 κλειδί (underflow), δηλαδή είναι 2-κόμβος.
- κάποιος αδελφός κόμβος (sibling node) έχει ≥ 2 κλειδιά
- **balance**: π.χ διαγραφή του 90



Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

Διαγραφή από Φύλλο

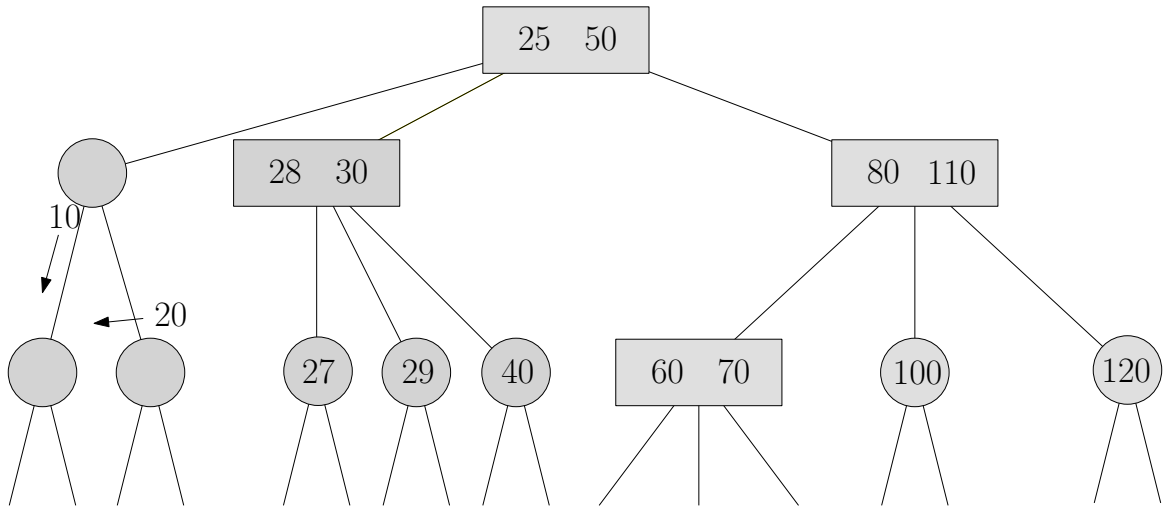
- Το φύλλο έχει 1 κλειδί (underflow), δηλαδή είναι 2-κόμβος.
- όλοι οι αδελφοί κόμβοι (sibling nodes) έχουν 1 κλειδί (είναι δηλαδή 2-κόμβοι)
- **fusion**: π.χ διαγραφή του 1



Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

Διαγραφή από Φύλλο

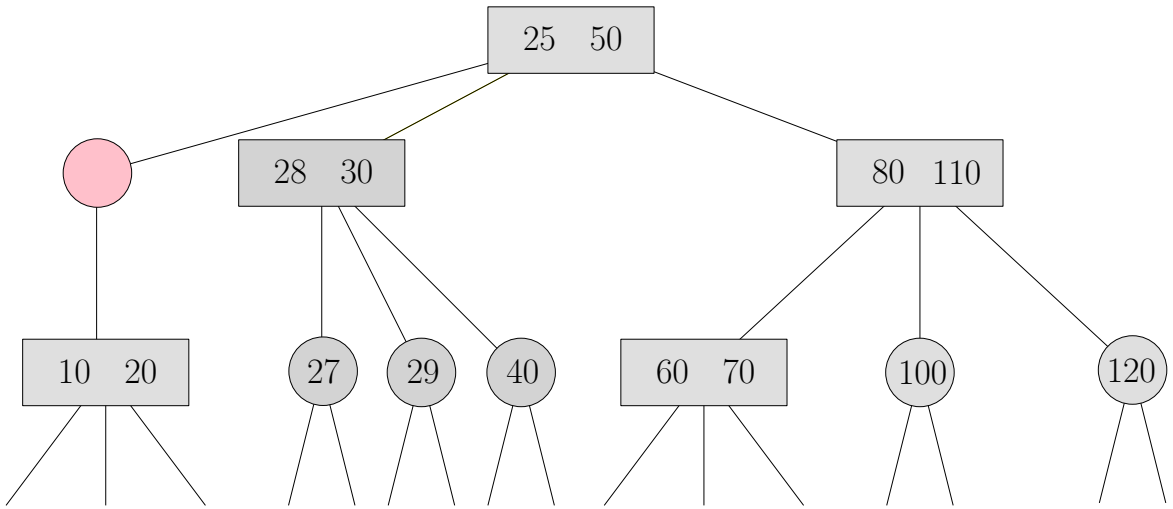
- Το φύλλο έχει 1 κλειδί (underflow), δηλαδή είναι 2-κόμβος.
- όλοι οι αδελφοί κόμβοι (sibling nodes) έχουν 1 κλειδί (είναι δηλαδή 2-κόμβοι)
- **fusion**: π.χ διαγραφή του 1



Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

Διαγραφή από Φύλλο

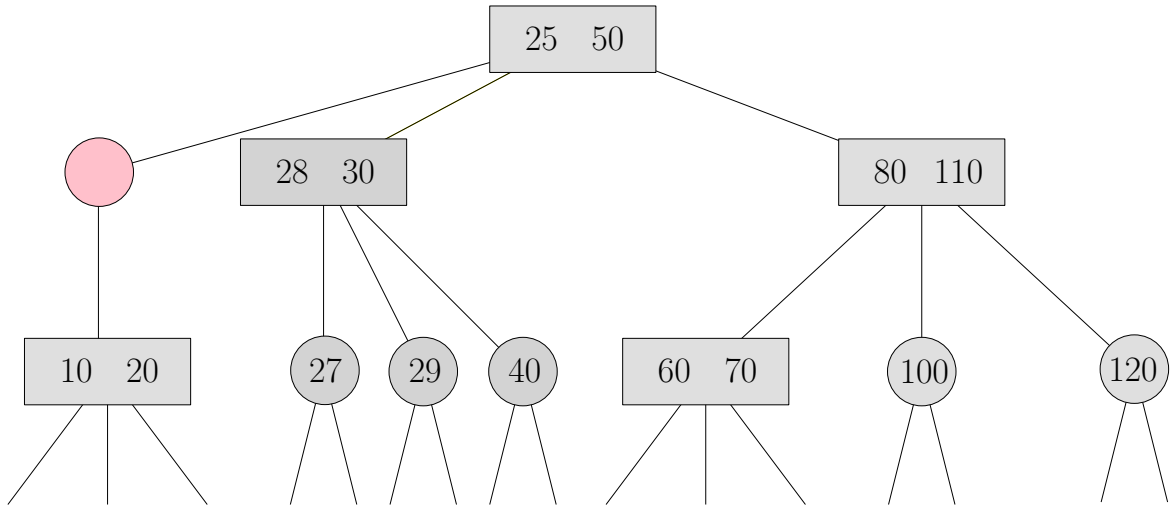
- Το φύλλο έχει 1 κλειδί (underflow), δηλαδή είναι 2-κόμβος.
- όλοι οι αδελφοί κόμβοι (sibling nodes) έχουν 1 κλειδί (είναι δηλαδή 2-κόμβοι)
- **fusion**: π.χ διαγραφή του 1



Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

Διαγραφή από Φύλλο

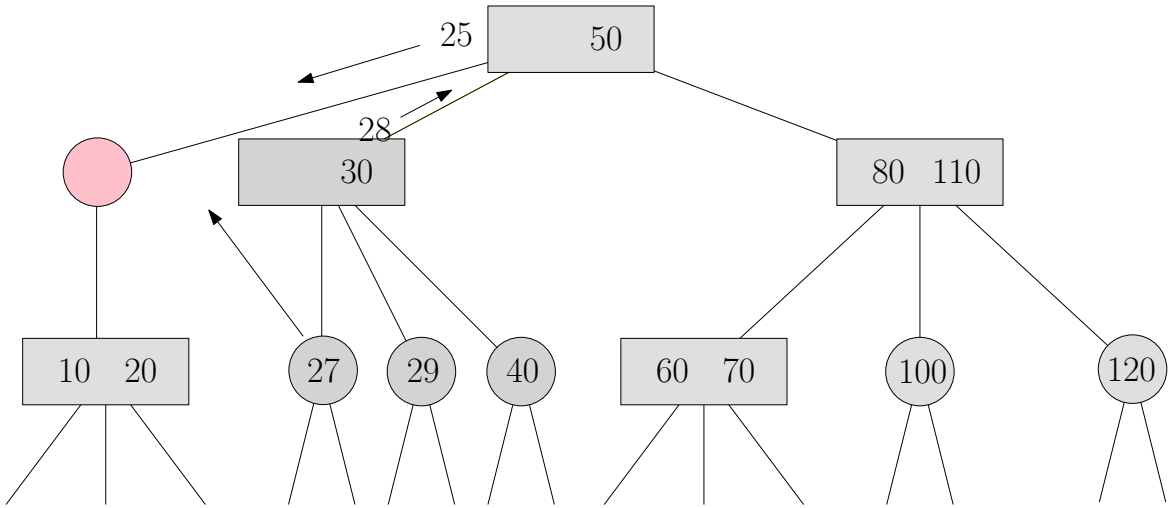
- Ο πατέρας έχει τώρα και αυτός πρόβλημα (underflow)
- επαλαμβάνουμε ή balance ή fusion ανάλογα με τους αδελφούς κόμβους του



Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

Διαγραφή από Φύλλο

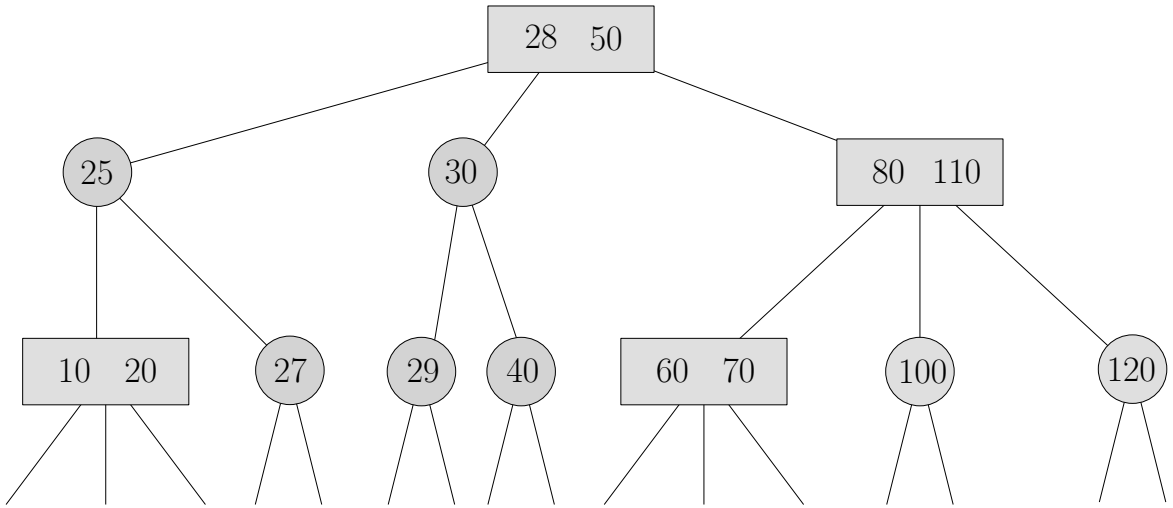
- Ο πατέρας έχει τώρα και αυτός πρόβλημα (underflow)
- επαλαμβάνουμε ή balance ή fusion ανάλογα με τους αδελφούς κόμβους του



Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

Διαγραφή από Φύλλο

- Ο πατέρας έχει τώρα και αυτός πρόβλημα (underflow)
- επαλαμβάνουμε ή balance ή fusion ανάλογα με τους αδελφούς κόμβους του



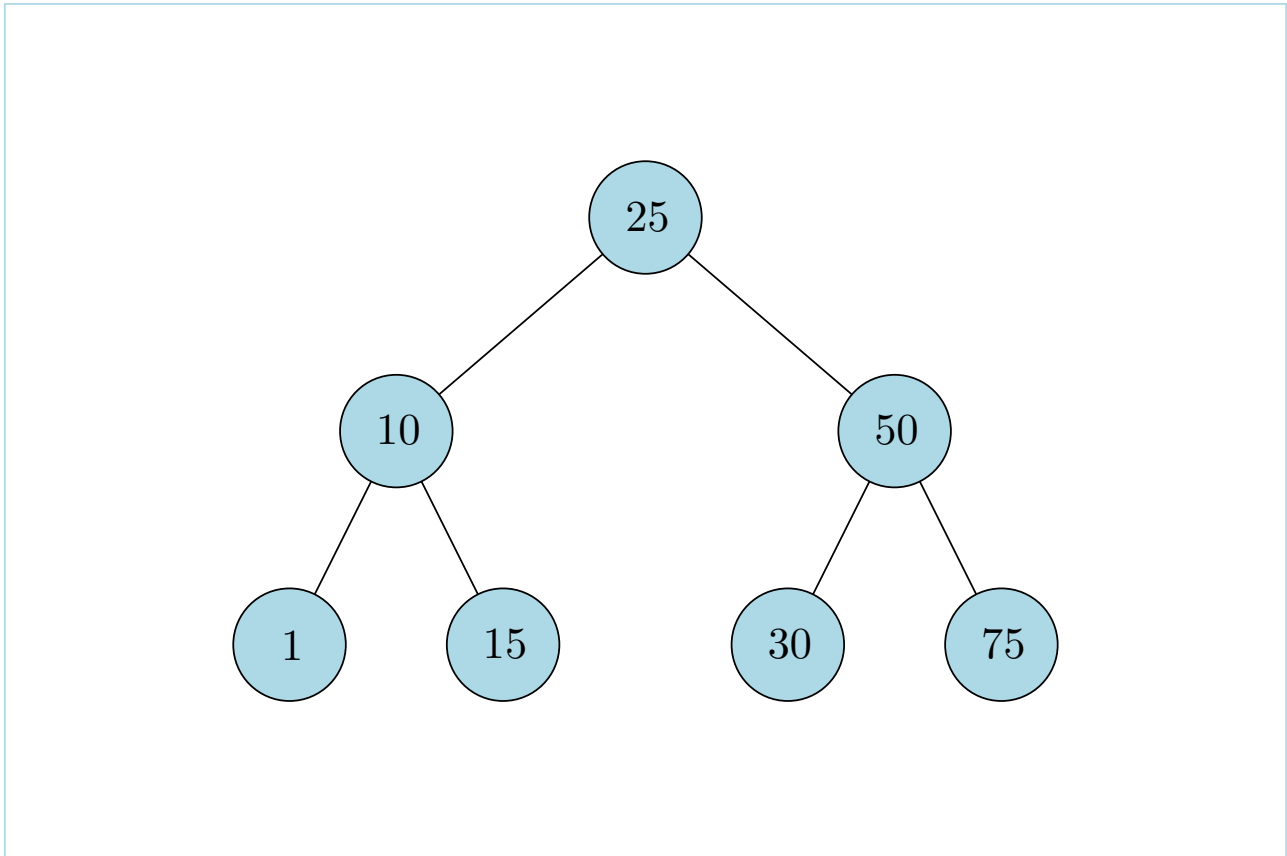
Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

- Το fusion μπορεί να γίνει μέχρι και την ρίζα.
- Σε περίπτωση που φτάσει στην ρίζα το ύψος του δέντρου μειώνεται κατά ένα.

Η διαγραφή υλοποιείτε σε χρόνο $\mathcal{O}(\log n)$.

Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

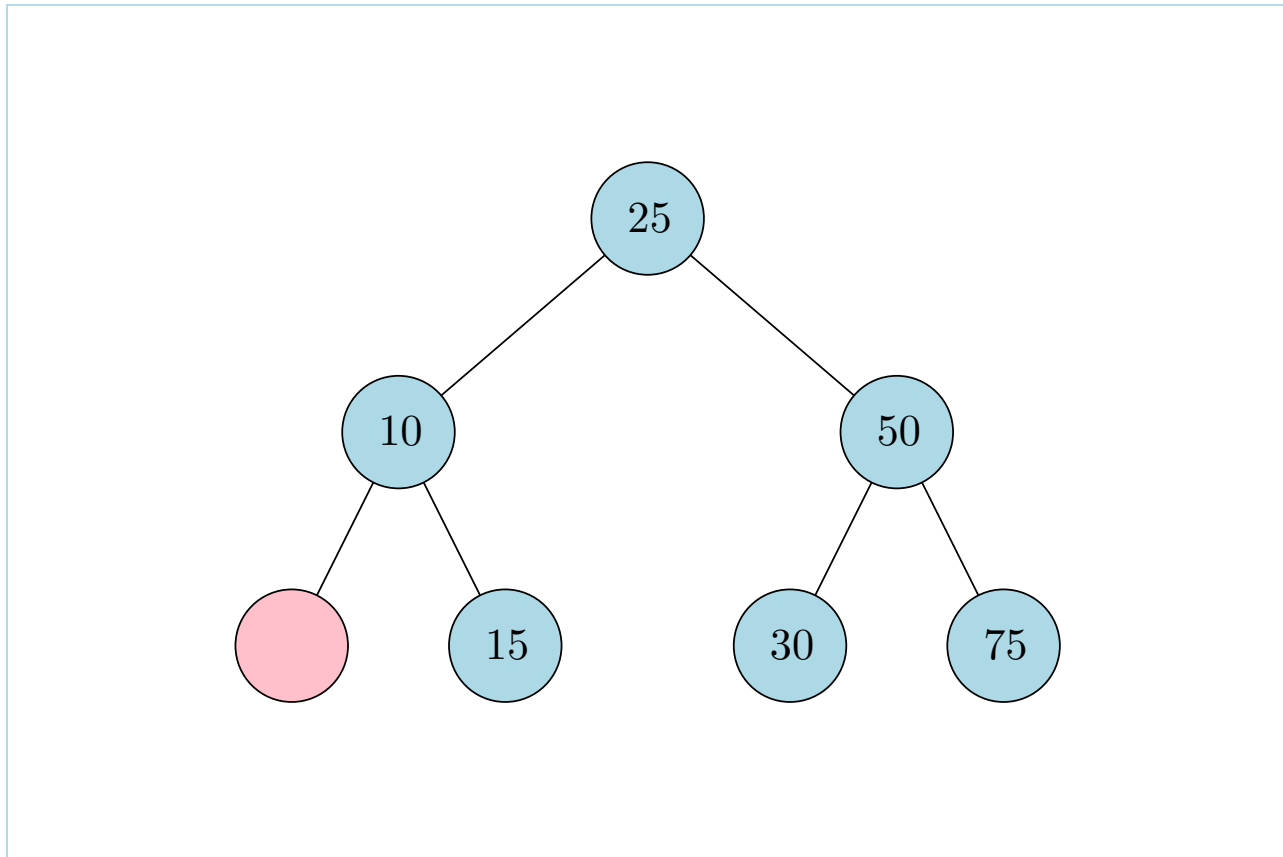
Παράδειγμα αλυσιδωτής διαγραφής μέχρι την ρίζα



Διαγραφή του κλειδιού 1

Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

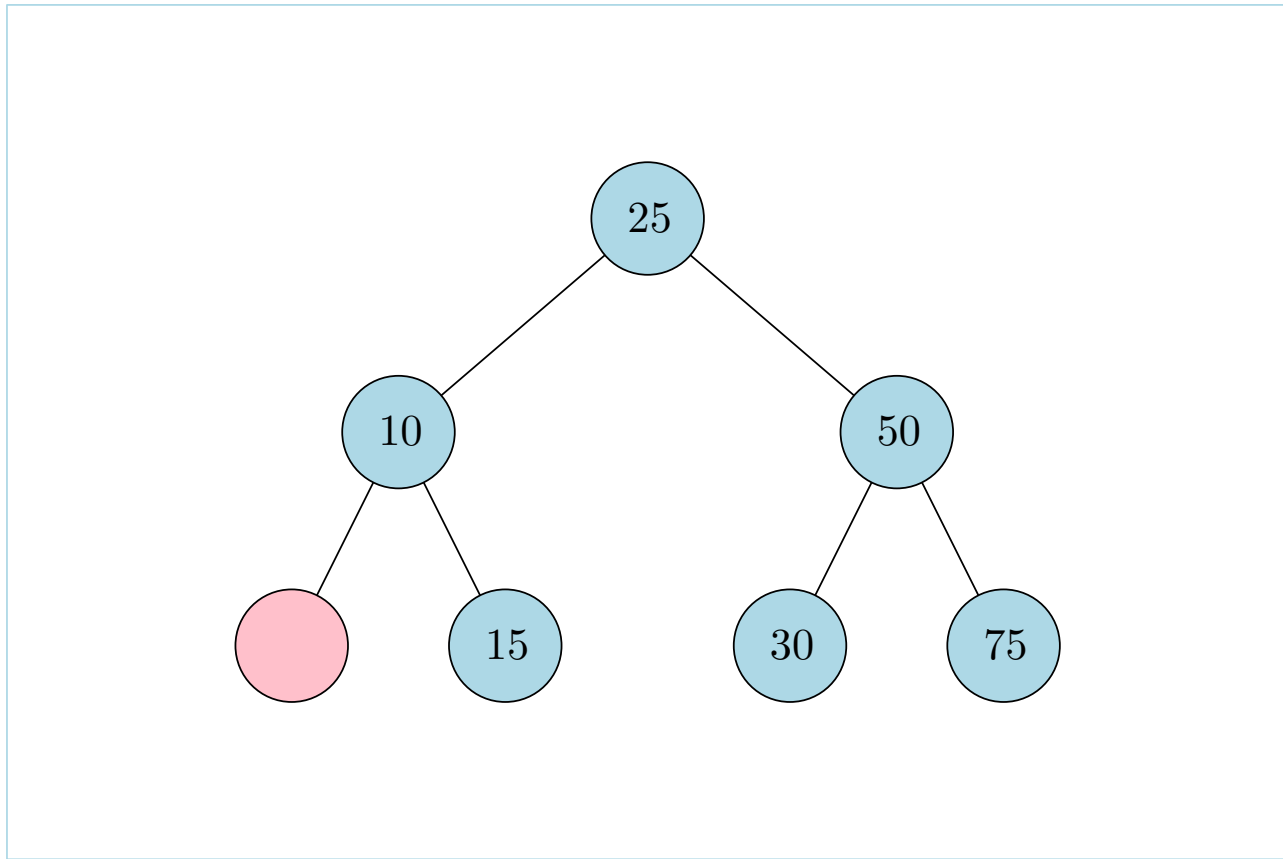
Παράδειγμα αλυσιδωτής διαγραφής μέχρι την ρίζα



