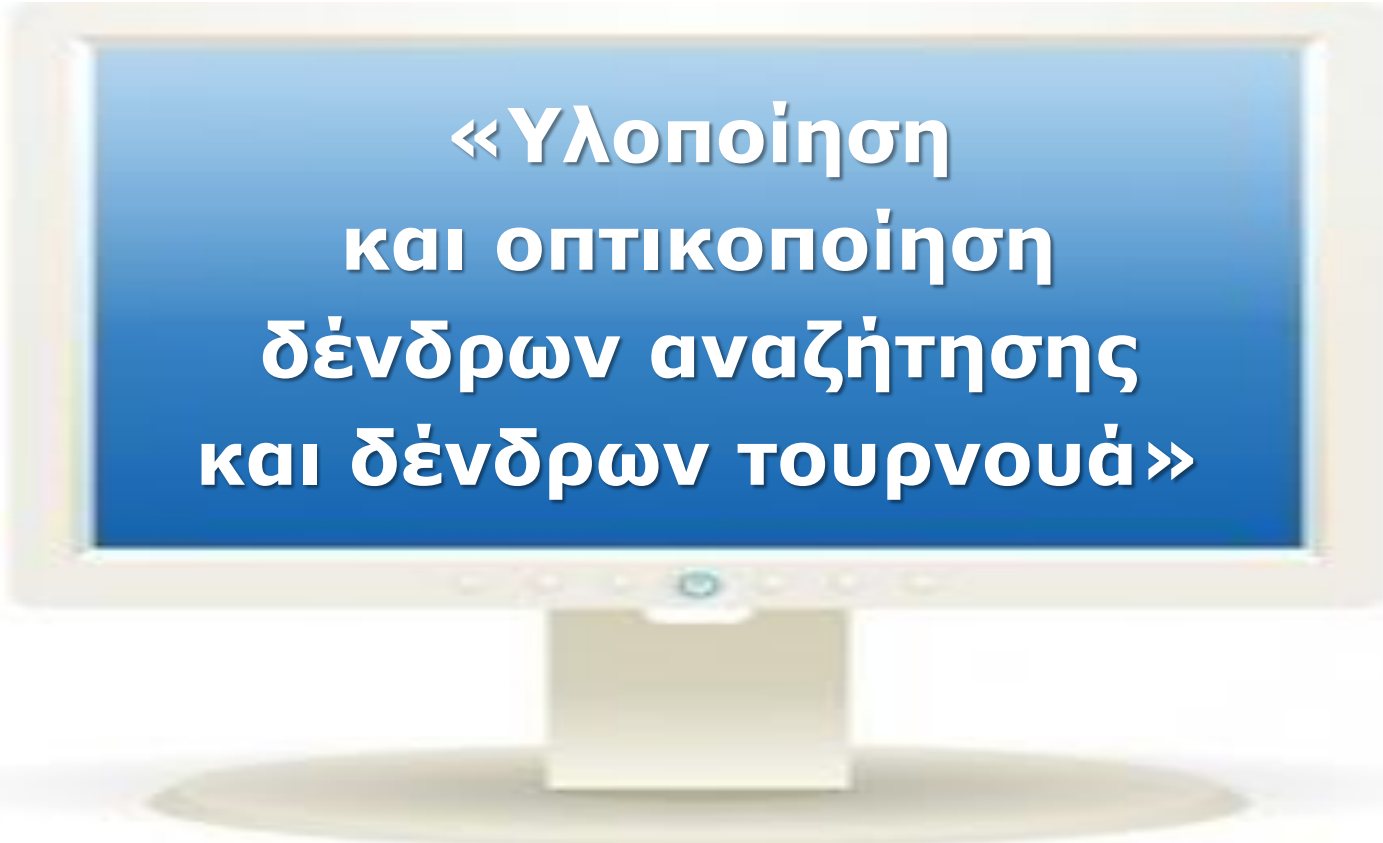




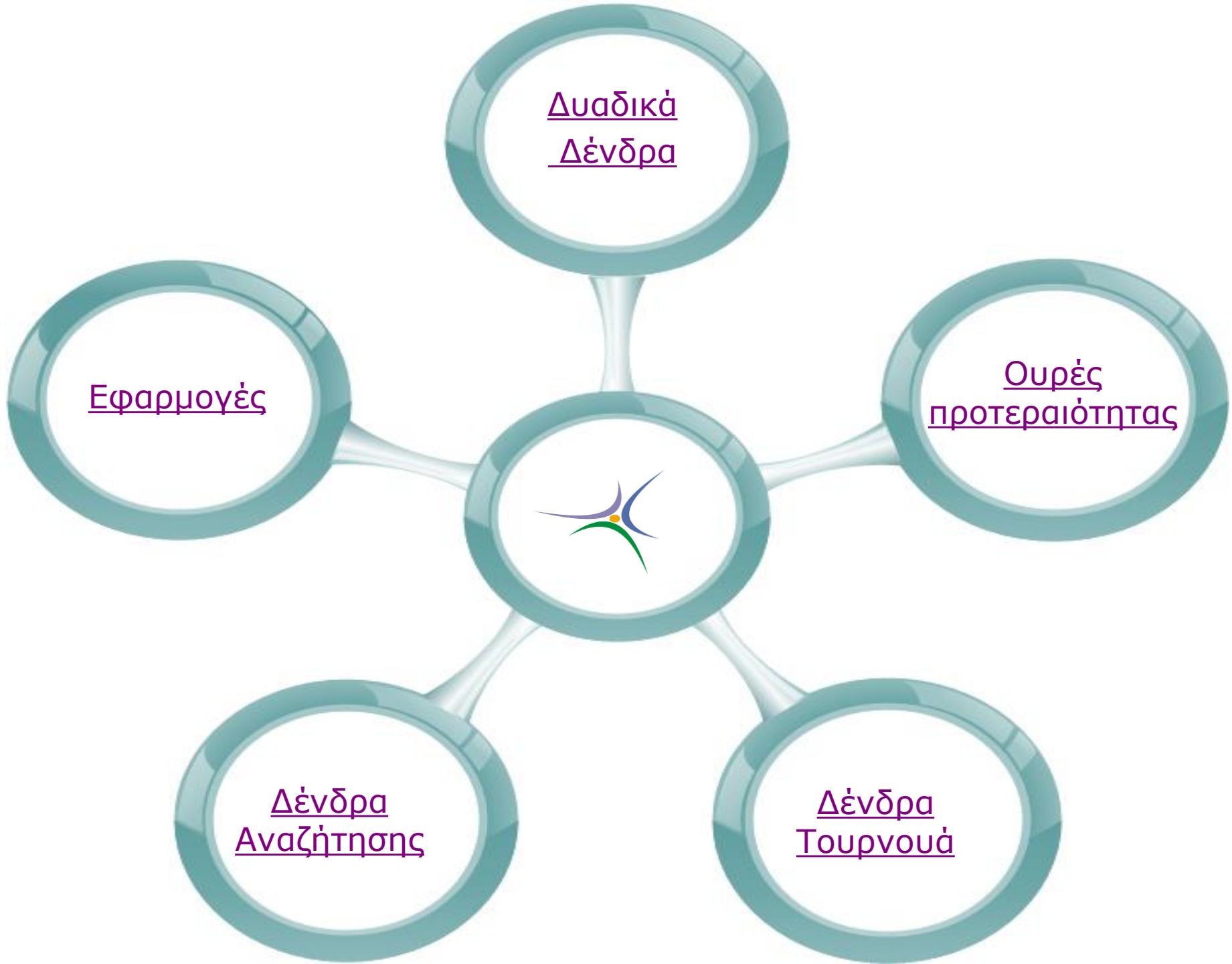
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κεντρικής Μακεδονίας  
Τμήμα Μηχανικών Πληροφορικής Τ.Ε.  
Τομέας Υπολογιστικών Τεχνικών & Συστημάτων



