

Προγραμματιστικές Τεχνικές

Διδάσκοντες:	Νίκος Παπασπύρου Αρης Παγουρτζής Γιώργος Στάμου Γιώργος Γκούμας Παρασκευή Τζούβελη Γιώργος Σιόλας Γιώργος Αλεξανδρίδης	(nickie@softlab.ntua.gr) (pagour@cs.ntua.gr) (gstam@cs.ntua.gr) (goumas@cslab.ece.ntua.gr) (tpar@image.ntua.gr) (gsiolas@islab.ntua.gr) (gealexan@mail.ntua.gr)
--------------	--	---

<http://mycourses.ntua.gr/courses/ECE1086/>

Δένδρα πολλαπλών οδεύσεων

Γιώργος Στάμου

Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Εθνικό Μετσόβιο Πολυτεχνείο



Δένδρα αναζήτησης m-οδεύσεων

- Ένα δένδρο αναζήτησης m -οδεύσεων, T , είναι ένα δέντρο όπου σε κάθε κόμβο υπάρχει ένα πεπερασμένο σύνολο τιμών (κλειδιών).
- Το δέντρο είναι είτε κενό, $T = \emptyset$ ή αποτελείται από n υποδένδρα m -οδεύσεων $T_0, T_1, \dots, T_{n-1}$, με $n-1$ κλειδιά $k_1, k_2, \dots, k_{n-1}$.
- Το δένδρο μπορεί να παρασταθεί σαν μία λίστα

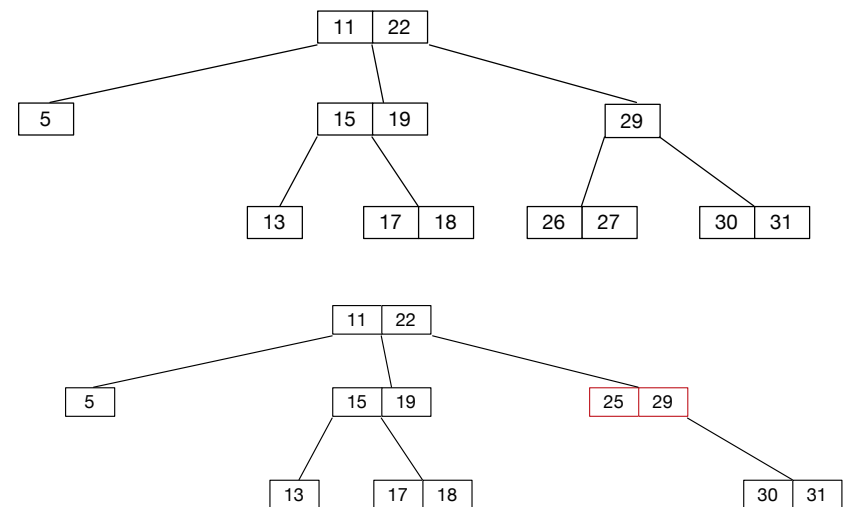
$$T = [T_0, k_1, T_1, k_2, T_2, \dots, k_{n-1}, T_{n-1}]$$

όπου $2 \leq n \leq M$, και τα κλειδιά και οι κόμβοι ικανοποιούν τις παρακάτω συνθήκες διάταξης:

Δένδρα αναζήτησης m-οδεύσεων

- Τα κλειδιά σε κάθε κόμβο είναι διακεκριμένα και ταξινομημένα δηλαδή, $k_i < k_{i+1}$ για $1 \leq i \leq n-2$.
- Όλα τα κλειδιά που περιέχονται σε ένα υποδέντρο T_i έχουν τιμή μικρότερη από την τιμή του κλειδιού k_{i+1} , δηλαδή,
 $\forall k \in T_i : k < k_{i+1}$ για $0 \leq i \leq n-2$. Το δέντρο T_i ονομάζεται το «αριστερό υποδέντρο» με αναφορά το κλειδί k_{i+1} .
- Όλα τα κλειδιά που περιέχονται στο υποδέντρο T_i έχουν τιμές μεγαλύτερες από την τιμή του κλειδιού k_i , δηλαδή,
 $\forall k \in T_i : k > k_i$ για $1 \leq i \leq n-1$. Το δέντρο T_{i+1} ονομάζεται το «δεξιό υποδέντρο» με αναφορά το κλειδί k_{i+1} .