Διαγραφή του κλειδιού 1

Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

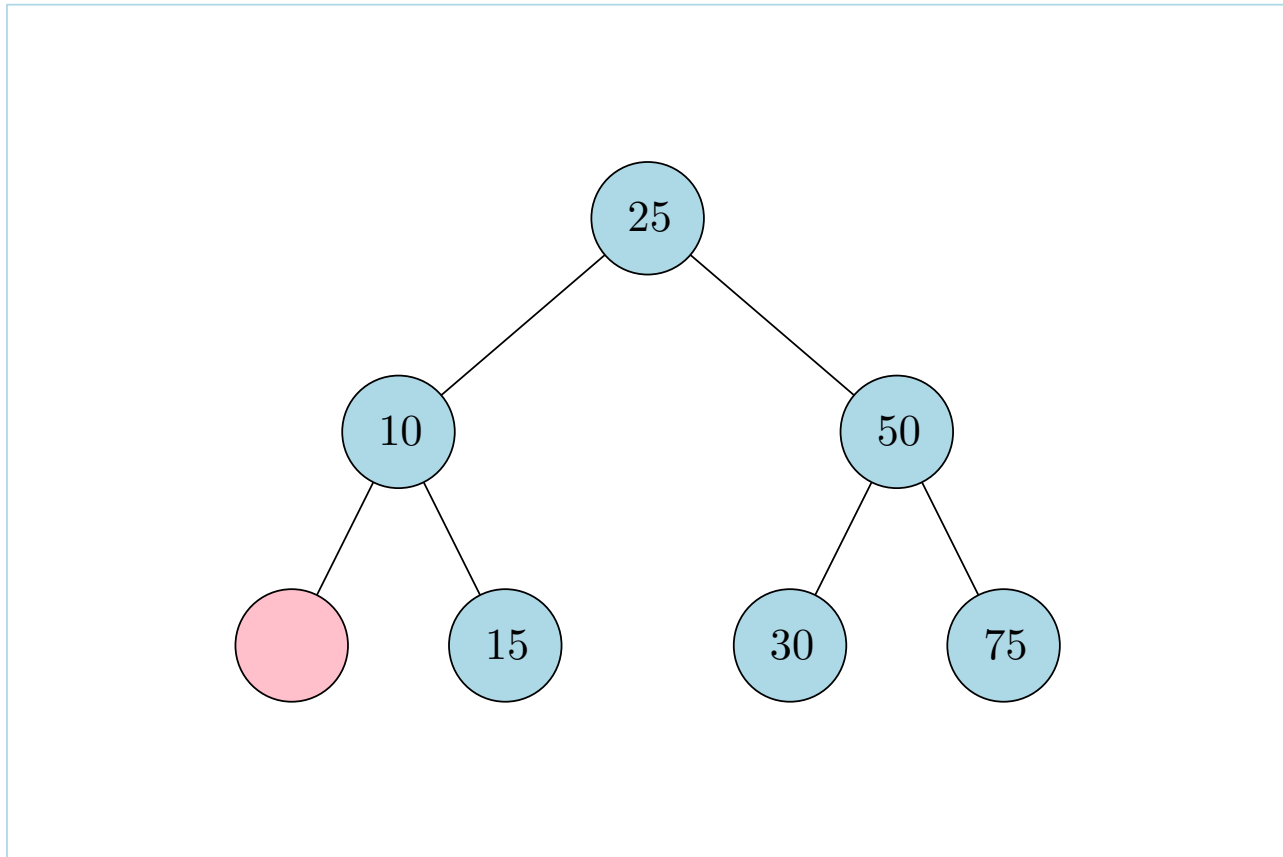
Παράδειγμα αλυσιδωτής διαγραφής μέχρι την ρίζα



Κόμβος με underflow - δοκιμάζουμε πρώτα balance και μετά fusion

Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

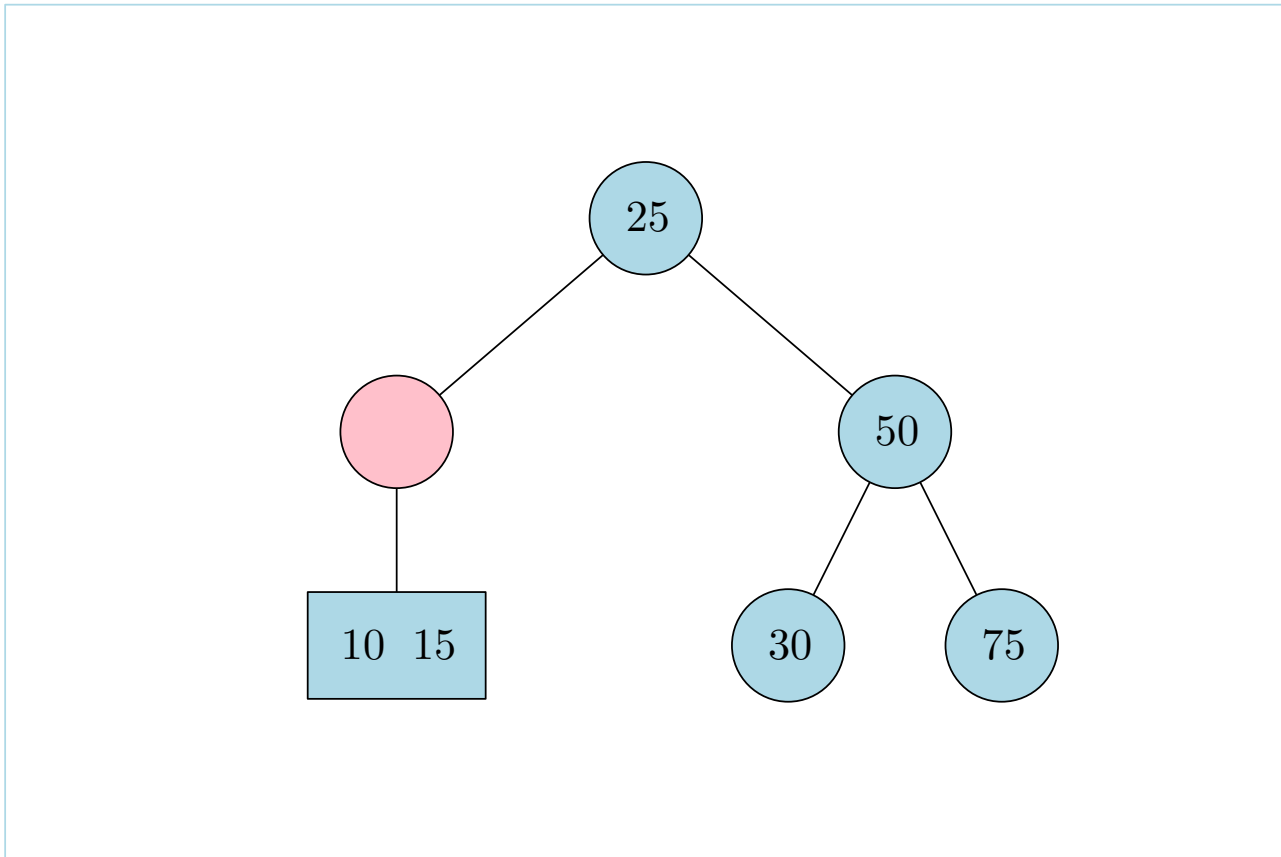
Παράδειγμα αλυσιδωτής διαγραφής μέχρι την ρίζα



Balance δεν γίνεται - οι γείτονες δεν έχουν επιπλέον κλειδιά

Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

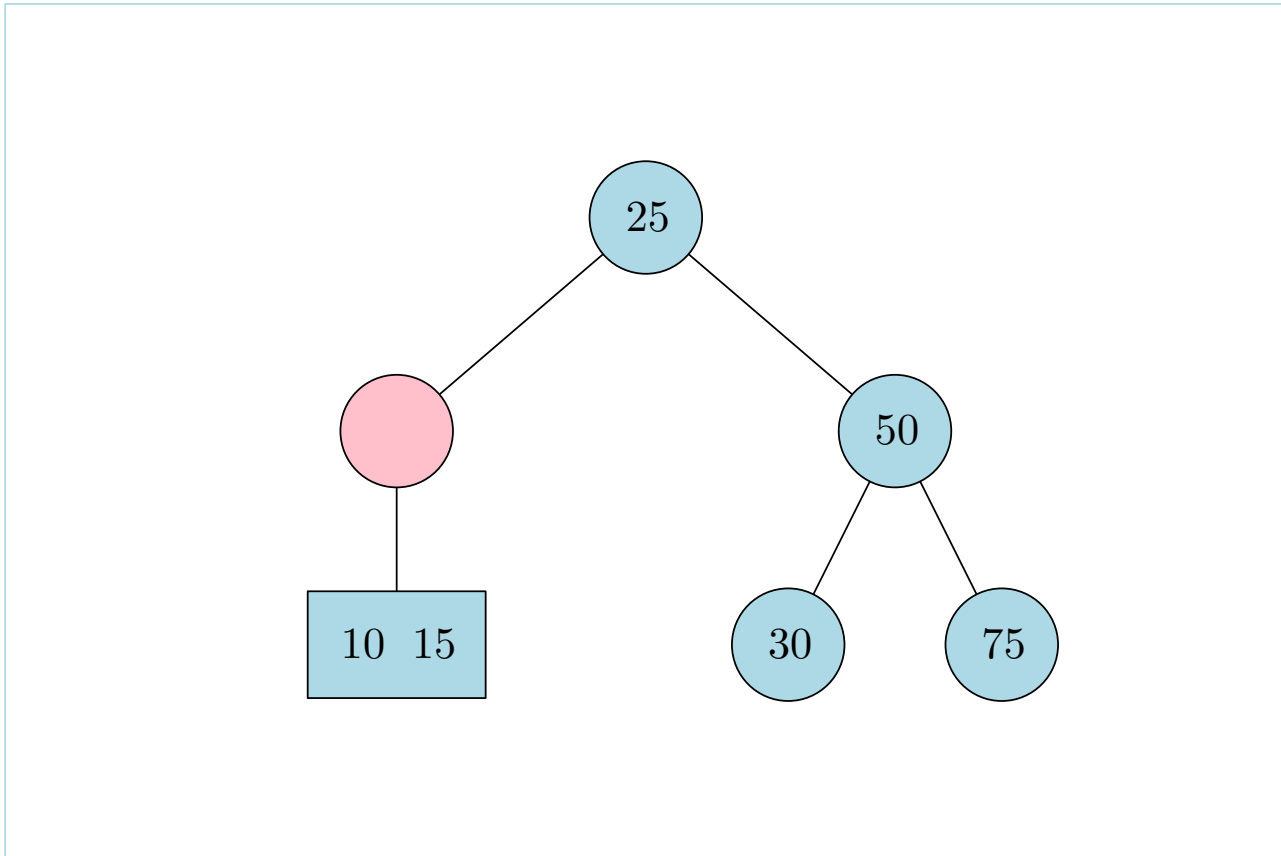
Παράδειγμα αλυσιδωτής διαγραφής μέχρι την ρίζα



Fusion - χρησιμοποιούμε το κλειδί του αδελφού και του πατέρα

Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

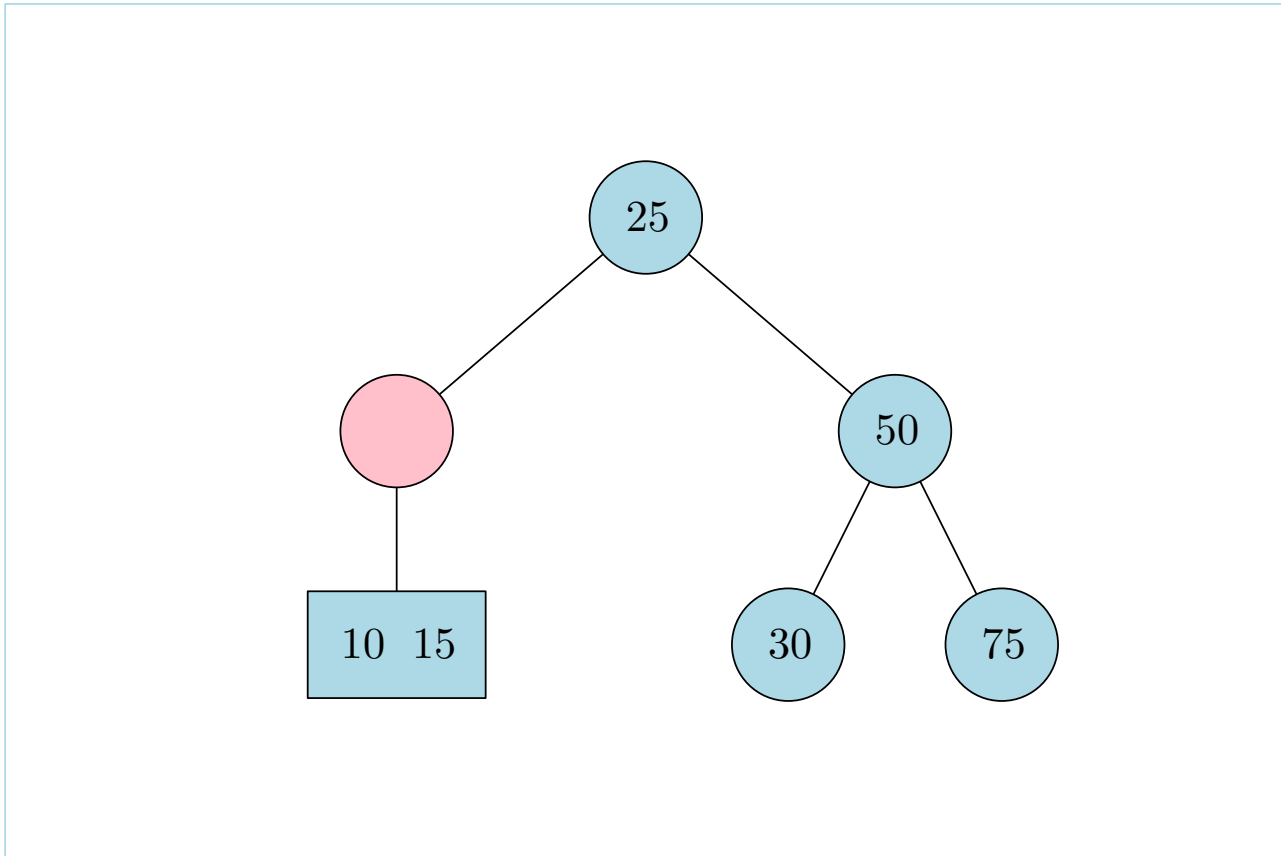
Παράδειγμα αλυσιδωτής διαγραφής μέχρι την ρίζα



Κόμβος με underflow - ένα επίπεδο επάνω

Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

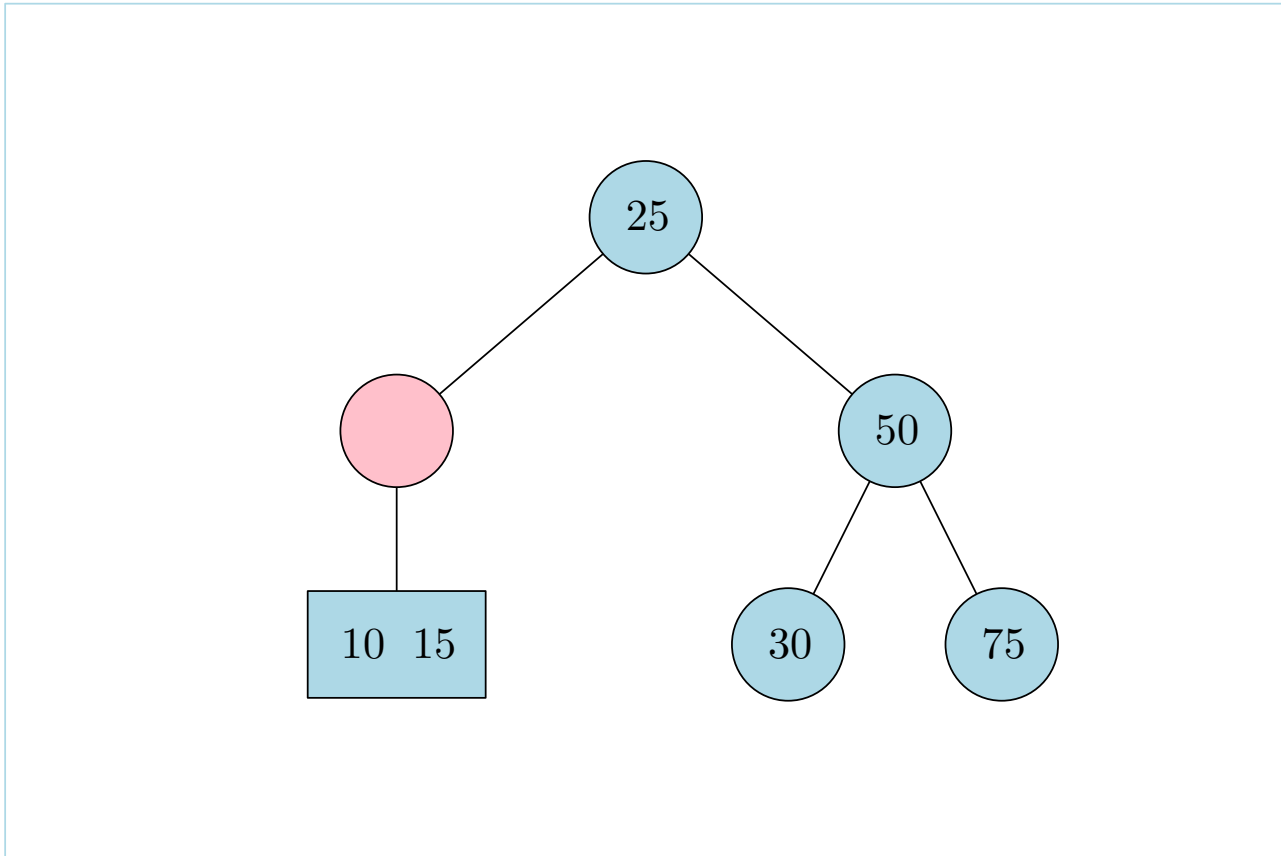
Παράδειγμα αλυσιδωτής διαγραφής μέχρι την ρίζα



Κόμβος με underflow - δοκιμάζουμε πρώτα balance και μετά fusion

Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

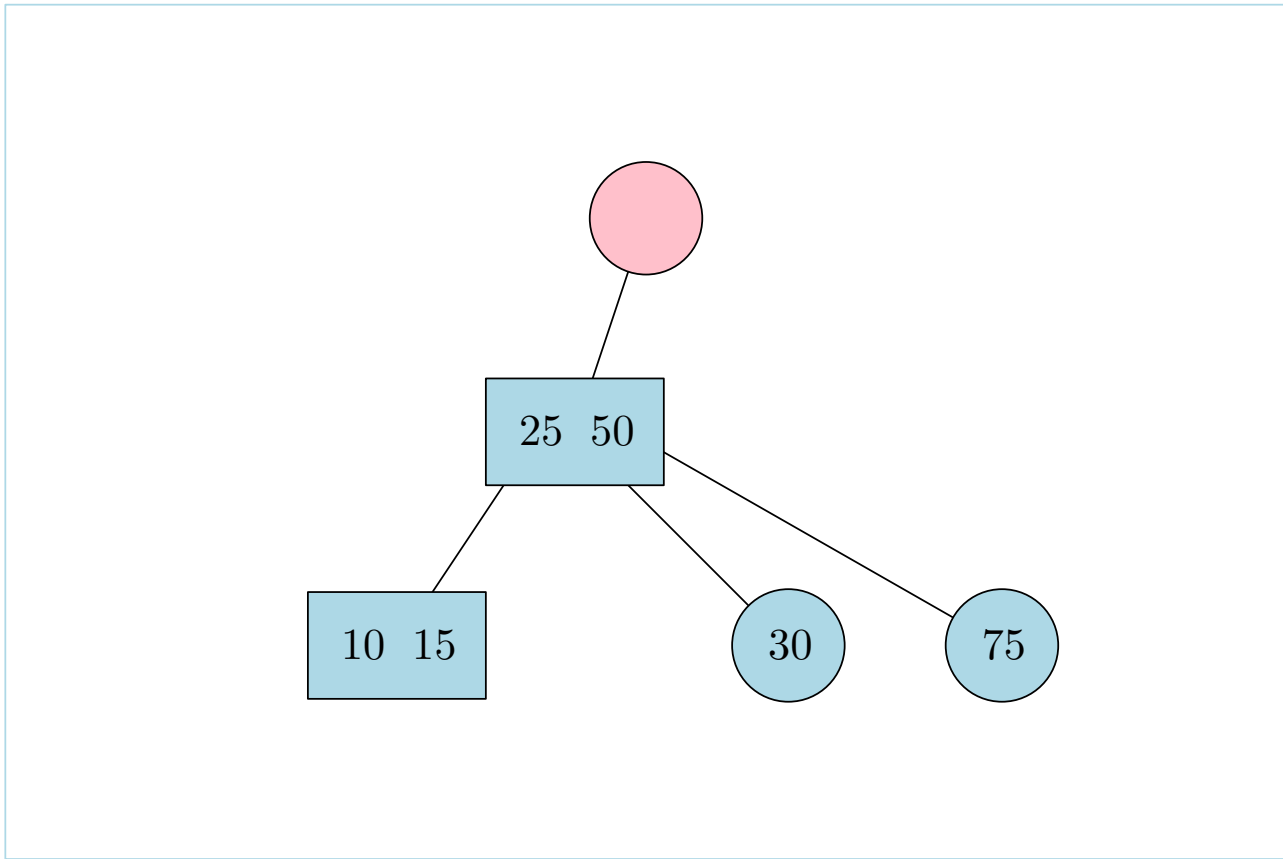
Παράδειγμα αλυσιδωτής διαγραφής μέχρι την ρίζα