**«Υλοποίηση  
και οπτικοποίηση  
δένδρων αναζήτησης  
και δένδρων τουρνουά»**

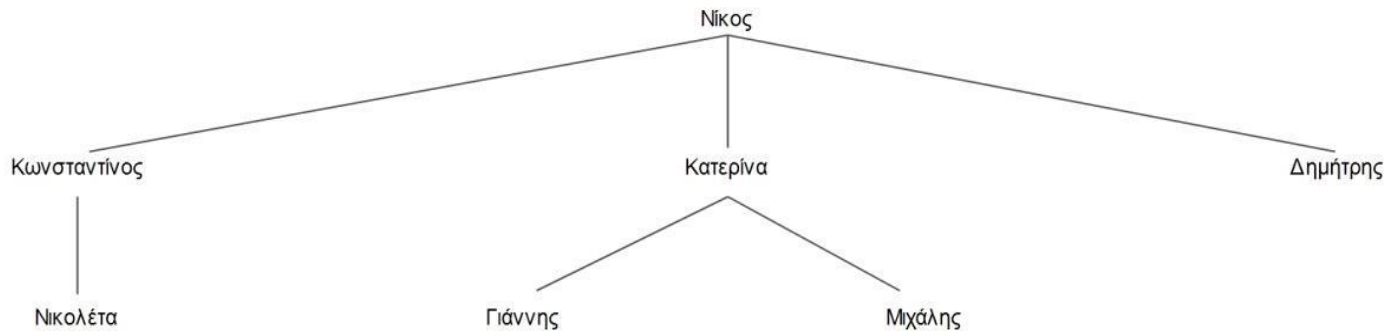
Σαβτσούκ Αλέξιος (2983)

Επιβλέπων Καθηγητής: Πάρις Μαστροκώστας



# Δένδρα

Ένα δένδρο  $t$  είναι ένα πεπερασμένο μη κενό σύνολο στοιχείων. Το πρώτο στοιχείο ονομάζεται ρίζα, ενώ τα υπόλοιπα στοιχεία (αν υπάρχουν) επιμερίζονται σε δένδρα που ονομάζονται υποδένδρα του  $t$ .



- Ιεραρχικό δένδρο
  - Γενεαλογικό
  - Εταιρική δομή
  - Ένα πρόβλημα
  - Εκφράσεων
- Δυαδικό δένδρο

# Διαδικά Δένδρα

Ένα διαδικό δένδρο  $t$  είναι ένα πεπερασμένο σύνολο κόμβων. Αν δεν είναι κενό τότε έχει μία ρίζα και τα υπόλοιπα στοιχεία (αν υπάρχουν) χωρίζονται σε δύο διαδικά δένδρα που το καθένα ονομάζεται αριστερό και δεξιό υποδένδρο του  $t$ .

Διαδικό δένδρο	Γενικό δένδρο
Μπορεί να είναι κενό.	Δεν μπορεί να είναι κενό.
Σε κάθε κόμβο μπορεί να υπάρχουν μέχρι και δύο υποδένδρα.	Σε κάθε κόμβο μπορούν να υπάρχουν οσαδήποτε μη κενά υποδένδρα.
Τα παιδιά ενός κόμβου αποτελούν ένα διατεταγμένο δένδρο.	Τα υποδένδρα δεν είναι διατεταγμένα.