Δένδρα αναζήτησης m-οδεύσεων



Αναζήτηση σε δένδρα m -οδεύσεων

- Η αναζήτηση ξεκινά από τη ρίζα.
- Εάν το δένδρο είναι κενό η αναζήτηση αποτυγχάνει.
- Αλλιώς, εξετάζουμε τα κλειδιά που περιέχονται στη ρίζα για να δούμε εάν τιμή που αναζητούμε βρίσκεται στο σύνολο των κλειδιών της ρίζας. Εάν βρούμε την τιμή τότε η αναζήτηση τελειώνει με επιτυχία.

Αναζήτηση σε δένδρα m -οδεύσεων

- Εάν δεν βρούμε την τιμή που αναζητούμε, τότε υπάρχουν τρεις περιπτώσεις (εδώ ψάχνουμε για την τιμή x):
 - η τιμή x , είναι μικρότερη από k_1 , όπου σε αυτή τη περίπτωση ψάχνουμε στο υποδένδρο T_0
 - η τιμή x είναι μεγαλύτερη από k_{n-1} , όπου σε αυτή τη περίπτωση ψάχνουμε στο υποδένδρο T_{n-1}
 - υπάρχει δείκτης θέσης i τέτοιος ώστε $1 \leq i < n - 1$ για τον οποίο $k_i < x < k_{i+1}$, όπου σε αυτή τη περίπτωση ψάχνουμε στο υποδένδρο T_i

Πολυπλοκότητα αναζήτησης σε δένδρα m -οδεύσεων

- Η απλοϊκή μέθοδος χρησιμοποιεί απλή γραμμική αναζήτηση σε κάθε κόμβο
- Η χρονική πολυπλοκότητα για τη χειρότερη περίπτωση είναι της τάξης (T_{compare} είναι η πολυπλοκότητα της σύγκρισης δύο κλειδιών):

$$(M - 1)(h+1)T_{\text{compare}} + O(Mh)$$

όπου h είναι το ύψος του δένδρου

- Μια καλύτερη έκδοση χρησιμοποιεί δυαδική αναζήτηση
Η χρονική πολυπλοκότητα για τη χειρότερη περίπτωση είναι της τάξης:

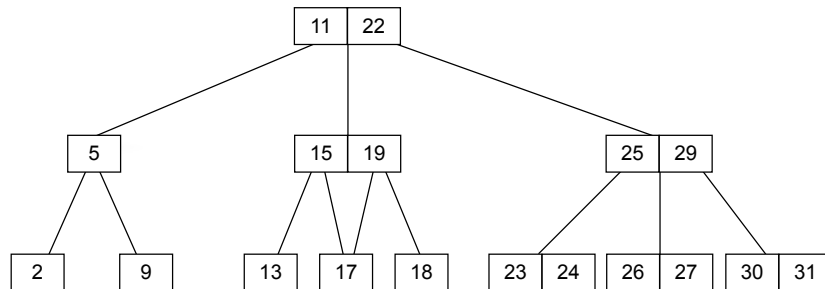
$$(h+1)(\lceil \log_2(M - 1) \rceil + 2)T_{\text{compare}} + O(h \log M)$$

B-δένδρα

Ένα B-δένδρο τάξης M ορίζεται ως είτε ένα κενό δένδρο ή ως ένα δέντρο αναζήτησης m -οδεύσεων T με τις παρακάτω ιδιότητες T :

1. Η ρίζα του δένδρου T έχει τουλάχιστον δύο υποδένδρα και το πολύ M υποδένδρα.
2. Όλοι οι εσωτερικοί κόμβοι του δένδρου T (εκτός της ρίζας) έχουν μεταξύ $\lceil M/2 \rceil$ και M υποδένδρα.
3. Όλα τα φύλλα του δένδρου T βρίσκονται στο ίδιο επίπεδο.