Balance δεν γίνεται - οι γείτονες δεν έχουν επιπλέον κλειδιά

Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

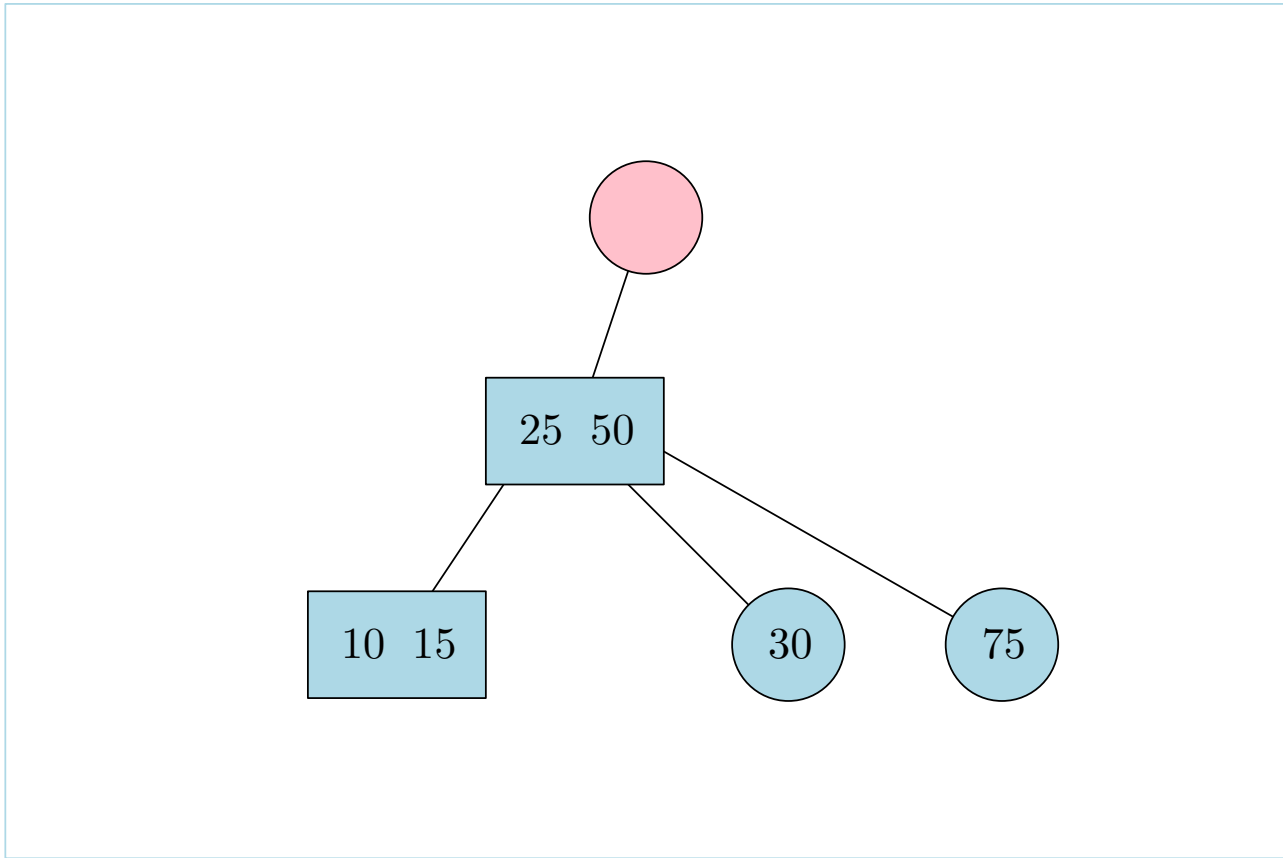
Παράδειγμα αλυσιδωτής διαγραφής μέχρι την ρίζα



Fusion - χρησιμοποιούμε το κλειδί του αδελφού και του πατέρα

Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

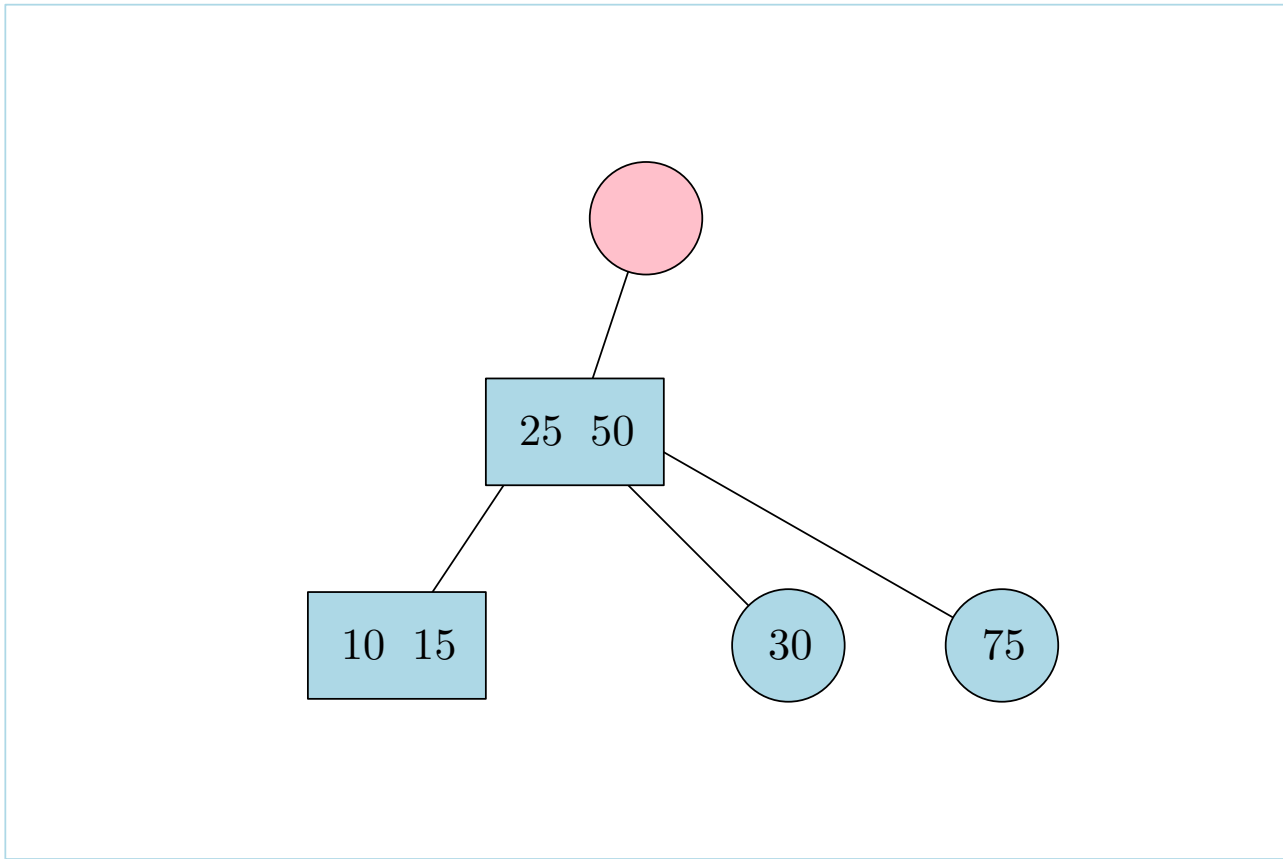
Παράδειγμα αλυσιδωτής διαγραφής μέχρι την ρίζα



Κόμβος με underflow - ένα επίπεδο επάνω

Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

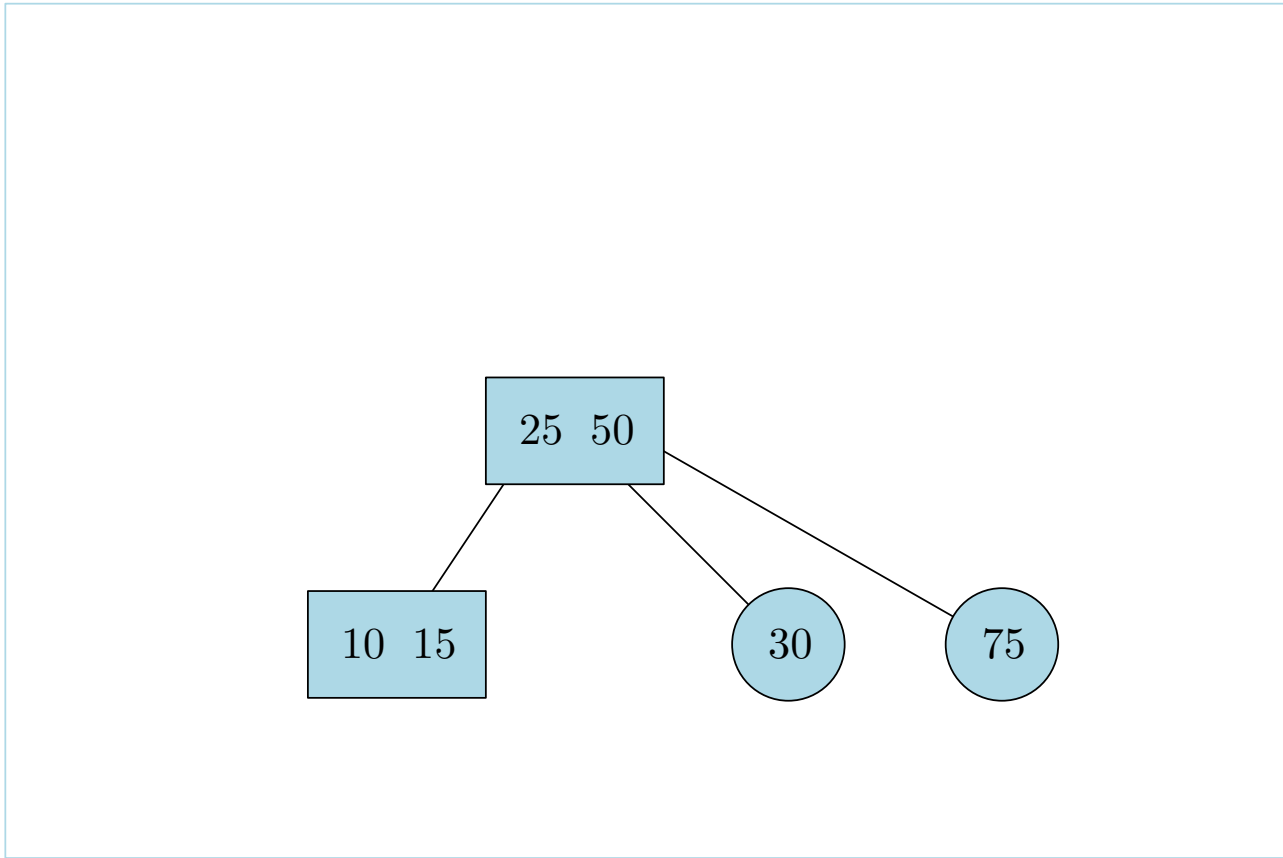
Παράδειγμα αλυσιδωτής διαγραφής μέχρι την ρίζα



Εαν η ρίζα έχει underflow - απλή διαγραφή

Διαγραφή σε Ισορροπημένο 2-3-4 Δέντρο

Παράδειγμα αλυσιδωτής διαγραφής μέχρι την ρίζα



Εαν η ρίζα έχει underflow - απλή διαγραφή

2-3-4 Δέντρα

θετικά:

- απλός αλγόριθμος εισαγωγής για διατήρηση ύψους
- πολυπλοκότητα $\mathcal{O}(\log n)$ για εισαγωγή, διαγραφή και αναζήτηση

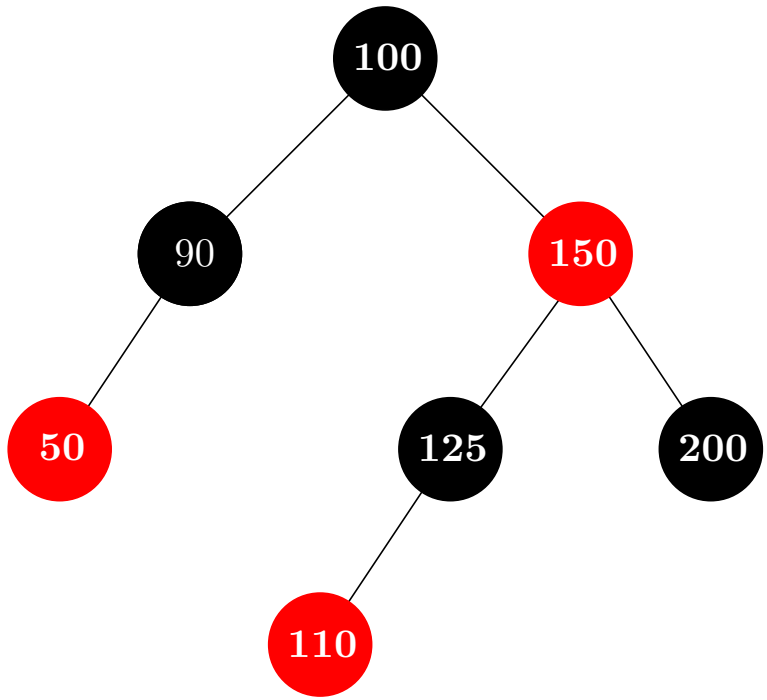
αρνητικά:

- 3 διαφορετικοί κόμβοι
- πολύπλοκες διαδικασίες λόγω πολλών συνδέσμων, αντιγραφών συνδέσμων, κ.τ.λ

Κόκκινα-Μαύρα Δέντρα

Ένα κόκκινο-μαύρο δέντρο είναι ένα δυαδικό δέντρο αναζήτησης (ΔΔΑ) με τις επιπλέον ιδιότητες:

- Κάθε κόμβος είναι ή κόκκινος ή μαύρος
- Η ρίζα είναι μαύρη
- Κανένας κόκκινος κόμβος δεν έχει κόκκινο παιδί
- Κάθε μονοπάτι από την ρίζα προς εξωτερικό κόμβο (null) περνάει από τον ίδιο αριθμό μαύρων κόμβων



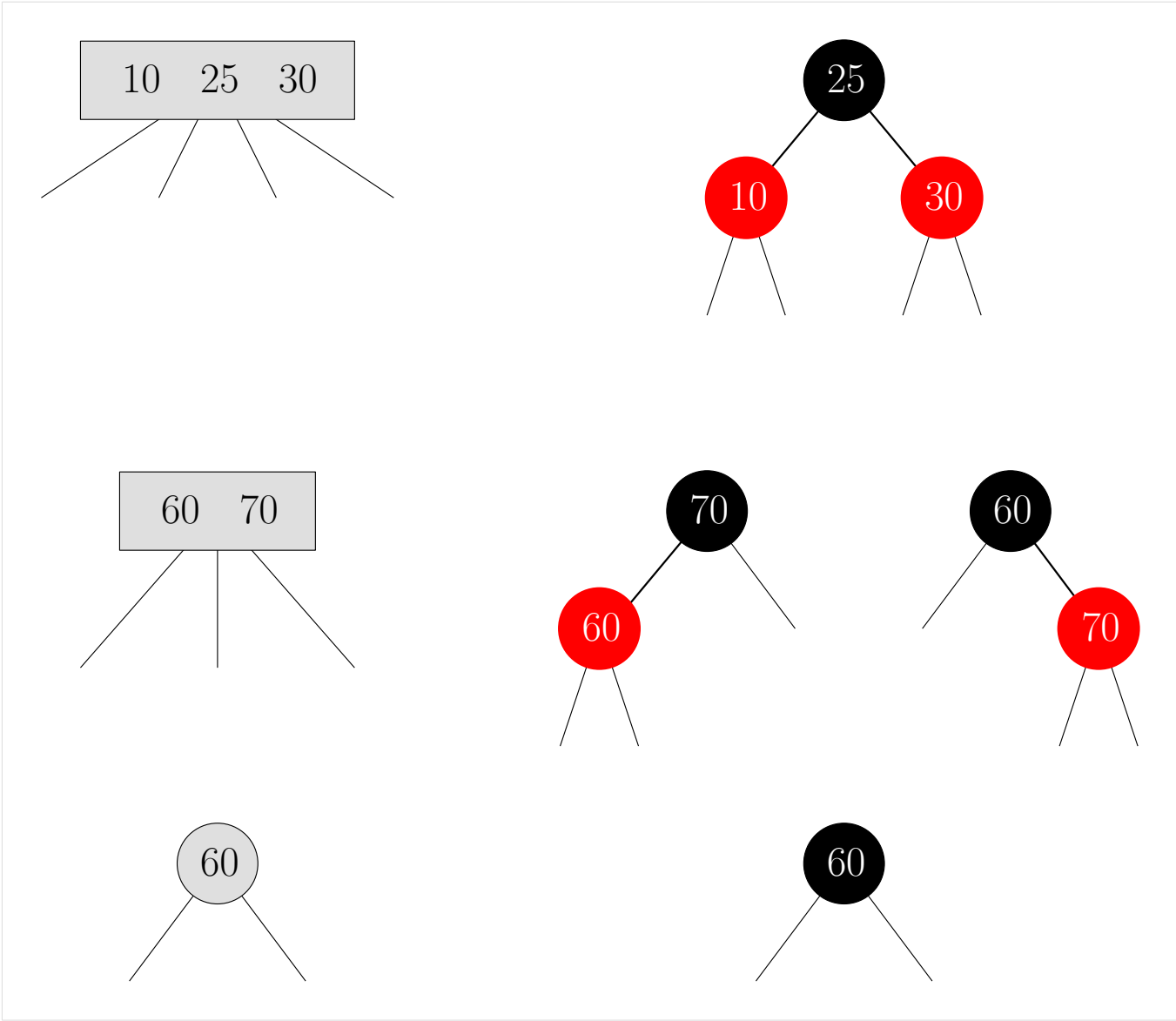
Κόκκινα-Μαύρα Δέντρα

Θεώρημα

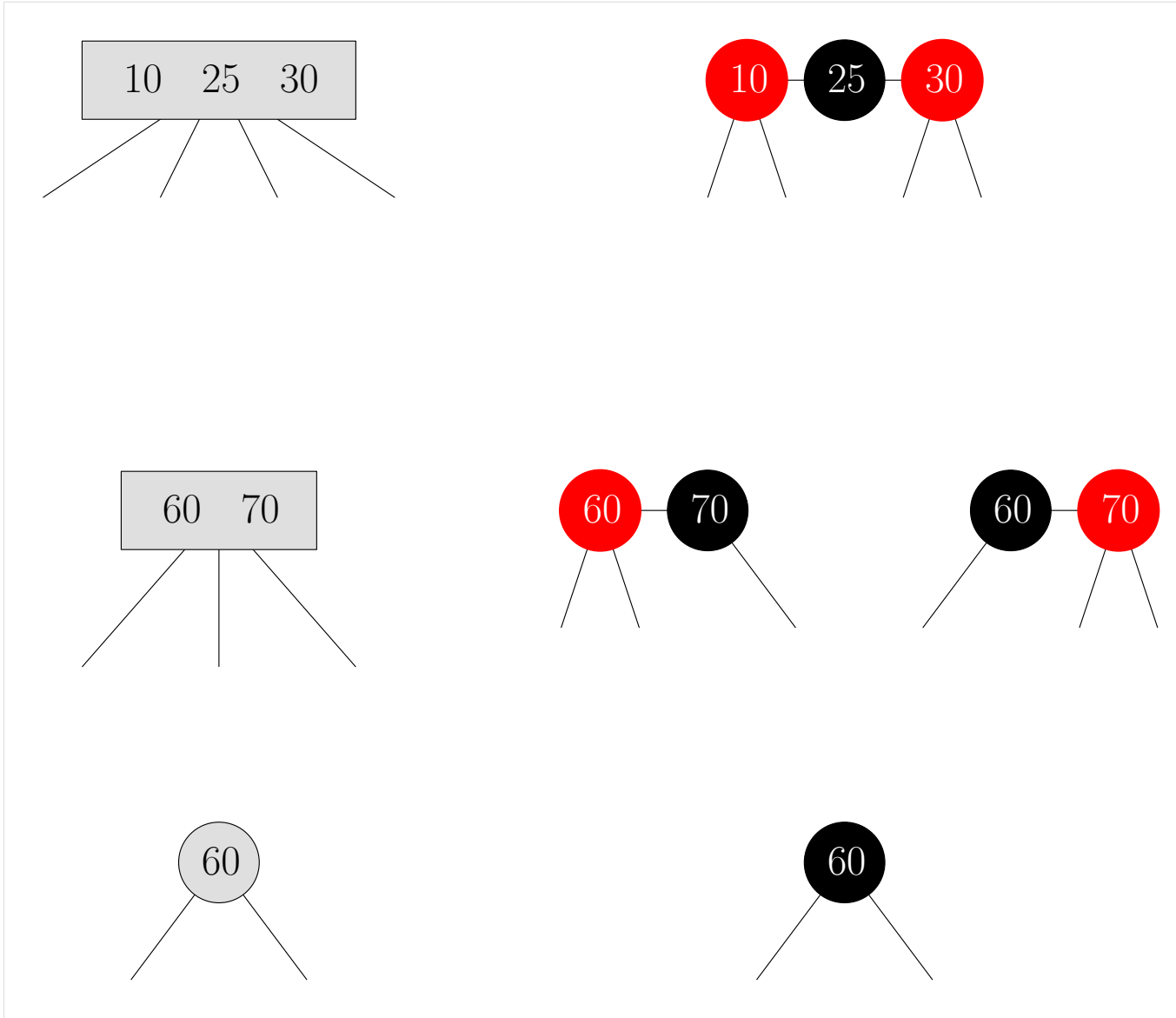
Ένα κόκκινο-μαύρο δέντρο με n κόμβους έχει ύψος $\mathcal{O}(\log n)$.