# Ιδιότητες Δυαδικών Δένδρων

---

*Ένα δυαδικό δένδρο:*

- έχει ακριβώς  $n-1$  ακμές, για  $n > 0$  στοιχεία
- έχει τουλάχιστον  $h$  και το πολύ  $2^h-1$  στοιχεία, για  $h > 0$  ύψος
- έχει ύψος το πολύ  $n$  και τουλάχιστον  $\log_2(n+1)$ , για  $n > 0$  στοιχεία
- είναι πλήρες όταν περιέχει ακριβώς  $2^h-1$  στοιχεία, για  $h$  ύψος
- είναι συμπληρωμένο όταν προκύπτει από ένα πλήρες δυαδικό δένδρο εάν διαγραφούν  $k$  στοιχεία με αρίθμηση  $2^h-1$ , για  $1 \leq i < k$  ( $k \geq 0$ ) και  $h$  ύψος

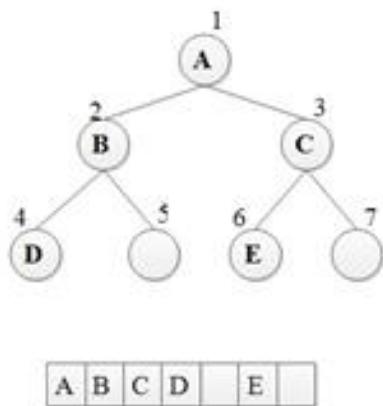
# Αναπαράσταση Δυαδικών Δένδρων

## □ Αναπαράσταση Βασισμένη σε Τύπους

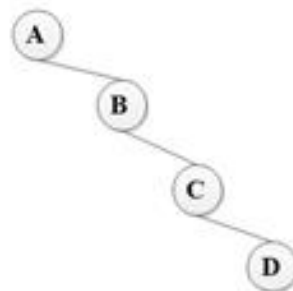
- Συμπληρωμένο δυαδικό δένδρο
- Δεξιά λοξό δυαδικό δένδρο

## □ Συνδεδεμένη Αναπαράσταση

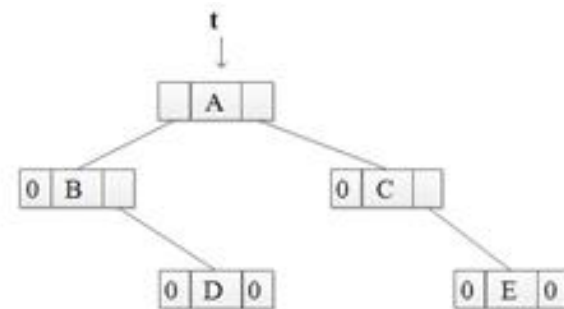
- Χρήση αυτής της αναπαράστασης των δυαδικών δένδρων



(α) Συμπληρωμένο δυαδικό δένδρο



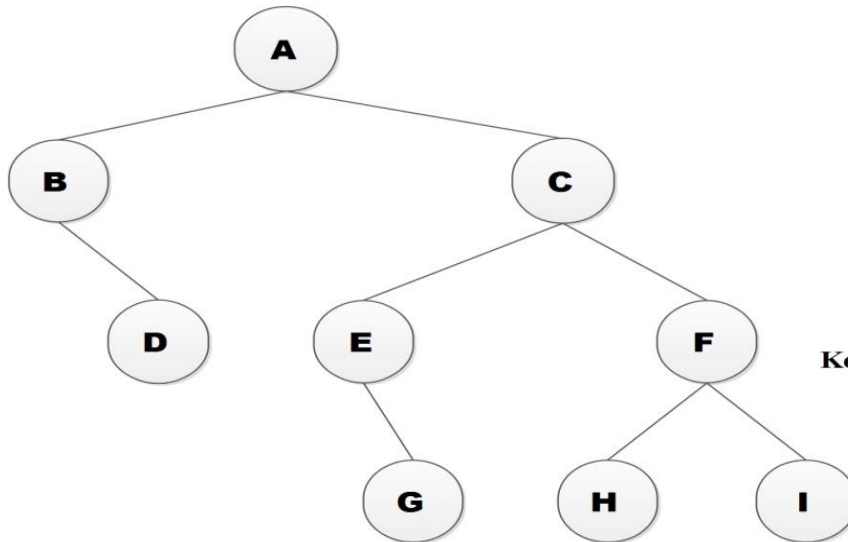
(β) Δεξιά λοξό δυαδικό δένδρο



(γ) Συνδεδεμένη αναπαράσταση

# Διάσχιση Δυαδικού Δένδρου

- ❑ Προ διατεταγμένη διάσχιση (PreOrder)
- ❑ Μετα διατεταγμένη διάσχιση (PostOrder)
- ❑ Ένδο διατεταγμένη διάσχιση (InOrder)
- ❑ Κατά σειρά επιπέδων (LevelOrder)