B-δένδρα



Αριθμός κλειδιών σε ένα B-δένδρο

- Θεώρημα: Ο ελάχιστος αριθμός κλειδιών (τιμών των κόμβων) σε ένα B-δένδρο τάξης $M \geq 2$ και ύψους $h \geq 0$ είναι $n_h = 2 \lceil M/2 \rceil^h - 1$.
- Απόδειξη: με επαγωγή

Εισαγωγή στοιχείων σε B-δένδρο

- Η «αναζήτηση» για το στοιχείο που πάμε να εισάγουμε καταλήγει σε φύλλο
- Πάντα εισάγουμε ένα νέο στοιχείο σαν φύλλο
- Εάν το φύλλο περιέχει λιγότερα από $M - 1$ κλειδιά (τιμές) $[T_0, k_1, T_1, k_2, T_2, \dots, k_{n-1}, T_{n-1}]$
 - Εισάγουμε το νέο στοιχείο (κλειδί) και φτιάχνουμε ένα κενό κόμβο όπως πρέπει για να κρατηθεί το σχήμα του δέντρου:

$$[T_0, k_1, T_1, k_2, T_2, \dots, k_i, T_i, x, \emptyset, k_{i+1}, T_{i+1}, \dots, k_{n-1}, T_{n-1}]$$

Εισαγωγή στοιχείων σε B-δένδρο (συμπληρωμένα φύλλα)

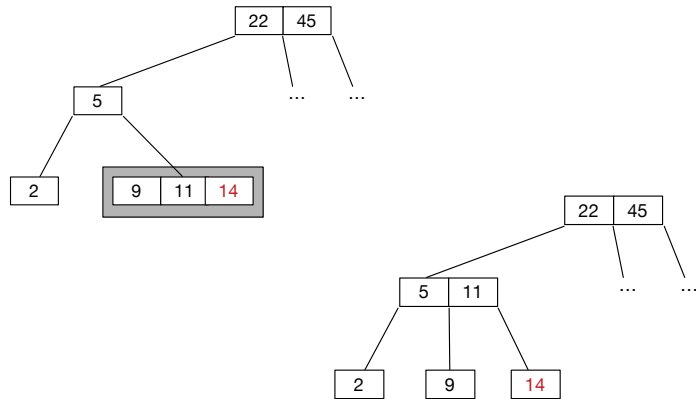
- Χωρίζουμε τον κόμβο φύλλο στη μέση (σημειώστε ότι κάθε μισό εξακολουθεί να είναι μέρος ενός έγκυρου B-δένδρου!)

$$T'_L = [T_0, k_1, T_1, \dots, k_{\lceil M/2 \rceil - 1}, T_{\lceil M/2 \rceil - 1}]$$

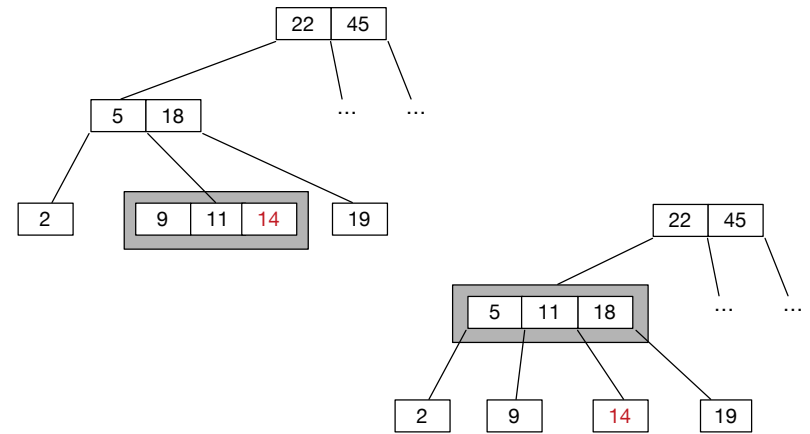
$$T'_R = [T_{\lceil M/2 \rceil}, k_{\lceil M/2 \rceil + 1}, T_{\lceil M/2 \rceil + 1}, \dots, k_M, T_M]$$
- Έστω ότι ο κόμβος T δεν είναι η ρίζα
 - Έχουμε δύο κόμβους, T'_L and T'_R , και ένα κλειδί που περισσεύει, $k_{\lceil M/2 \rceil}$
 - T'_L αντικαθιστά τον κόμβο T στον γονέα του T
 - Το ζευγάρι $(k_{\lceil M/2 \rceil}, T'_R)$ εισάγεται αναδρομικά στον γονέα του T
- Εάν ο κόμβος T είναι ρίζα, κατασκευάζουμε ένα νέο κόμβο:

$$[T'_L, k_{\lceil M/2 \rceil}, T'_R]$$

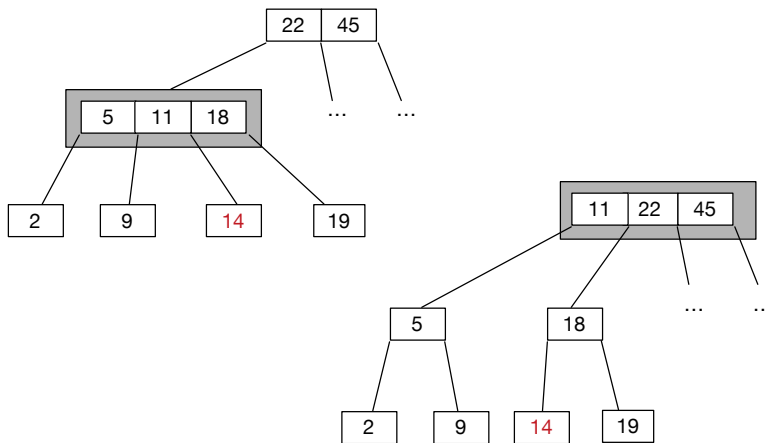
Εισαγωγή στοιχείων σε B-δένδρο – Παράδειγμα



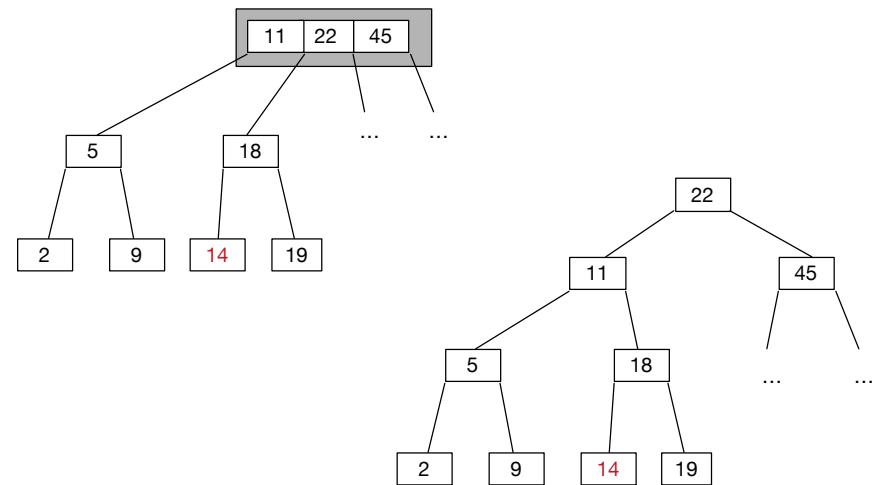
Εισαγωγή στοιχείων σε B-δένδρο – Παράδειγμα



Εισαγωγή στοιχείων σε B-δένδρο – Παράδειγμα



Εισαγωγή στοιχείων σε B-δένδρο – Παράδειγμα



Διαγραφή στοιχείων από B-δένδρο

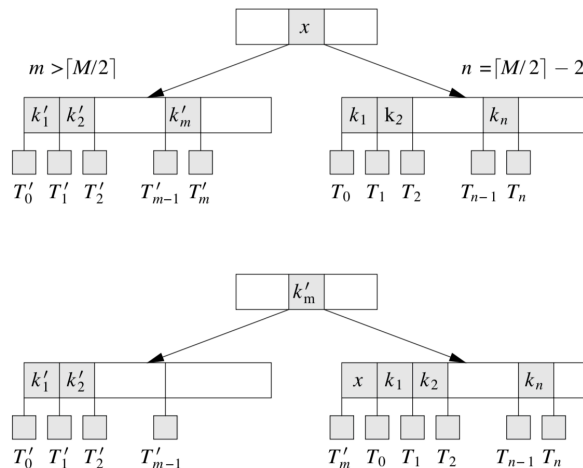
- Η «αναζήτηση» για το στοιχείο που πάμε να διαγράψουμε μπορεί να καταλήξει σε οποιοδήποτε κόμβο
- Αν ο κόμβος δεν είναι φύλλο διαγράφουμε το στοιχείο, αντικαθιστώντας το με προηγούμενο (δεξιότερο στοιχείο του αριστερού του υποδένδρου) ή το επόμενο (αριστερότερο στοιχείο του δεξιότερου υποδένδρου), όπως και στην περίπτωση των BST
- Διαγράφουμε το στοιχείο το οποίο χρησιμοποιήσαμε στην αντικατάσταση (αυτό είναι σε φύλλο - γιατί;)
- Η διαγραφή αυτή μπορεί να δημιουργήσει πρόβλημα (αν δεν είναι το φύλλο αυτό ρίζα - γιατί αν είναι στη ρίζα δεν υπάρχει πρόβλημα;).