- Αντί για απόδειξη θα δούμε μία αντιστοίχιση των κόκκινων/μαύρων δέντρων με τα 2-3-4 δέντρα.
- Μπορούμε να δούμε τα κόκκινα-μαύρα δέντρα ως μία αναπαράσταση των 2-3-4 δέντρων.
- Αυτή η αντιστοίχιση μας επιτρέπει να δείξουμε εύκολα τις ιδιότητες των κόκκινων-μαύρων δέντρων.

Κόκκινα-Μαύρα Δέντρα



Κόκκινα-Μαύρα Δέντρα



Αναζήτηση

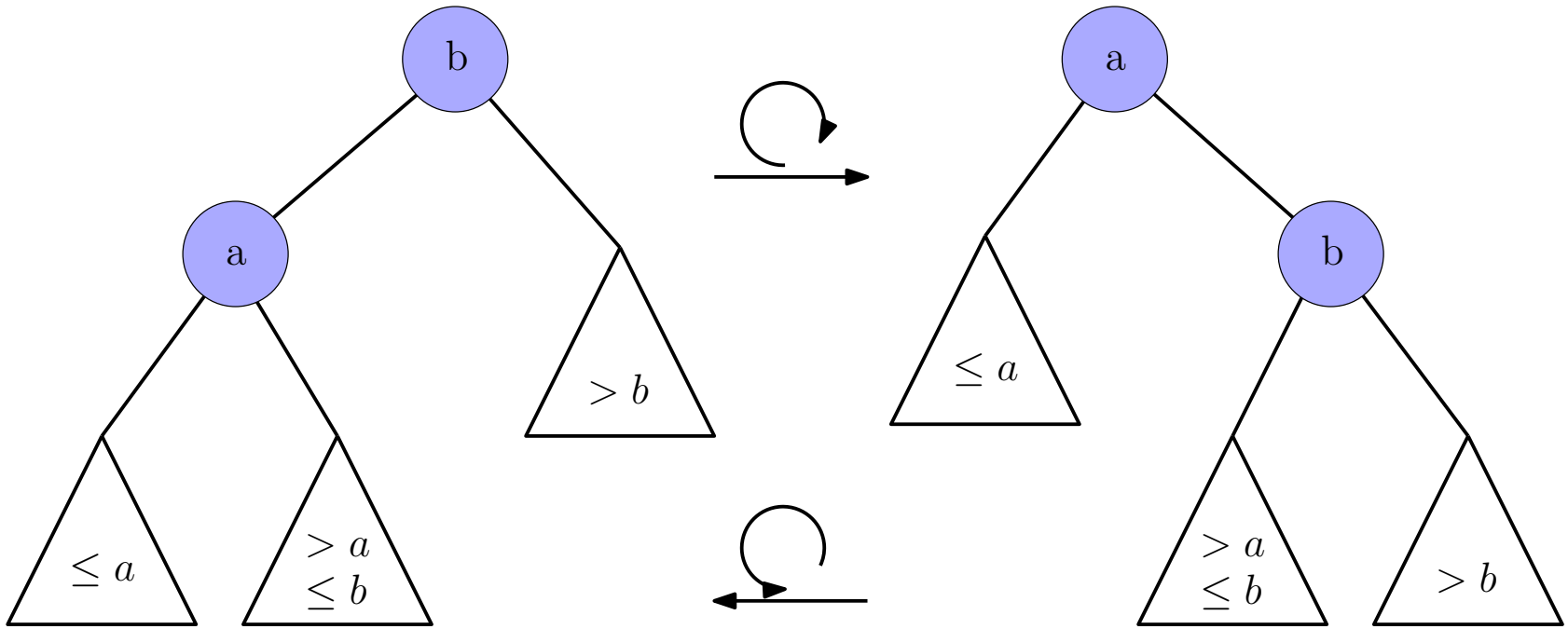
Τα κόκκινα-μαύρα δέντρα είναι ΔΔΑ !

Εισαγωγή και Διαγραφή

Αντιστοιχία με τα 2-3-4 δέντρα, απλά περιγράφουμε τις κινήσεις με αλλαγή χρωμάτων και περιστροφές.

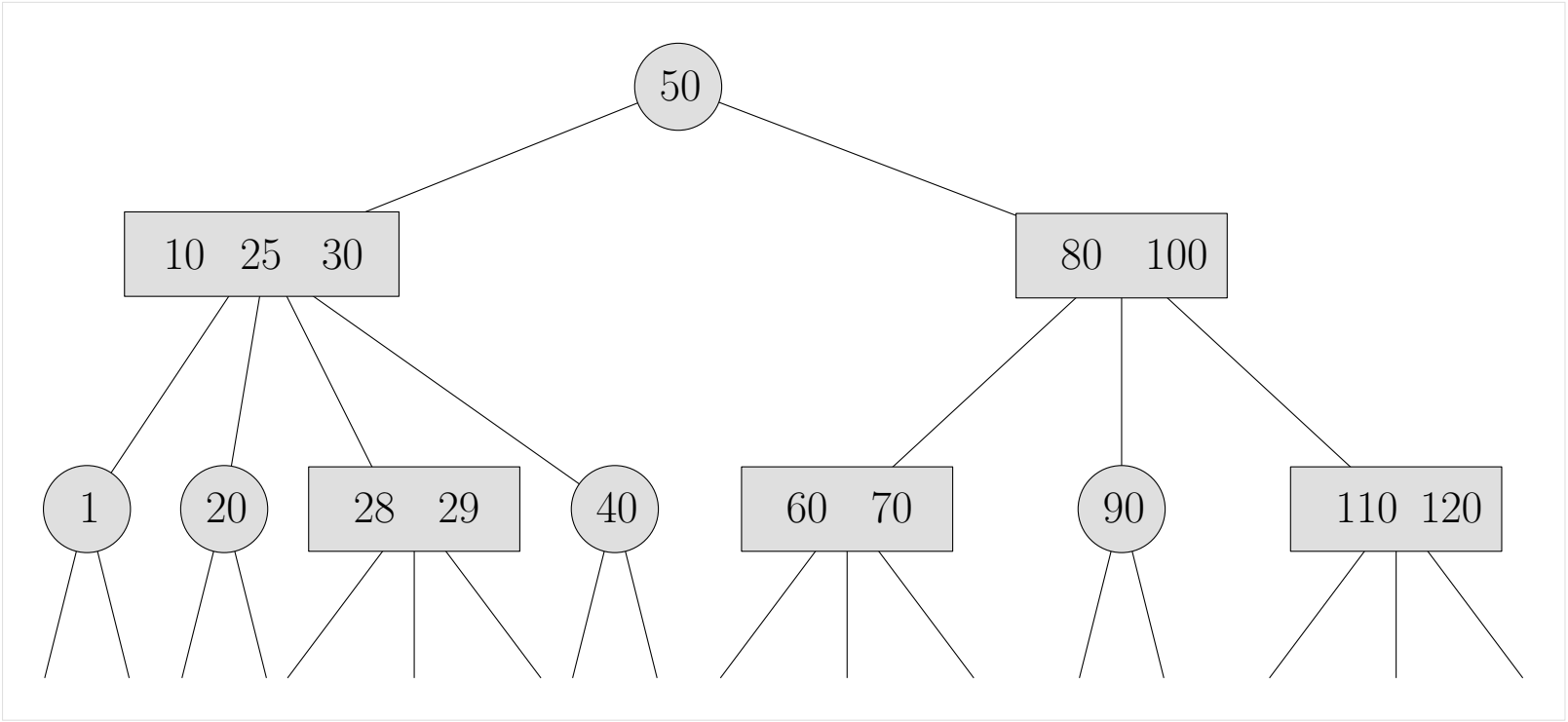
Κόκκινα-Μαύρα Δέντρα

Περιστροφές ΔΔΑ



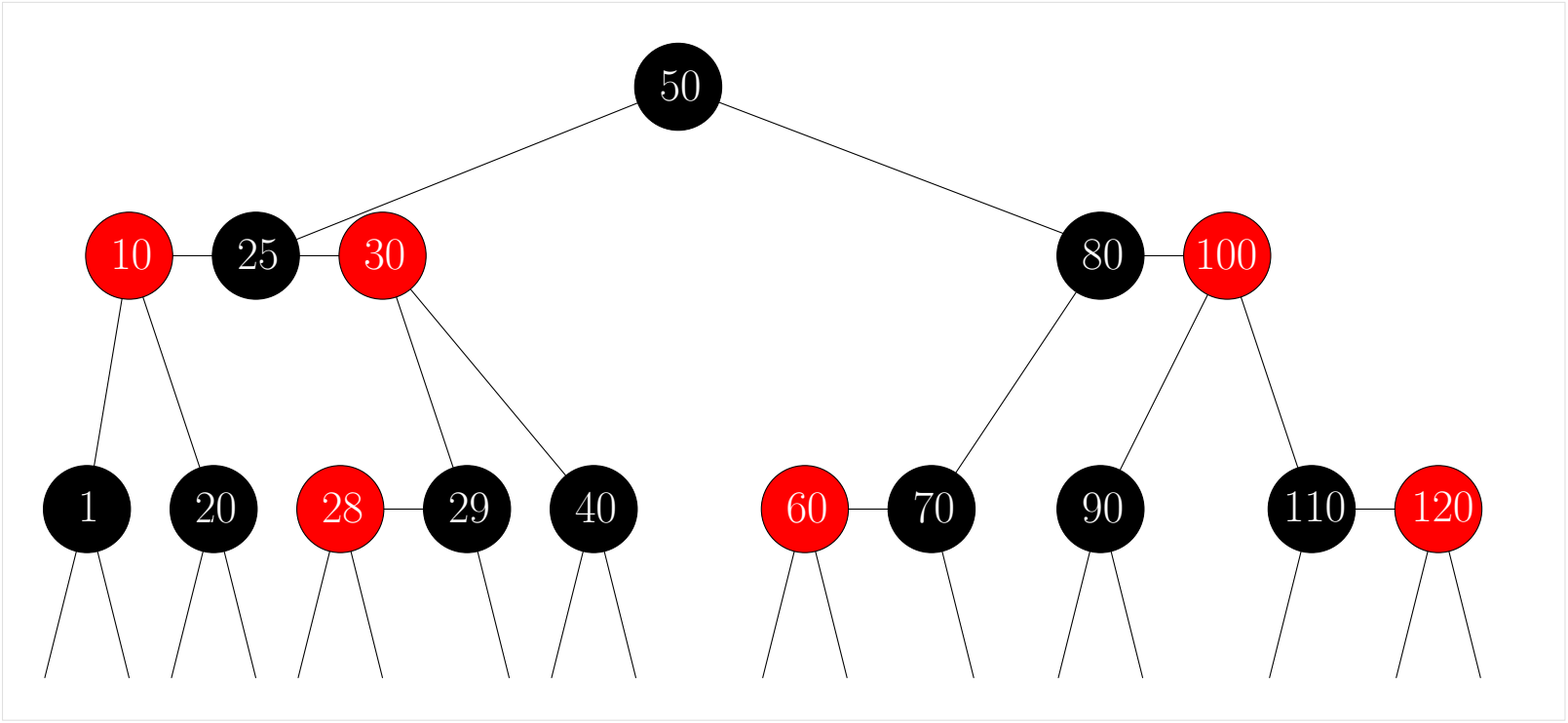
Κόκκινα-Μαύρα Δέντρα

Παράδειγμα



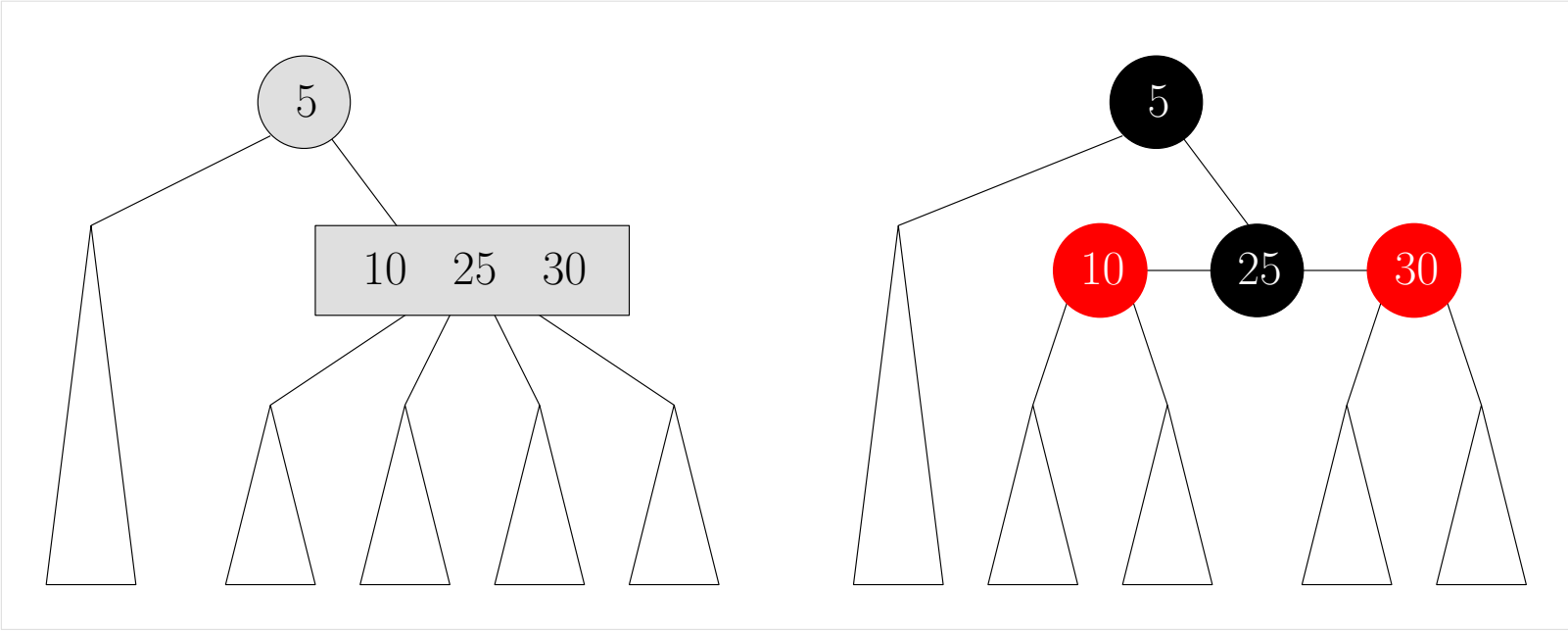
Κόκκινα-Μαύρα Δέντρα

Παράδειγμα



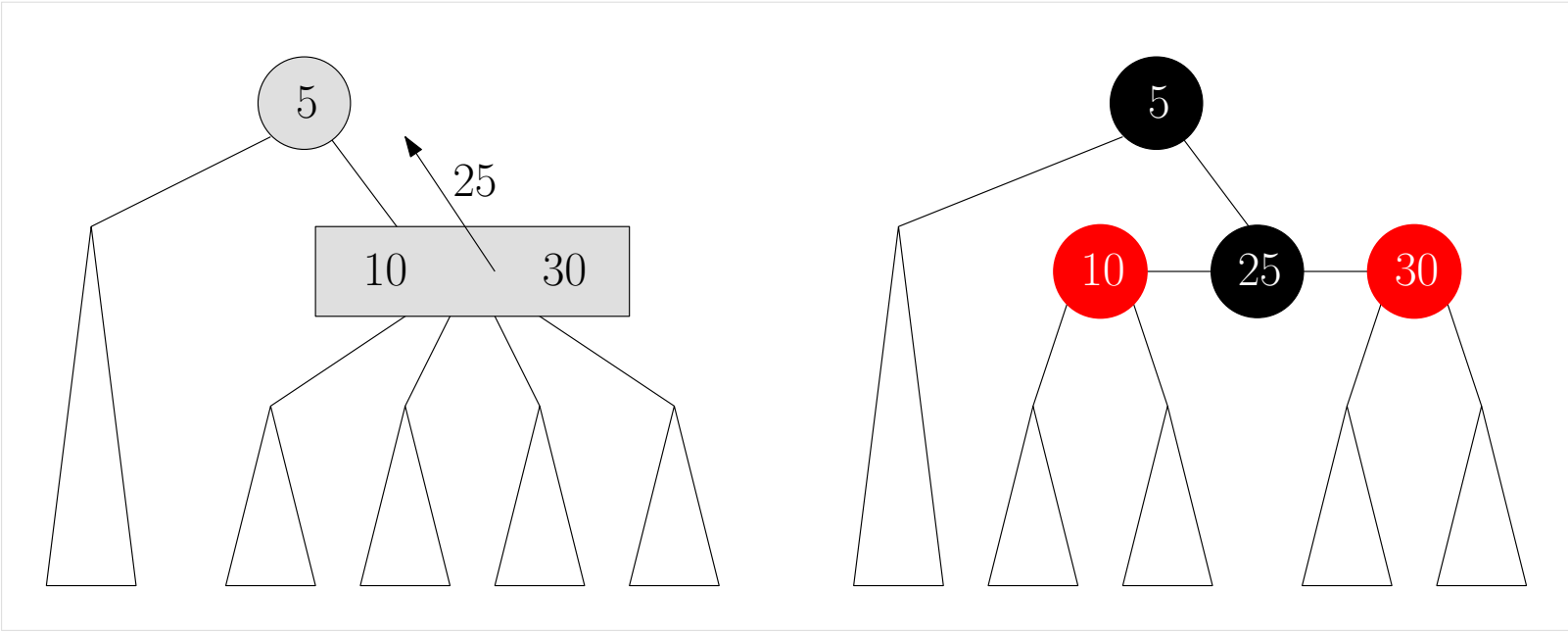
Κόκκινα-Μαύρα Δέντρα

Περιστροφές (πρώτη)



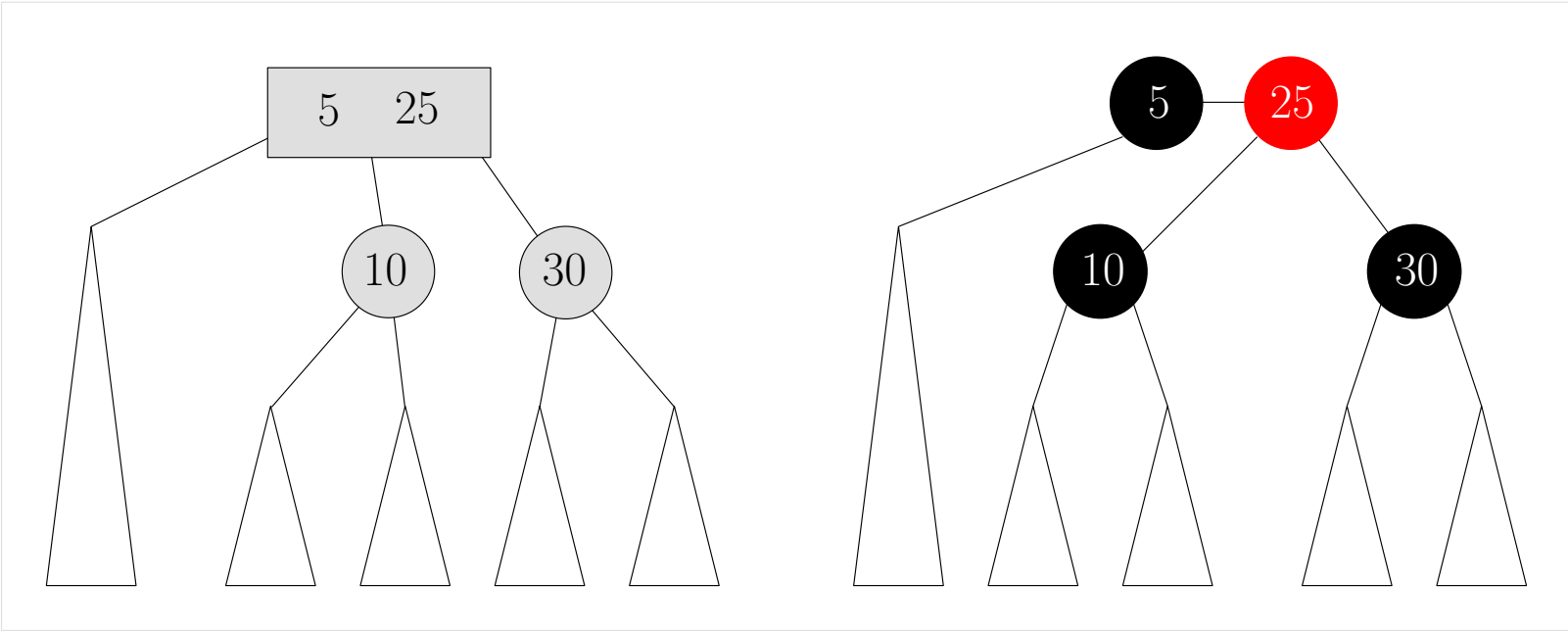
Κόκκινα-Μαύρα Δέντρα

Περιστροφές (πρώτη)



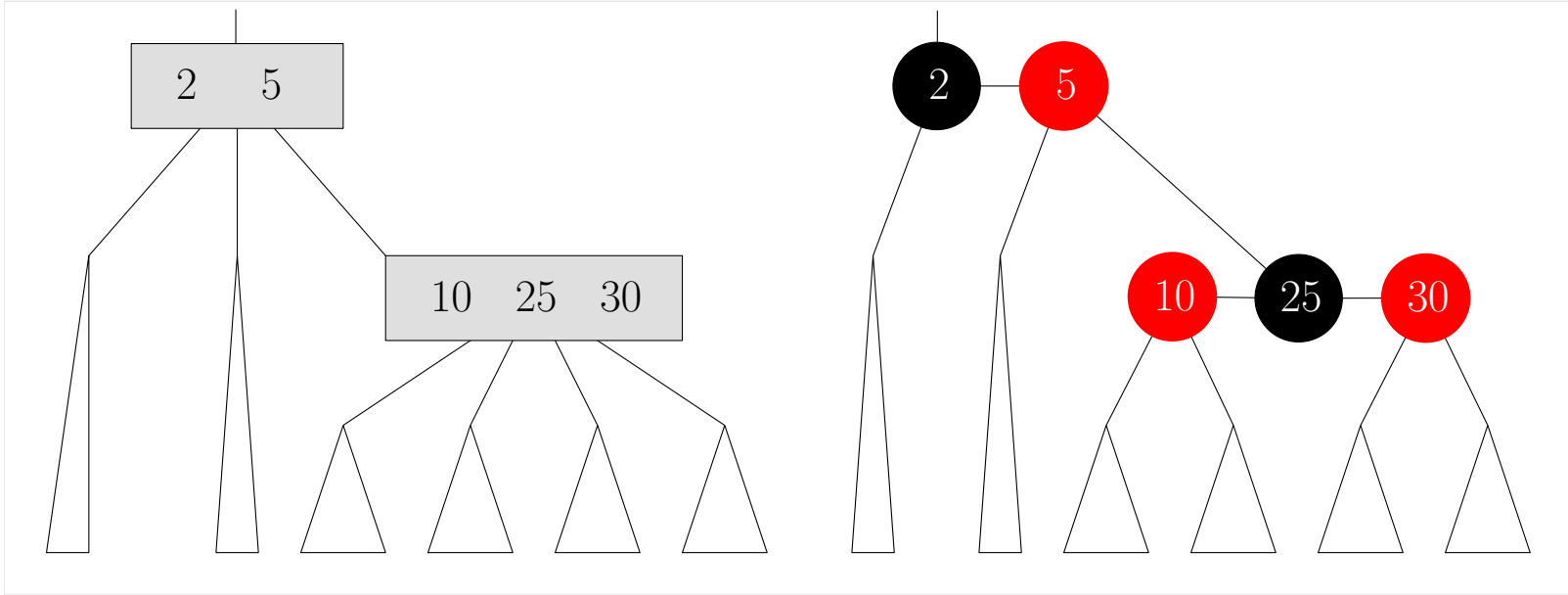
Κόκκινα-Μαύρα Δέντρα

Περιστροφές (πρώτη)



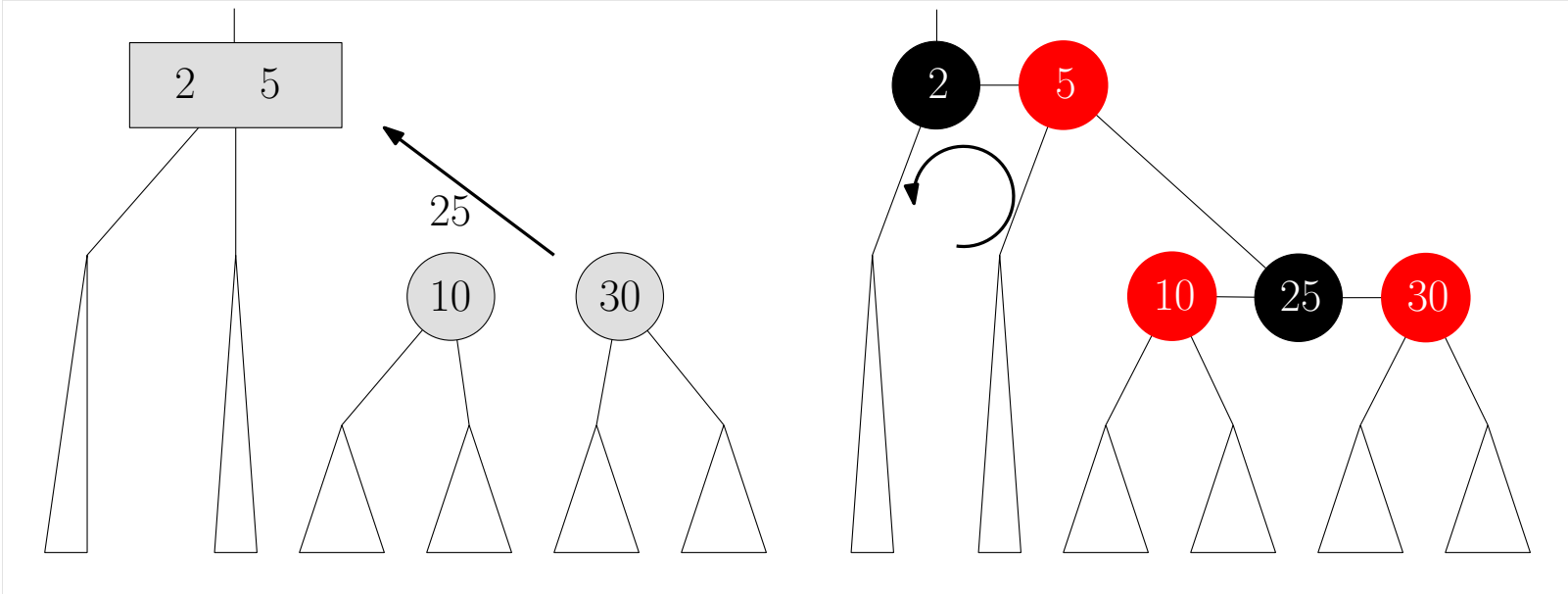
Κόκκινα-Μαύρα Δέντρα

Περιστροφές (δεύτερη)



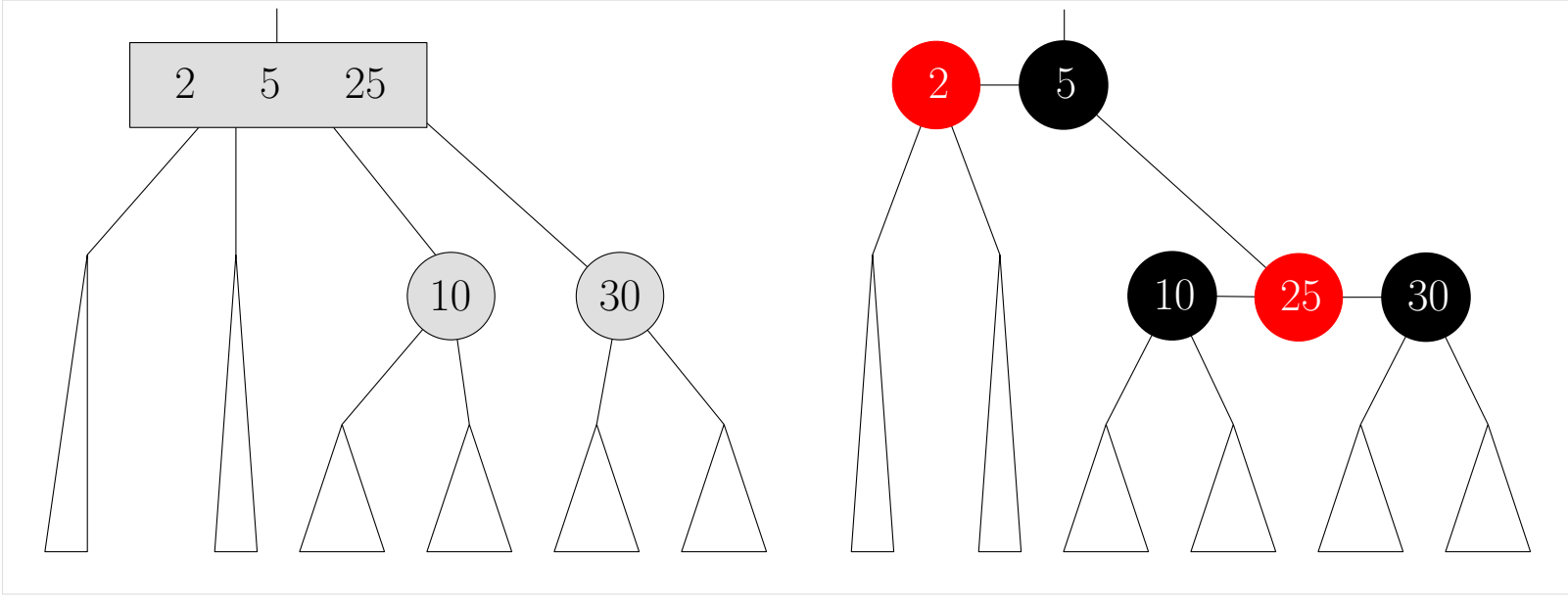
Κόκκινα-Μαύρα Δέντρα

Περιστροφές (δεύτερη)



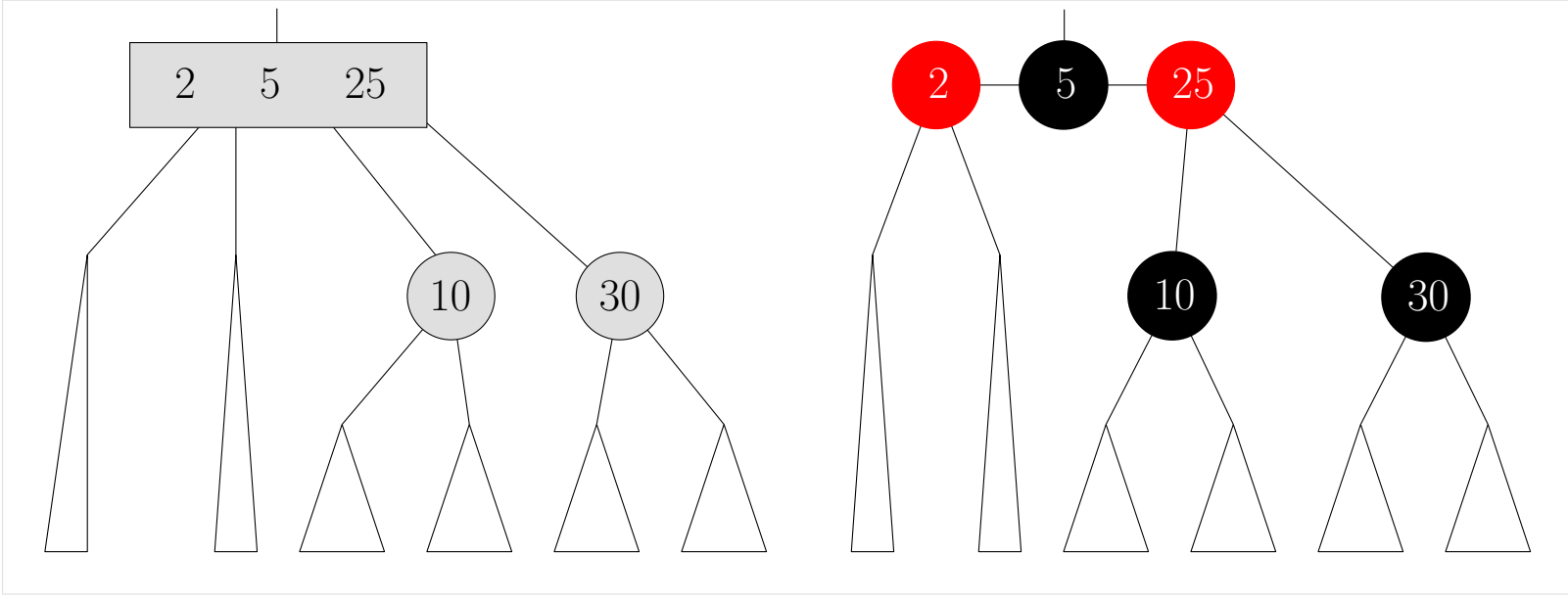
Κόκκινα-Μαύρα Δέντρα

Περιστροφές (δεύτερη)



Κόκκινα-Μαύρα Δέντρα

Περιστροφές (δεύτερη)



Υπάρχουν και άλλες περιπτώσεις...

Κόκκινα-Μαύρα Δέντρα

Χρόνοι Εισαγωγής και Διαγραφής

- Υπάρχουν αρκετοί πολύπλοκοι κανόνες! Όλοι όμως βγάζουν εύκολα νόημα εαν τους συνδέσουμε με το αντίστοιχο 2-3-4 δέντρο.
- Η προσομοίωση των αλλαγών ενός κόμβου σε ένα 2-3-4 δέντρο με τους αντίστοιχους κόμβους ενός κόκκινου-μαύρου δέντρου παίρνει $\mathcal{O}(1)$ χρόνο.
- Αφού τα 2-3-4 δέντρα υποστηρίζουν εισαγωγή σε $\mathcal{O}(\log n)$ χρόνο, το ίδιο ισχύει και για τα κόκκινα-μαύρα δέντρα.
- Η ίδια ιδέα λειτουργεί και για τις διαγραφές.

Θεώρημα

Ένα κόκκινο-μαύρο δέντρο με n κόμβους έχει ύψος $\mathcal{O}(\log n)$.

Απόδειξη

Μπορούμε να μαζέψουμε όλους τους κόκκινους κόμβους στους γονείς τους, ώστε να μετατρέψουμε το δέντρο σε 2-3-4.

Το ύψος του δέντρου το πολύ να υποδιπλασιαστεί.

Το 2-3-4 δέντρο που προκύπτει έχει ύψος $\mathcal{O}(\log n)$, και άρα το αρχικό κόκκινο/μαύρο δέντρο έχει ύψος $2 \cdot \mathcal{O}(\log n) = \mathcal{O}(\log n)$.

□

Άλλα Ισορροπημένα Δέντρα

- AVL δέντρα: $\mathcal{O}(\log n)$ ύψος χρησιμοποιώντας περιστροφές
- splay δέντρα
- scapegoat δέντρα

AVL Δέντρα

Το πρώτο ισοζυγισμένο δέντρο!

G.M. **A**delson-**V**elskii and E.M. **L**andis. An algorithm for the organization of information, Proceedings of the USSR Academy of Sciences 146: 263–266, 1962.

Ονομάστηκε από τα αρχικά των συγγραφέων.

AVL Δέντρα

- Δυαδικό δέντρο αναζήτησης
- Επιπλέον συνθήκη ισορροπίας

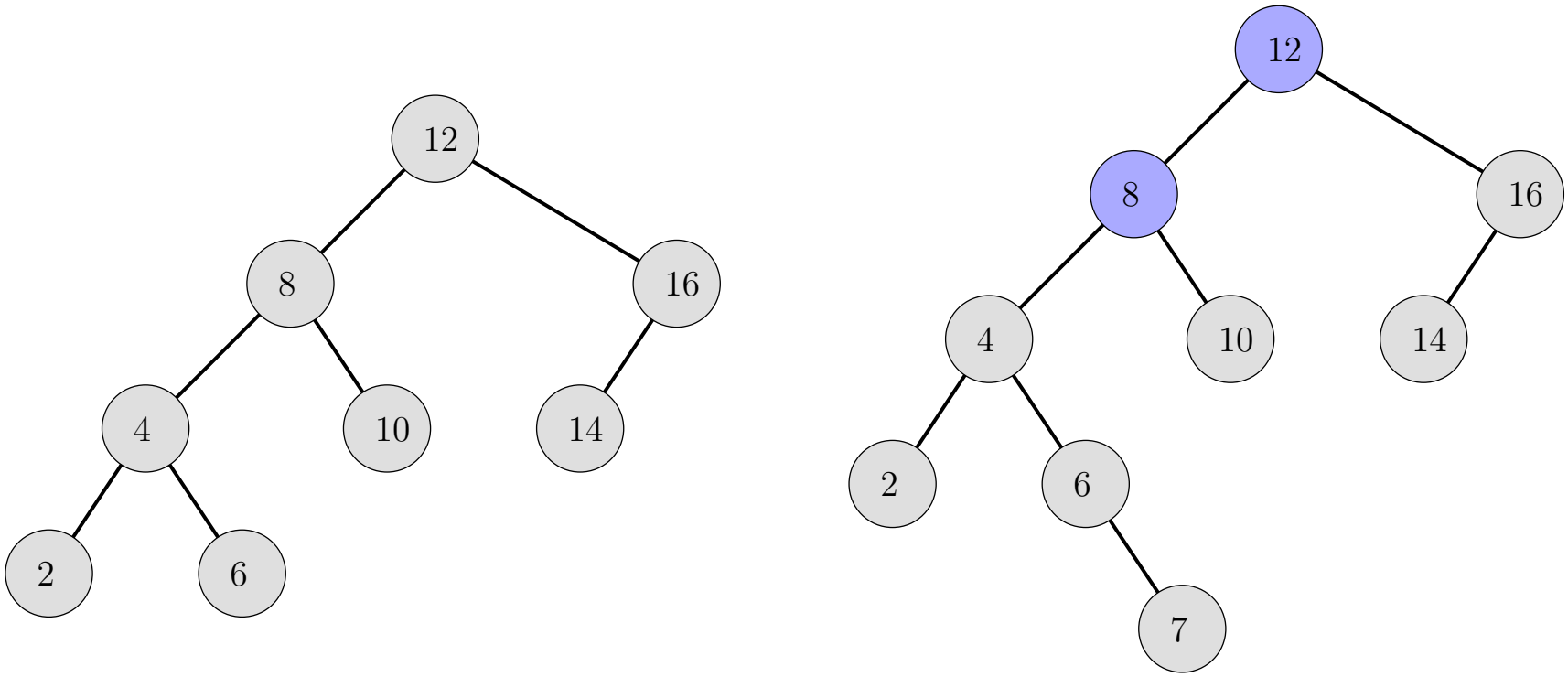
Η επιπλέον συνθήκη ισορροπίας πρέπει να διατηρείται εύκολα και να διατηρεί το ύψος του δέντρου σε $\mathcal{O}(\log n)$.

AVL Δέντρα

Ορισμός

Ένα δέντρο AVL είναι ένα δυαδικό δέντρο αναζήτησης T όπου για κάθε κόμβο $v \in T$ τα ύψη των υποδέντρων του διαφέρουν το πολύ κατά 1. Το ύψος του κενού υποδέντρου ορίζεται σε -1.

AVL Δέντρα



Αριστερά AVL δέντρο, δεξιά μη-AVL δέντρο.

AVL Δέντρα

Ύψος

Θεώρημα

Ένα AVL δέντρο με n κλειδιά έχεις ύψος $\mathcal{O}(\log n)$.

Θεώρημα

Ένα AVL δέντρο με n κλειδιά έχεις ύψος $\mathcal{O}(\log n)$.

Απόδειξη

Έστω $n(h)$ ο ελάχιστος αριθμός κόμβων ενός AVL δέντρου με ύψος h.

□

Θεώρημα

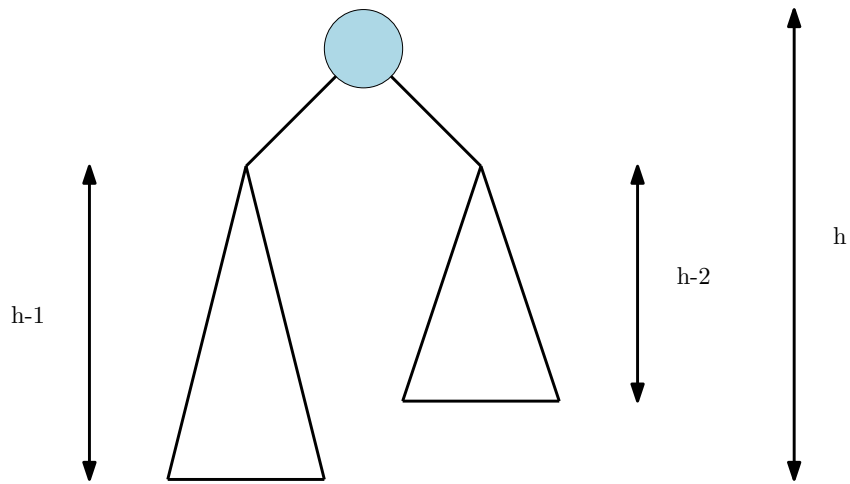
Ένα AVL δέντρο με n κλειδιά έχεις ύψος $\mathcal{O}(\log n)$.

Απόδειξη

Έστω $n(h)$ ο ελάχιστος αριθμός κόμβων ενός AVL δέντρου με ύψος h.

Εύκολα φαίνεται πως $n(1) = 1$ και $n(2) = 2$.

Για $h > 2$ ένα AVL δέντρο έχει την ρίζα και δύο υποδέντρα, ένα AVL υποδέντρο με ύψος $h - 1$ και ένα AVL υποδέντρο με ύψος $h - 2$.



Άρα $n(h) = 1 + n(h - 1) + n(h - 2)$.

□

Θεώρημα

Ένα AVL δέντρο με n κλειδιά έχεις ύψος $\mathcal{O}(\log n)$.

Απόδειξη

Έστω $n(h)$ ο ελάχιστος αριθμός κόμβων ενός AVL δέντρου με ύψος h.