Διαγραφή στοιχείων από B-δένδρο

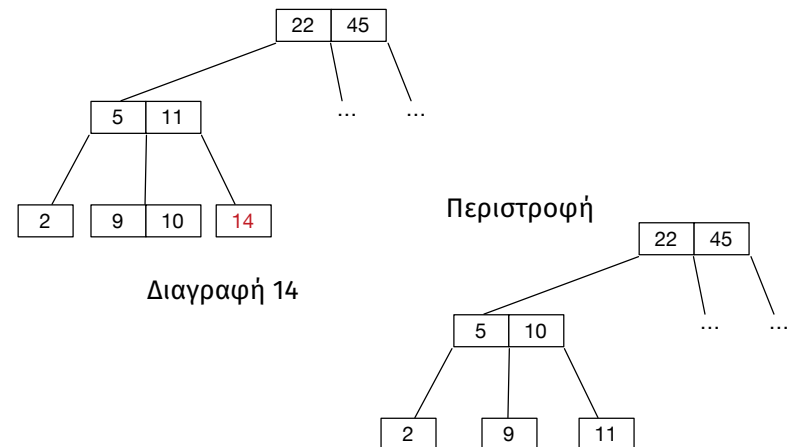
- Το πρόβλημα μπορεί να δημιουργηθεί από την παραβίαση της συνθήκης 2 του ορισμού του B-δένδρου (όλοι οι εσωτερικοί κόμβοι του δέντρου T (εκτός της ρίζας) έχουν μεταξύ $\lceil M/2 \rceil$ και M υποδένδρα.)
- Η διαχείριση του προβλήματος αυτού γίνεται με τις εξής τεχνικές:
 - **Περιστοφή**, αν υπάρχει αριστερός ή δεξιός γείτονας με περισσότερα από $\lceil M/2 \rceil$ παιδιά (βλ. Σχήμα επόμενης διαφάνειας)
 - **Συγχώνευση**, αν και οι δύο γείτονές του (αριστερός και δεξιός) έχουν ακριβώς $\lceil M/2 \rceil$ παιδιά (βλ. Σχήμα μεθεπόμενης διαφάνειας)

Διαγραφή στοιχείων σε B-δένδρο

Περιστοφή



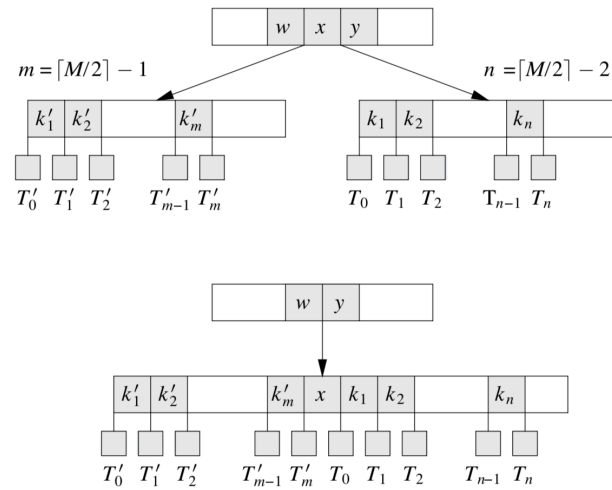
Διαγραφή στοιχείων σε B-δένδρο



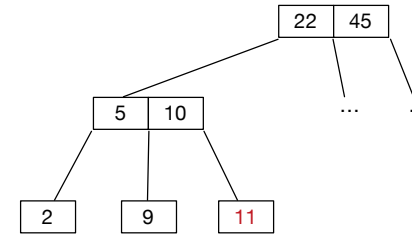
Διαγραφή 14

Διαγραφή στοιχείων σε B-δένδρο

Συγχώνευση



Διαγραφή στοιχείων σε B-δένδρο



Διαγραφή 11

Συγχώνευση