Επειδή $n(h-1) > n(h-2)$ έχουμε πως

$$n(h) = 1 + n(h-1) + n(h-2) > 2n(h-2) > 4n(h-4) > 8n(h-6) > \dots$$

και μέσω αναγωγής

$$n(h) > 2^i n(h-2i)$$

Επειδή $n(1) = 1$ πέρνουμε πως $n(h) > 2^{(h-1)/2}$ και λογαριθμίζοντας πέρνουμε $h < 2\log n(h) + 1$.

□

Εισαγωγή σε AVL δέντρο

Η εισαγωγή γίνεται κανονικά όπως και στα ΔΔΑ αλλά στην συνέχεια μπορεί να χρειαστεί να φτιάξουμε το δέντρο.

Θα φροντίσουμε να κάνουμε $\mathcal{O}(d)$ πράξεις όπου d είναι το βάθος όπου γίνεται η εισαγωγή.

Εισαγωγή σε AVL δέντρο

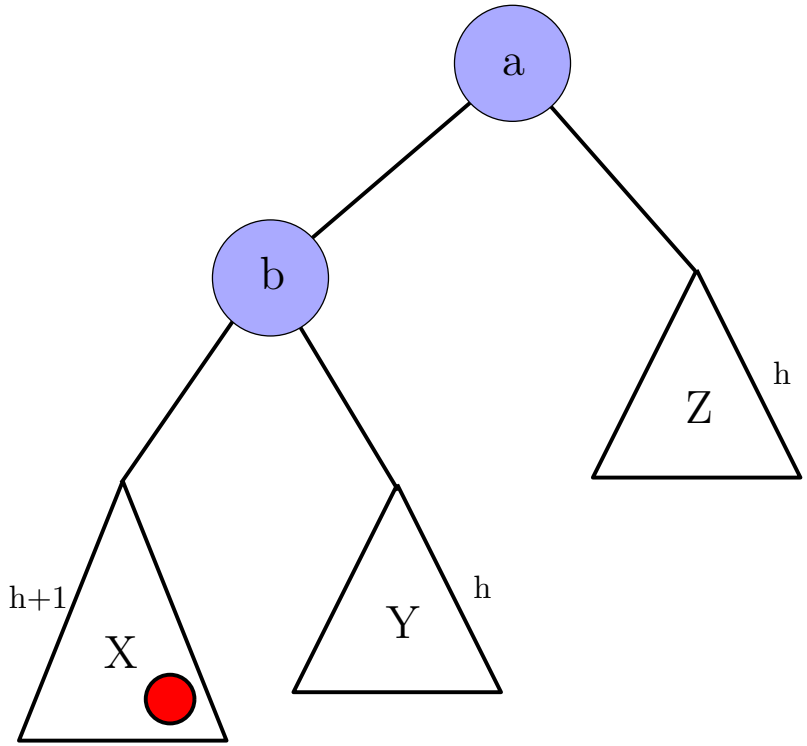
Θα φτιάξουμε το δέντρο με περιστροφές. Έστω x ο κόμβος με το μεγαλύτερο βάθος που είναι μη-ισοζυγισμένος.

Υπάρχουν 4 περιπτώσεις ανάλογα με το αν η εισαγωγή είναι στο

1. αριστερό υποδέντρο του αριστερού παιδιού του x
2. δεξί υποδέντρο του αριστερού παιδιού του x
3. αριστερό υποδέντρο του δεξιού παιδιού του x
4. δεξί υποδέντρο του δεξιού παιδιού του x

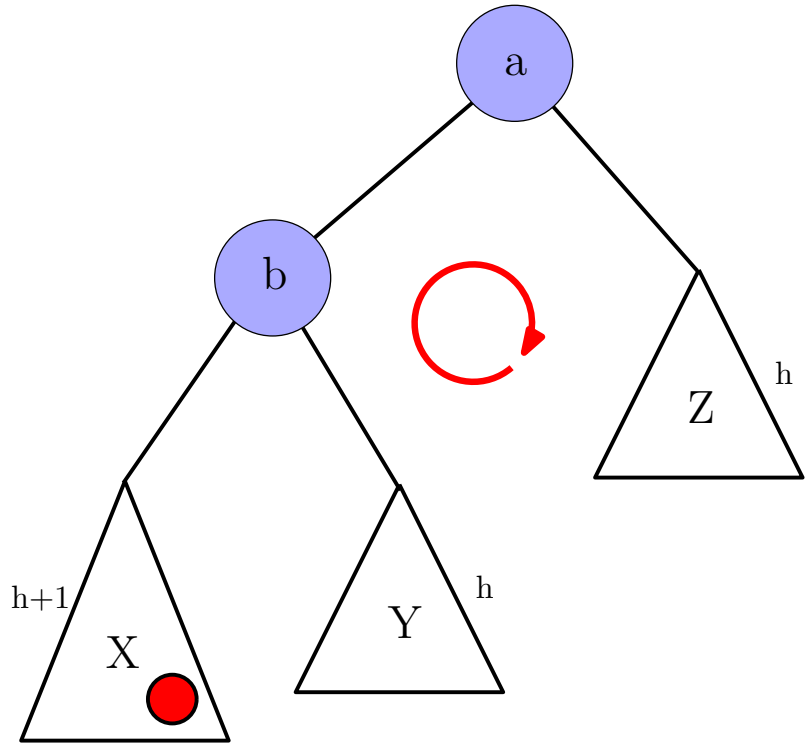
Οι περιπτώσεις $1\&4$ λύνονται με *μονή περιστροφή* ενώ οι περιπτώσεις $2\&3$ με *διπλή περιστροφή*.

Μονή Περιστροφή



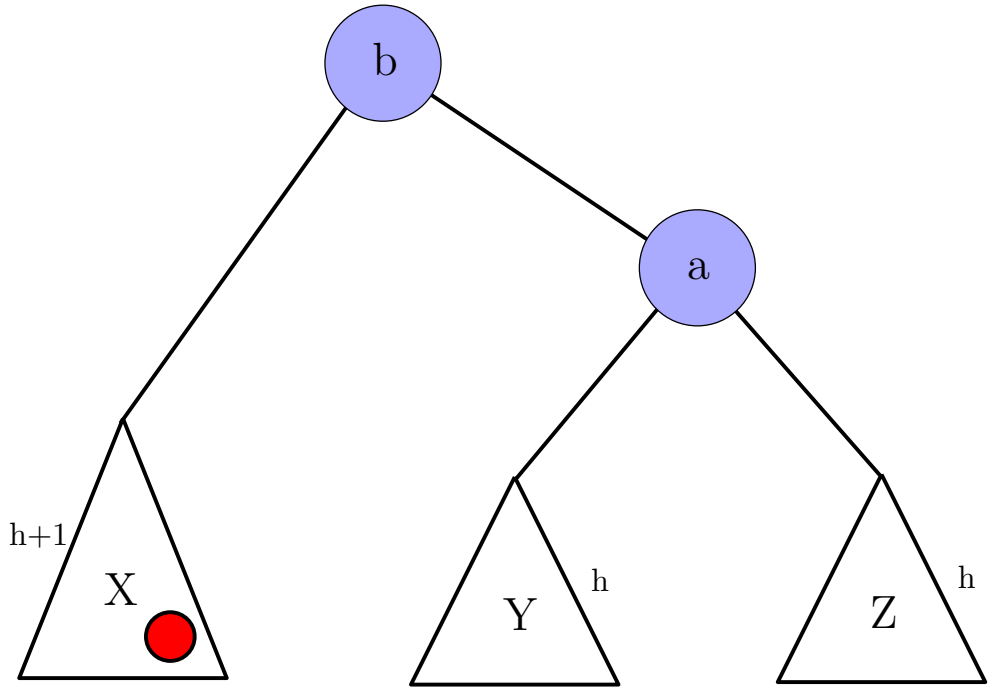
$$X < b < Y < a < Z$$

Μονή Περιστροφή



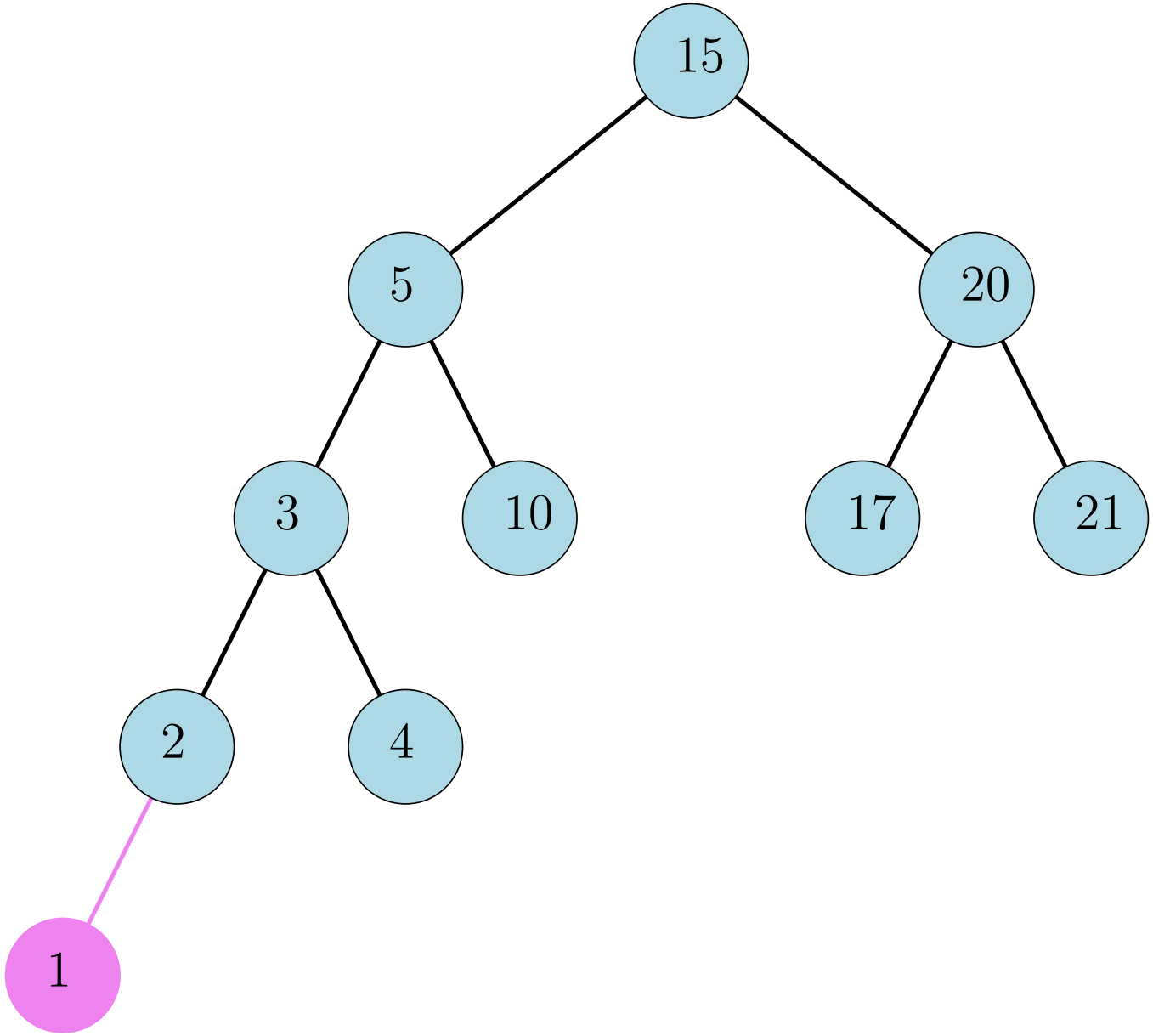
$$X < b < Y < a < Z$$

Μονή Περιστροφή

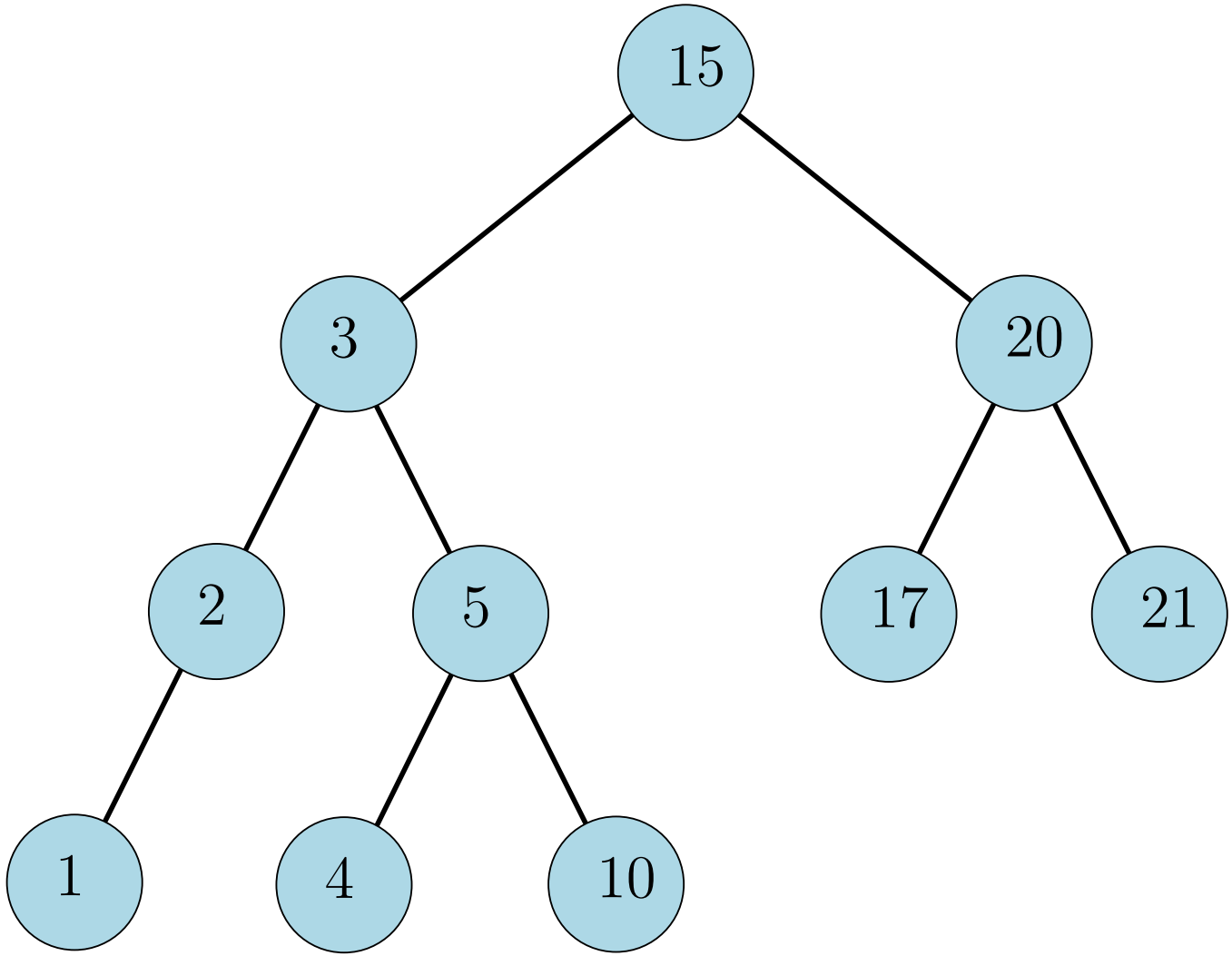


$$X < b < Y < a < Z$$

Μονή Περιστροφή

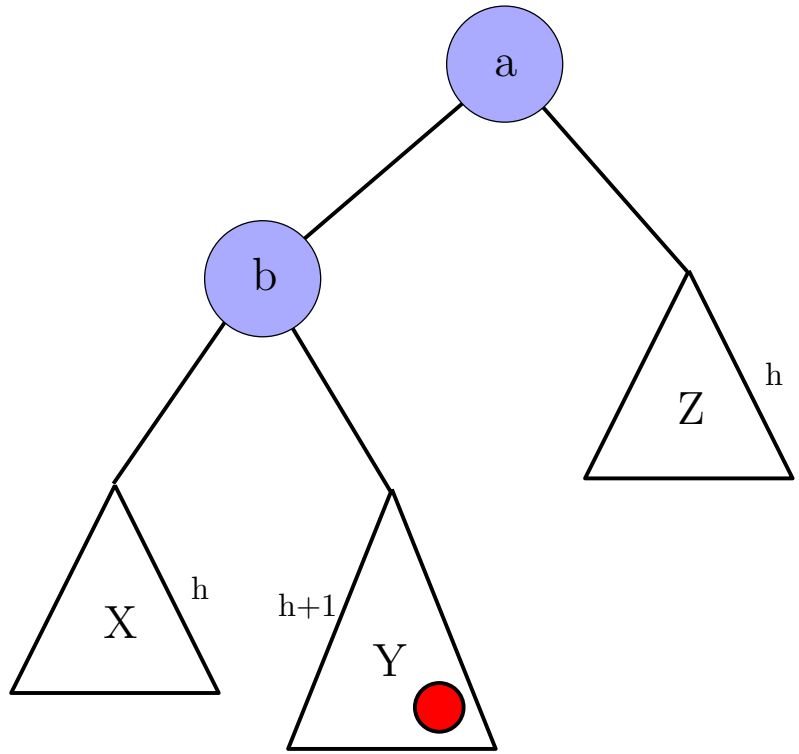


Μονή Περιστροφή



Άλλες περιπτώσεις

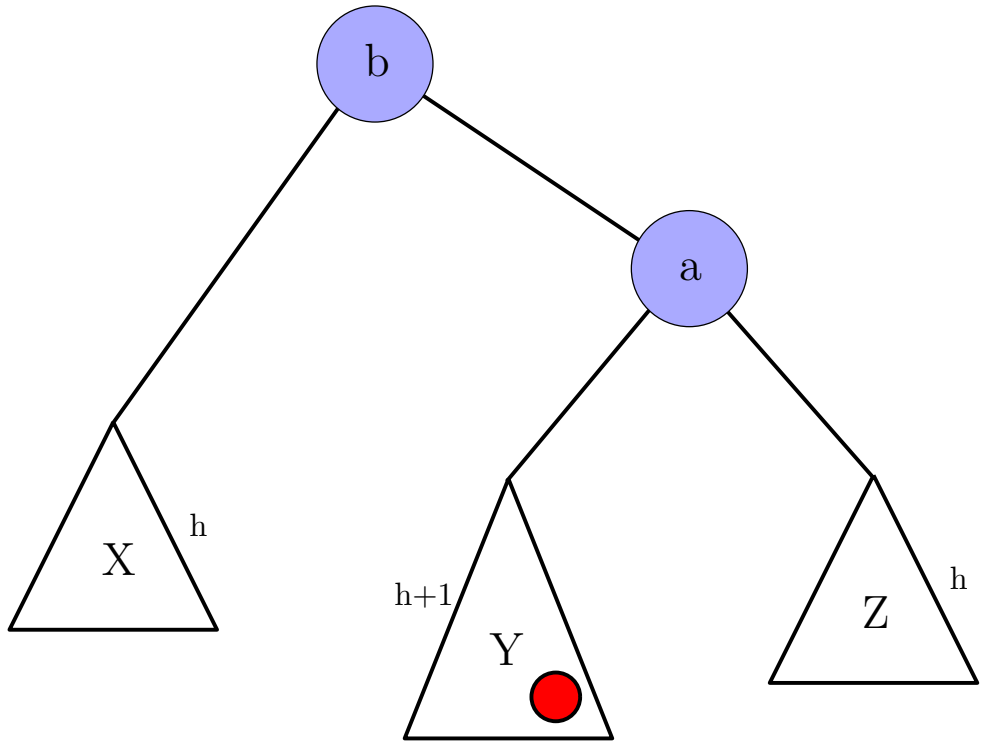
Στις περιπτώσεις $2\&3$ δεν λειτουργεί η μονή περιστροφή.



$$X < b < Y < a < Z$$

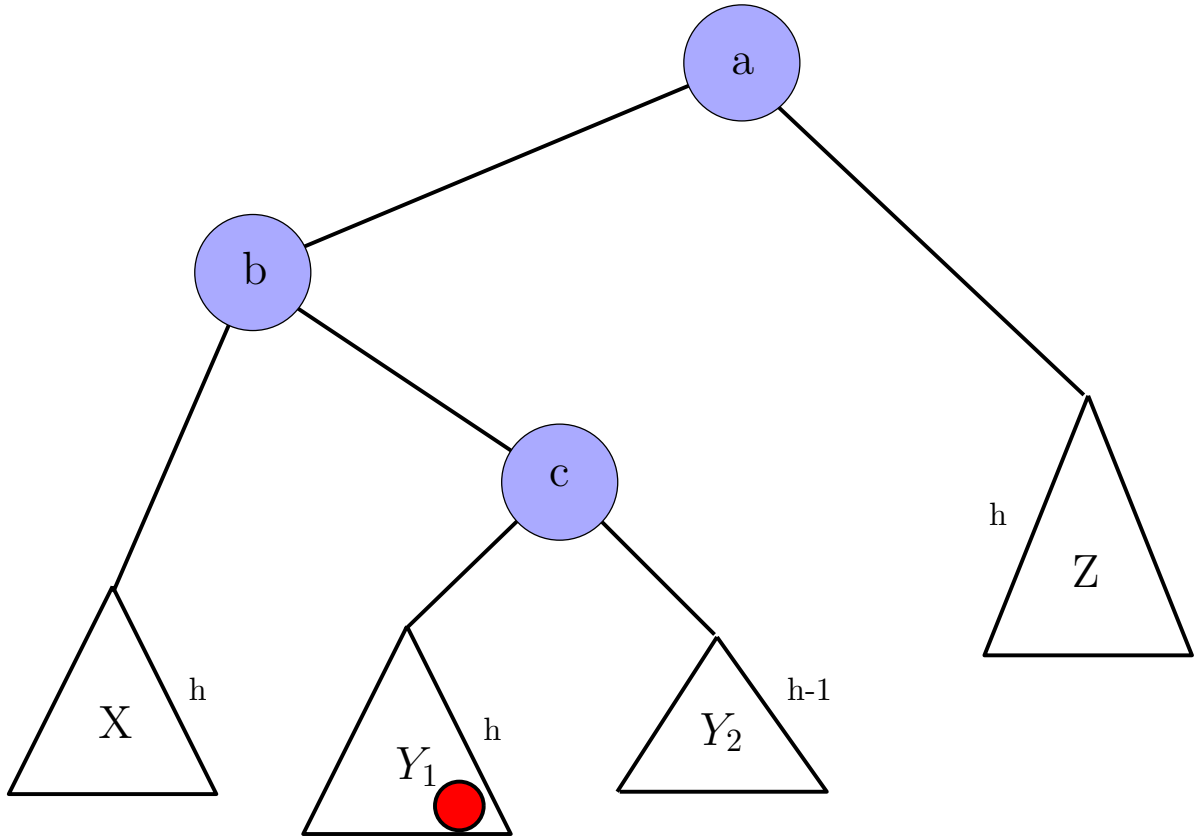
Άλλες περιπτώσεις

Στις περιπτώσεις $2\&3$ δεν λειτουργεί η μονή περιστροφή.



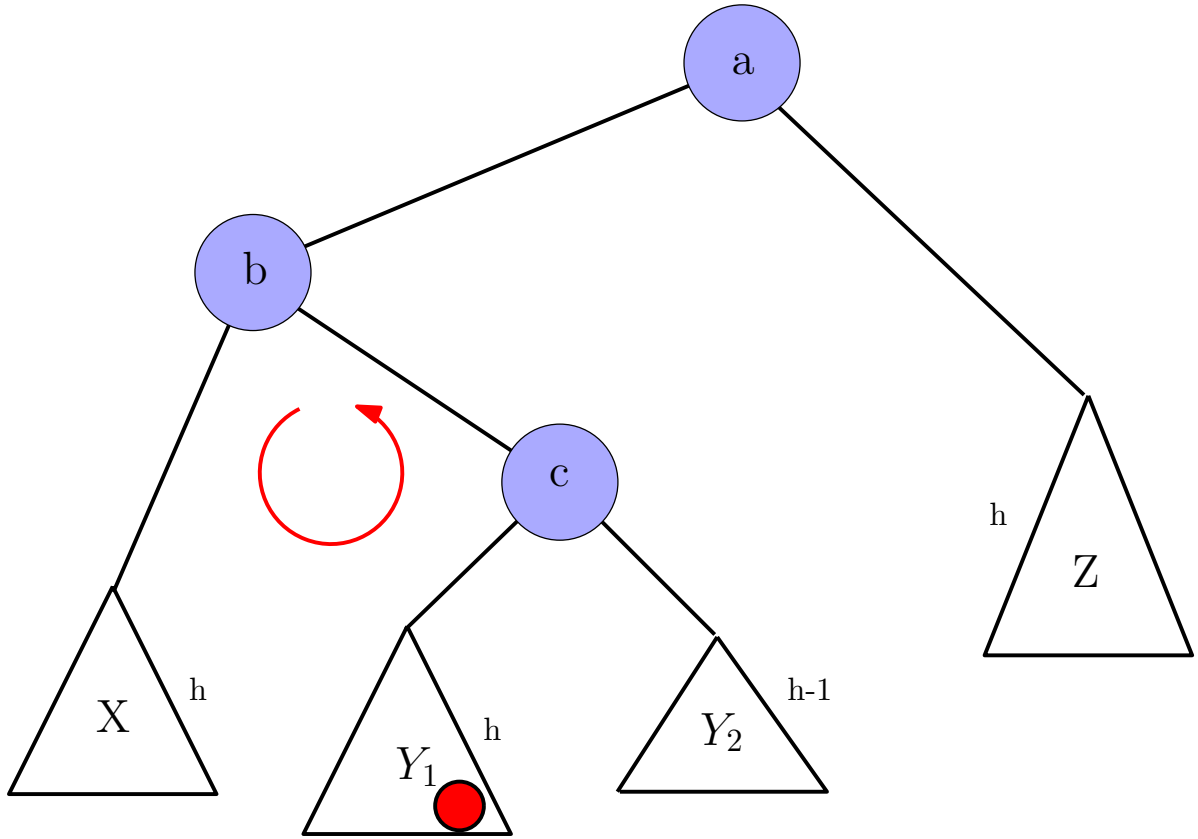
$$X < b < Y < a < Z$$

Διπλή Περιστροφή



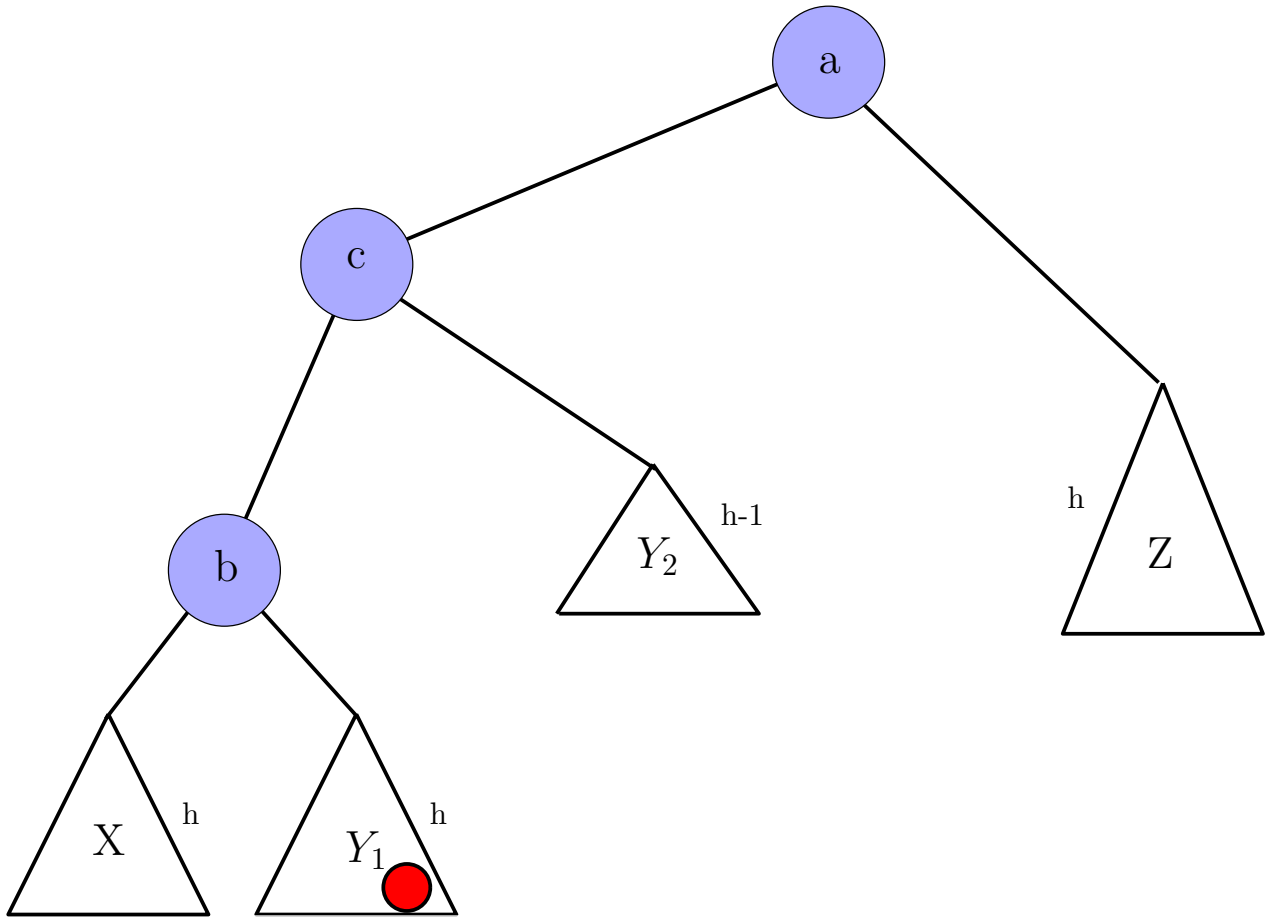
$$X < b < Y_1 < c < Y_2 < a < Z$$

Διπλή Περιστροφή



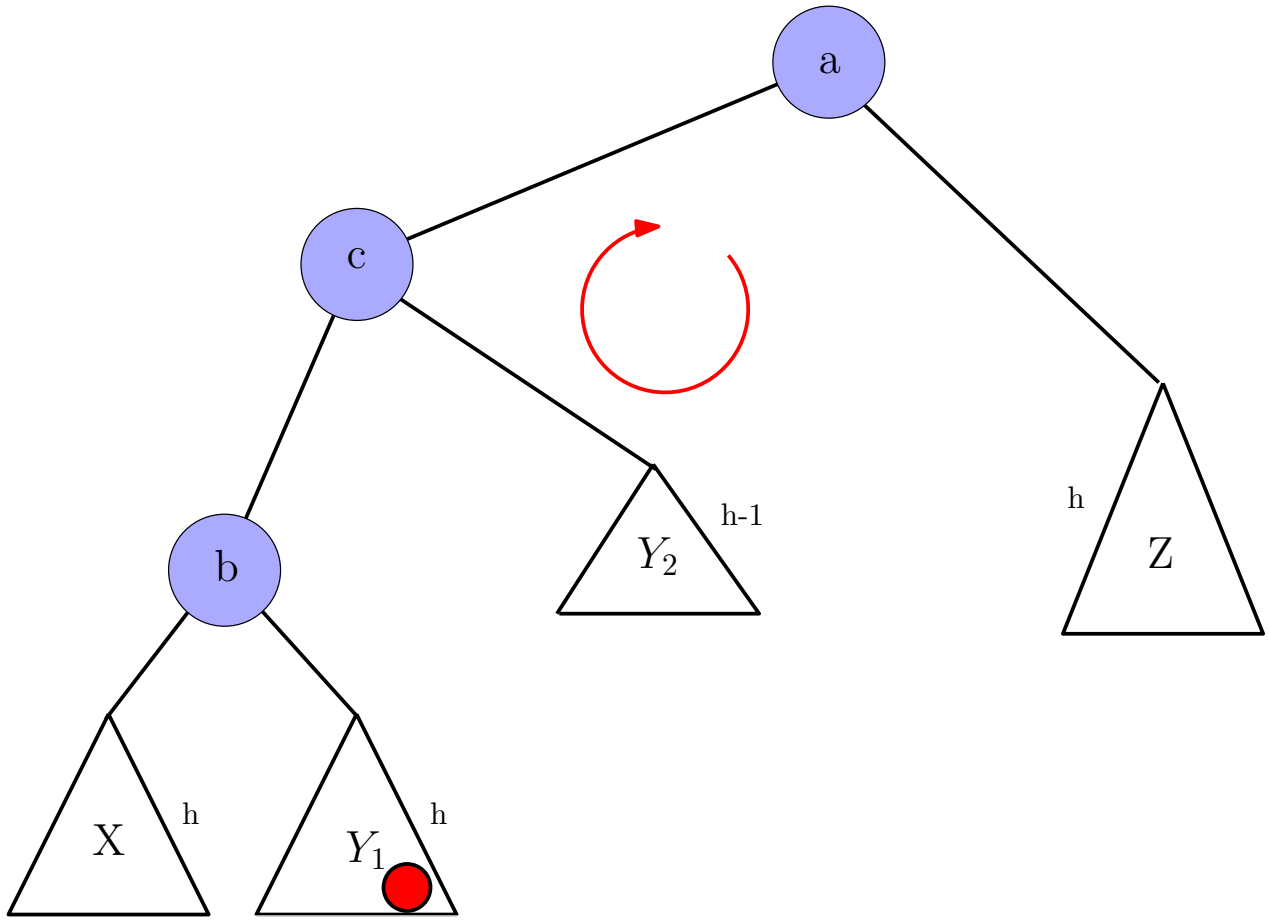
$$X < b < Y_1 < c < Y_2 < a < Z$$

Διπλή Περιστροφή



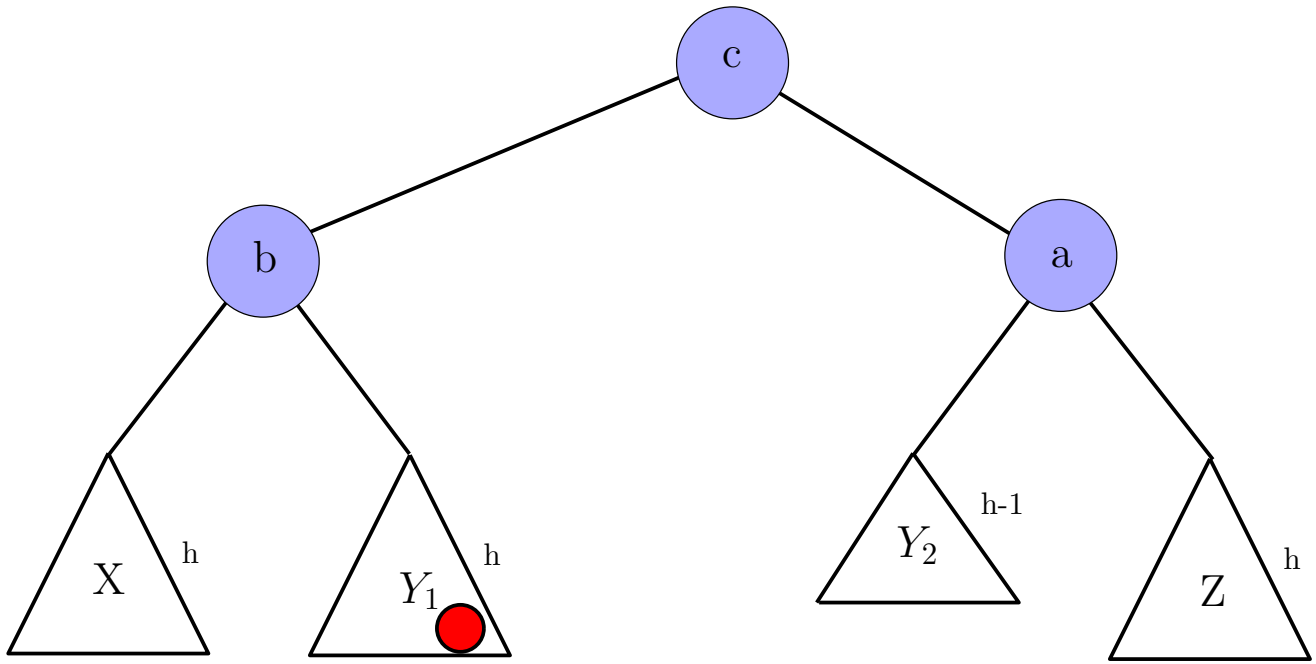
$$X < b < Y_1 < c < Y_2 < a < Z$$

Διπλή Περιστροφή



$$X < b < Y_1 < c < Y_2 < a < Z$$

Διπλή Περιστροφή



$$X < b < Y_1 < c < Y_2 < a < Z$$

Διαγραφή

Η διαγραφή στοιχείων από ένα AVL δέντρο γίνεται με τον ίδιο τρόπο όπως και σε ΔΔΑ με επιπλέον πιθανές περιστροφές.

$\mathcal{O}(\log n)$ στην χειρότερη περίπτωση

Ιεραρχία Μνήμης

(Χρονιά 2010)

Υπάρχει μεγάλο δίλημα μεταξύ ταχύτητας και μεγέθους όσο αφορά την μνήμη.

SRAM (Static Random Access Memory)

- Φτιάχνονται οι καταχωρητές
- Πολύ γρήγορη ($1ns = 10^{-9}sec$ - αντέχει ταχύτητες GHz)
- Πολύ ακριβή ($1GB \approx 5000\$$)

DRAM (Dynamic Random Access Memory)

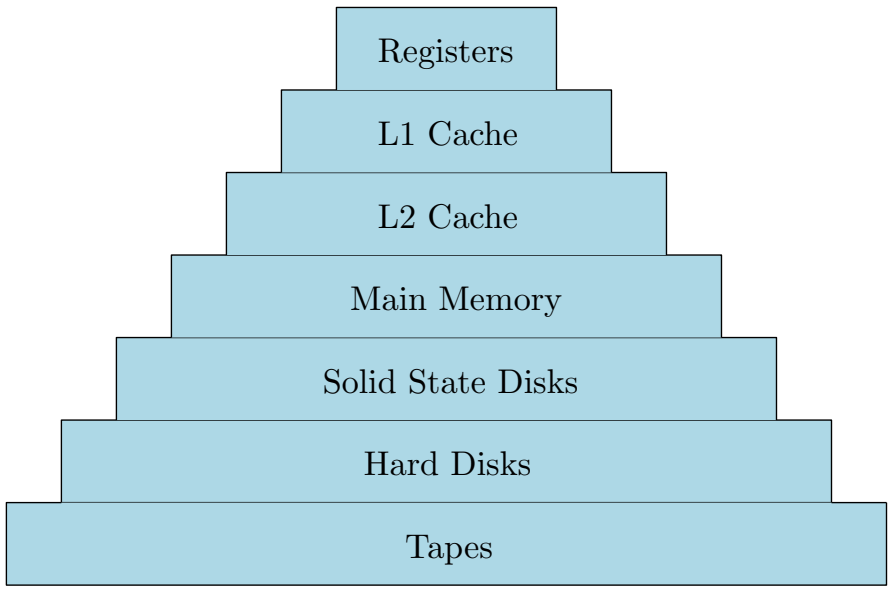
- Φτιάχνεται η μνήμη
- Γρήγορη (25ns)
- Λιγότερο ακριβή ($16GB \approx 100\$$)

Σκληρός Δίσκος

- Πολύ αργός ($5ms = 5 \cdot 10^{-3}sec$)
- Πολύ φθηνός

Ιεραρχία Μνήμης
(Χρονιά 2010)

Ιδέα: Χρησιμοποίησε πολλούς τύπους μνήμης για καλύτερη απόδοση.



0.25 - 1ns

1 - 5ns

5 - 25ns

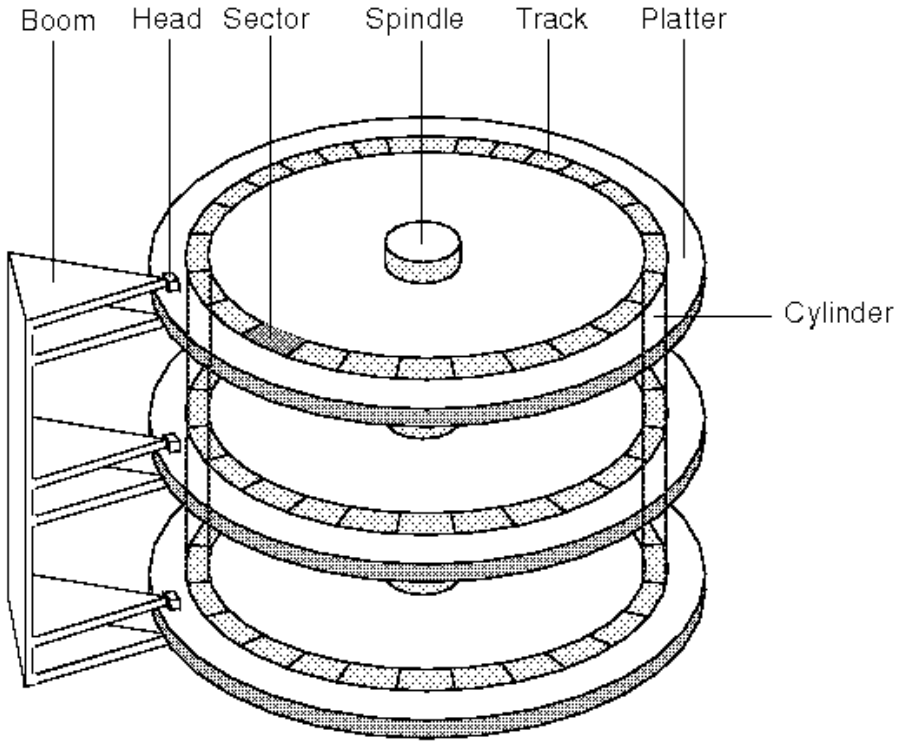
25 -100ns

100 - 150ns

2 - 20ms

sequential access

Εξωτερική Μνήμη



- seek time: χρόνος ώστε οι κεφαλές να μετακινηθούν στο κατάλληλο σημείο (track)
- rotational delay: χρόνος αναμονής ώστε να περιστραφεί ο κύλινδρος και να έρθει το κατάλληλο sector που ψάχνουμε

Εξωτερική Μνήμη

- η μνήμη των υπολογιστών (DRAM) έχει χρόνο πρόσβασης της τάξης των nanoseconds: 10^{-9} sec, πχ 25 - 100 nsec
- οι δίσκοι έχουν χρόνο πρόσβασης της τάξης των milliseconds: 10^{-3}, πχ 2 - 20 ms

Συμπέρασμα

Δεν συμφέρει να διαβάζουμε λίγα δεδομένα από το δίσκο. Για αυτό τον λόγο οι δίσκοι διαβάζουν πάντα ένα block που είναι της τάξης των Kilobytes, πχ 512K.

Βήτα Δέντρα

- δομή αναζήτησης η οποία είναι αποτελεσματική για αποθήκευση σε δίσκο,
- γενικεύει τα 2-3-4 δέντρα σε δέντρα με κόμβους μεταξύ t και $2t$ για $t \geq 2$.

Ορισμός Βήτα Δέντρων

Ένα δέντρο *B-tree* είναι ένα δέντρο με ρίζα και τις εξής ιδιότητες:

1. κάθε κόμβος x έχει τα εξής πεδία:
   (a) $n[x]$, ο αριθμός των κλειδιών που έχει ο κόμβος x
   (b) τα $n[x]$ κλειδιά σε σειρά: $key_1[x] \leq key_2[x] \leq \ldots \leq key_{n[x]}[x]$
   (c) τιμή $leaf[x]$ που είναι αλήθεια αν και μόνο αν ο κόμβος x είναι φύλλο

Ορισμός Βήτα Δέντρων

Ένα δέντρο *B-tree* είναι ένα δέντρο με ρίζα και τις εξής ιδιότητες:

1. κάθε κόμβος x έχει τα εξής πεδία:
   (a) $n[x]$, ο αριθμός των κλειδιών που έχει ο κόμβος x
   (b) τα $n[x]$ κλειδιά σε σειρά: $key_1[x] \leq key_2[x] \leq \ldots \leq key_{n[x]}[x]$
   (c) τιμή $leaf[x]$ που είναι αλήθεια αν και μόνο αν ο κόμβος x είναι φύλλο
2. εαν ο x είναι εσωτερικός κόμβος περιέχει $n[x] + 1$ δείκτες $c_1[x], c_2[x], \ldots, c_{n[x]+1}[x]$ στα παιδιά του. Τα φύλλα δεν έχουν παιδιά οπότε οι δείκτες αυτοί δεν ορίζονται.

Ορισμός Βήτα Δέντρων

Ένα δέντρο *B-tree* είναι ένα δέντρο με ρίζα και τις εξής ιδιότητες:

1. κάθε κόμβος x έχει τα εξής πεδία:
   (a) $n[x]$, ο αριθμός των κλειδιών που έχει ο κόμβος x
   (b) τα $n[x]$ κλειδιά σε σειρά: $key_1[x] \leq key_2[x] \leq \ldots \leq key_{n[x]}[x]$
   (c) τιμή $leaf[x]$ που είναι αλήθεια αν και μόνο αν ο κόμβος x είναι φύλλο
2. εαν ο x είναι εσωτερικός κόμβος περιέχει $n[x] + 1$ δείκτες $c_1[x], c_2[x], \ldots, c_{n[x]+1}[x]$ στα παιδιά του. Τα φύλλα δεν έχουν παιδιά οπότε οι δείκτες αυτοί δεν ορίζονται.
3. Τα κλειδιά $key_i[x]$ χωρίζουν τις κλίμακες των κλειδιών που αποθηκεύονται σε κάθε υποδέντρο: εαν το k_i είναι ένα κλειδί που αποθηκεύεται στο υποδέντρο με ρίζα $c_i[x]$

$$k_1 \leq key_1[x] \leq k_2 \leq key_2[x] \leq \cdots \leq key_{n[x]}[x] \leq k_{n[x]+1}.$$

Ορισμός Βήτα Δέντρων

4. Κάθε φύλλο έχει το ίδιο βάθος, το ύψος του δέντρου h.

Ορισμός Βήτα Δέντρων

4. Κάθε φύλλο έχει το ίδιο βάθος, το ύψος του δέντρου h.
5. Έστω $t \geq 2$ ο **ελάχιστος βαθμός** του δέντρου:
   1. Κάθε κόμβος εκτός της ρίζας πρέπει να έχει τουλάχιστον $t-1$ κλειδιά (δηλαδή τουλάχιστον t παιδιά). Εαν το δέντρο είναι μη-κενό, η ρίζα πρέπει να έχει τουλάχιστον ένα κλειδί.
   2. Κάθε κόμβος μπορεί να περιέχει μέχρι $2t-1$ κλειδιά, δηλαδή το πολύ $2t$ παιδιά. Ένας κόμβος λέγεται **γεμάτος** (full) εαν περιέχει ακριβώς $2t-1$ κλειδιά.

Ορισμός Βήτα Δέντρων

4. Κάθε φύλλο έχει το ίδιο βάθος, το ύψος του δέντρου h.
5. Έστω $t \geq 2$ ο **ελάχιστος βαθμός** του δέντρου:
   1. Κάθε κόμβος εκτός της ρίζας πρέπει να έχει τουλάχιστον $t-1$ κλειδιά (δηλαδή τουλάχιστον t παιδιά). Εαν το δέντρο είναι μη-κενό, η ρίζα πρέπει να έχει τουλάχιστον ένα κλειδί.
   2. Κάθε κόμβος μπορεί να περιέχει μέχρι $2t-1$ κλειδιά, δηλαδή το πολύ $2t$ παιδιά. Ένας κόμβος λέγεται **γεμάτος** (full) εαν περιέχει ακριβώς $2t-1$ κλειδιά.

Το πιο απλό Βήτα-δέντρο είναι για $t = 2$. Κάθε εσωτερικός κόμβος έχει 2, 3 ή 4 παιδιά δηλαδή ένα 2-3-4 δέντρο.

Στην πράξη όμως έχουμε πολύ μεγαλύτερες τιμές για το t.

Ύψος Βήτα Δέντρου

Θεώρημα

Ένα Βήτα δέντρο με $n \geq 1$ κλειδιά και ελάχιστο βαθμό $t \geq 2$ έχει ύψος

$$h \leq log_t \frac{n+1}{2}.$$

Απόδειξη

Ένα Βήτα δέντρο με ύψος h έχει ελάχιστο αριθμό κόμβων αν η ρίζα έχει ένα κλειδί και όλοι οι υπόλοιποι κόμβοι $t-1$ κλειδιά.

Στο επίπεδο 0 υπάρχει ένα κόμβος. Στο επίπεδο 1 υπάρχουν 2 κόμβοι. Στο επίπεδο 2 υπάρχουν $2t$ κόμβοι, στο επίπεδο 3 υπάρχουν $2t^2$ κόμβοι, κ.τ.λ □

Ύψος Βήτα Δέντρου

Θεώρημα

Ένα Βήτα δέντρο με $n \geq 1$ κλειδιά και ελάχιστο βαθμό $t \geq 2$ έχει ύψος

$$h \leq log_t \frac{n+1}{2}.$$

Απόδειξη

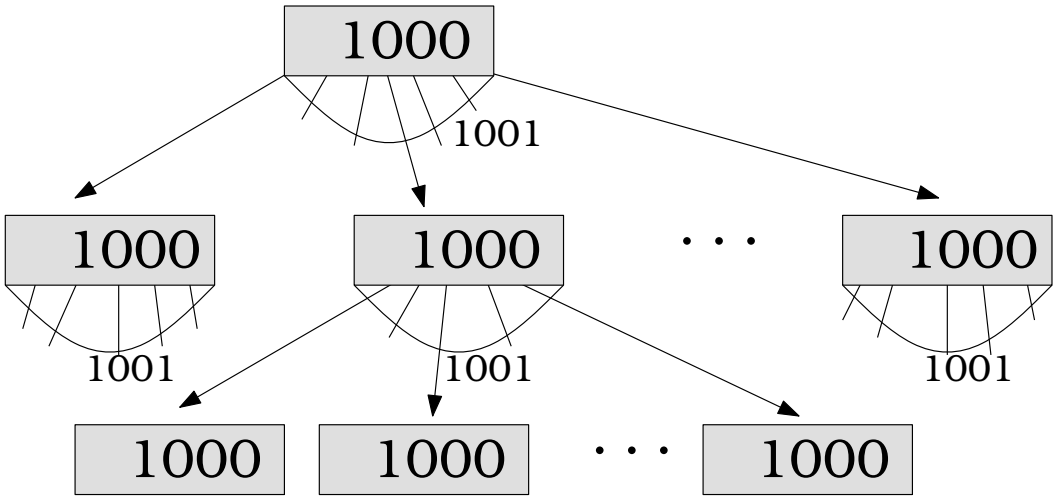
Ο αριθμός λοιπόν των κλειδιών είναι:

$$\begin{aligned} n &\geq 1 + (t-1)\sum_{i=1}^{h} 2t^{i-1} \\ &= 1 + 2(t-1)\left(\frac{t^h - 1}{t-1}\right) \\ &= 2t^h - 1 \end{aligned}$$

□

Προσπελάσεις Δίσκου

Φροντίζοντας κάθε κόμβος να χωρά σε μια σελίδα δίσκου, κάνουμε τόσες προσπελάσεις δίσκου όσες και το ύψος του δέντρου



1 κόμβος
1000 κλειδιά

1001 κόμβοι
1.001.000 κλειδιά

1.002.001 κόμβοι
1.002.001.000 κλειδιά

Το δέντρο έχει ύψος $\mathcal{O}(\log_t n)$.

Βασικές Παραδοχές για Υλοποίηση Βήτα Δέντρων

(i) Η ρίζα του Βήτα δέντρου θα είναι πάντα στην μνήμη

(ii) Θα έχουμε δύο συναρτήσεις που διαβάζουν και γράφουν έναν κόμβο στον σκληρό δίσκο

- `DISK_READ(x)`
- `DISK_WRITE(x)`

Αναζήτηση σε Βήτα Δέντρο

Η αναζήτηση δουλεύει όπως και στα ΔΔΑ (Δυαδικά Δέντρα Αναζήτησης) μόνο που τώρα έχουμε περισσότερες επιλογές σε κάθε κόμβο.

$B-TREE-SEARCH(x, k)$

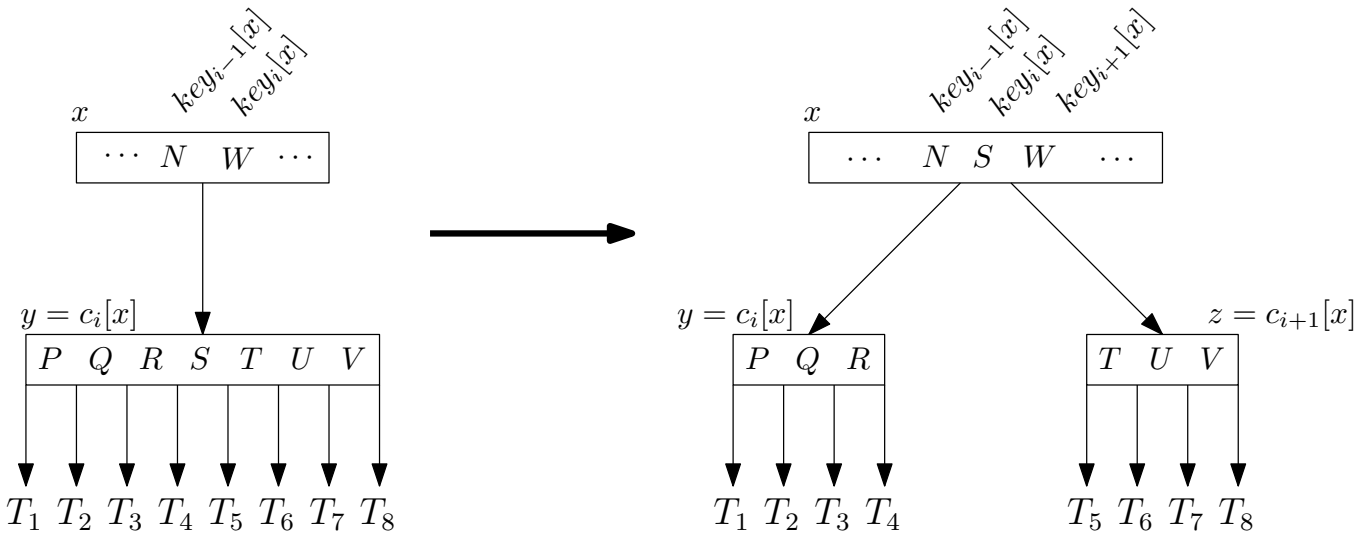
$i = 1$
while $i \leq n[x]$ *and* $k > key_i[x]$ **do**
  $i = i + 1$
end
if $i \leq n[x]$ *and* $k = key_i[x]$ **then**
  return (x, i)
end
if $leaf[x]$ **then**
  return *nil*
else
  $DISK-READ(c_i[x])$
end
return $B-TREE-SEARCH(c_i[x], k)$

Εισαγωγή σε Βήτα Δέντρα

- Καθώς αναζητούμε την θέση εισαγωγής ενός στοιχείου, διαιρούμε στα δύο οποιουσδήποτε κόμβους είναι γεμάτοι (έχουν $2t - 1$ κλειδιά).
- Όταν διαιρείται ένας κόμβος το μεσαίο κλειδί ανεβαίνει προς τα επάνω.
- Με αυτό τον τρόπο κάνουμε χώρο για πιθανές διαιρέσεις κόμβων που βρίσκονται παρακάτω.

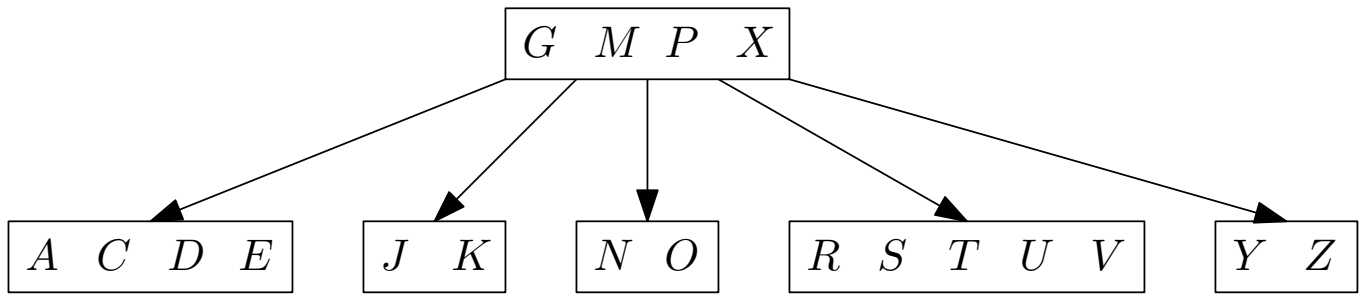
Διαίρεση κόμβου (split)

$t = 4$



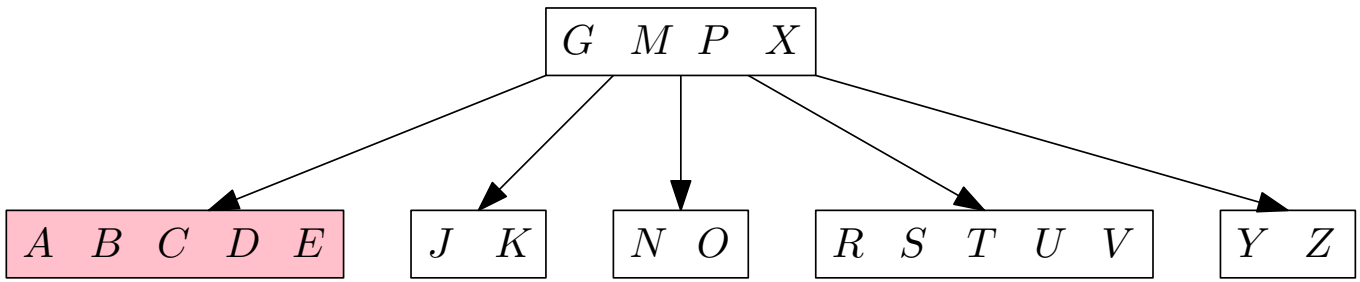
Παράδειγμα

$t = 3$



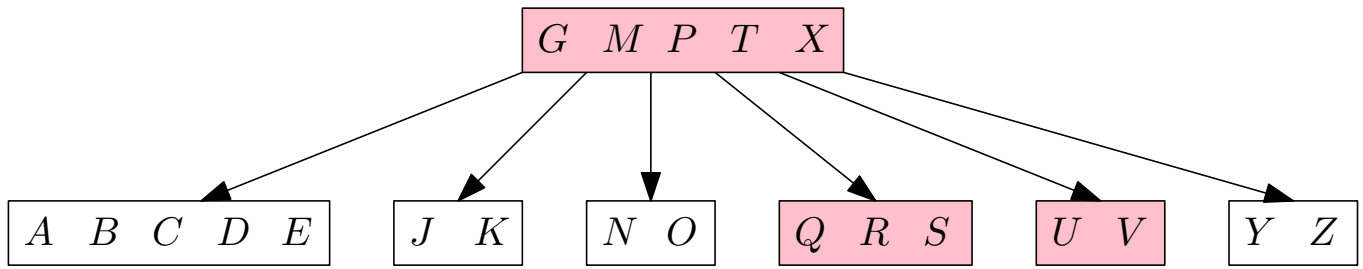
Παράδειγμα

$t = 3$, εισαγωγή του B



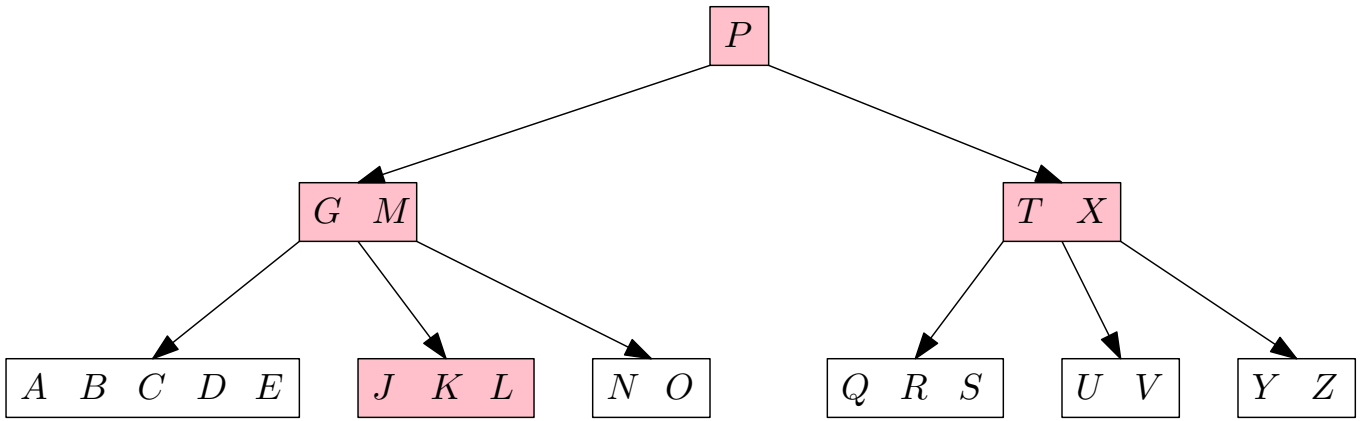
Παράδειγμα

$t = 3$, εισαγωγή του Q



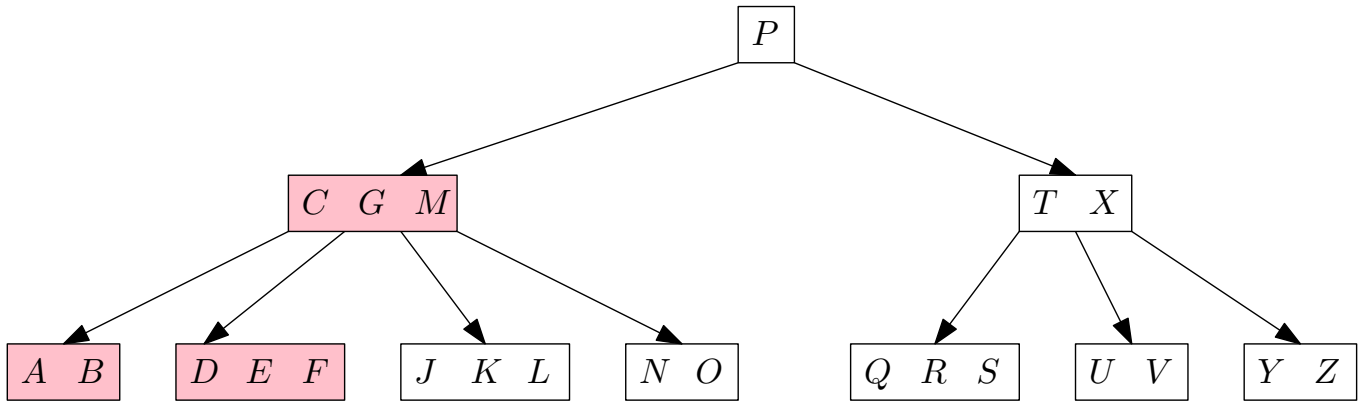
Παράδειγμα

$t = 3$, εισαγωγή του L



Παράδειγμα

$t = 3$, εισαγωγή του F



Διαγραφή από B-Δέντρα

Λειτουργεί με παρόμοιο τρόπο με τα 2-3-4 δέντρα.

Μπορεί να χρειαστεί να κάνουμε

- balance
- fuse

Διάβασμα και Επιπλέον Πηγές

- Ενότητες 7.1, 7.2, 7.3

  Kurt Mehlhorn και Peter Sanders. Αλγόριθμοι και Δομές Δεδομένων, Τα βασικά εργαλεία, Έκδοση 1η, Κλειδάριθμος, 2014.
- Ενότητες 12.5, 12.6, 12.8, 12.9, 13.3, 13.4, 16.3

  Robert Sedgewick. Αλγόριθμοι σε C: Θεμελιώδεις Έννοιες, Δομές Δεδομένων, Ταξινόμηση, Αναζήτηση. Έκδοση 3η, Κλειδάριθμος, 2006.
- Ενότητες 12.1, 12.2, 12.3, 13.1, 13.2, 13.3, 13.4

  Cormen, Leiserson, Rivest και Stein, Εισαγωγή στους Αλγορίθμους (σε ένα τόμο). Έκδοση 1η, Πανεπιστημιακές Εκδόσεις Κρήτης, 2012.