Προδιατεταγμένη: A, B, D, C, E, G, F, H, I

Μεταδιατεταγμένη: D, B, G, E, H, I, F, C, A

Ενδοδιατεταγμένη: B, D, A, G, E, C, H, F, I

Κατά σειρά επιπέδων: A, B, C, D, E, F, G, H, I

# Βασικές Πράξεις Δυαδικού Δένδρου

---

- ❑ Create: δημιουργία κενού δυαδικού δένδρου
- ❑ IsEmpty: επιστρέφει *true* αν είναι άδειο, αλλιώς *false*
- ❑ Root (*x*): ορίζεται για ρίζα το *x* και επιστρέφει *true* αν επιτύχει, αλλιώς *false*.
- ❑ MakeTree (*element*, *left*, *right*): δημιουργία δυαδικού δένδρου με ρίζα το *element* και παιδιά του τα *left* και *right*
- ❑ BreakTree (*root*, *left*, *right*): το αντίστροφο της *MakeTree*
- ❑ Τέσσερις περιπτώσεις Διάσχισης (*traversal*)





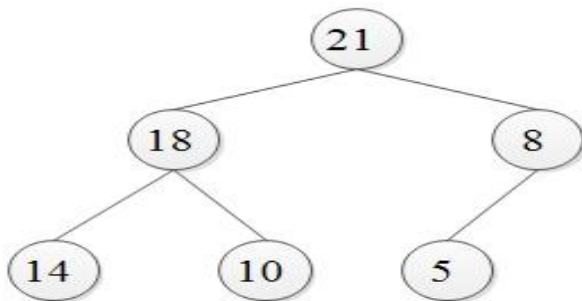
# Ουρές Προτεραιότητας

---

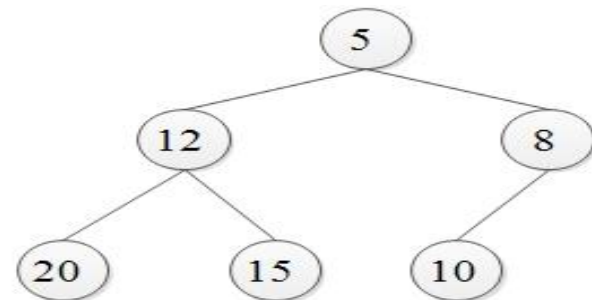
- ❑ Μία ουρά είναι μία δομή δεδομένων με τη μορφή παρατεταμένης συλλογής
- ❑ Μία ουρά προτεραιότητας είναι μία συλλογή από μηδέν ή περισσότερα στοιχεία καθορισμένα από μία προτεραιότητα
- ❑ Λειτουργίες για μία ουρά προτεραιότητας
- ❑ Γραμμικές λίστες
  - Αταξινόμητη λίστα
  - $L_0$ : πρώτο στοιχείο και  $L_{n-1}$  : ουρά, όπου  $L$  μήκος λίστας
- ❑ Σωρός

# Σωρός (Heap)

- Ένα δένδρο μεγίστων/ελαχίστων έχει μεγαλύτερη/μικρότερη ή ίση τιμή στην ρίζα με αυτήν των παιδιών του
- Σωρός είναι ένα δυαδικό δένδρο με ρίζα στο οποίο η τιμή του κάθε κόμβου είναι τουλάχιστον μεγαλύτερη από τις τιμές των παιδιών του
  - Σωροί μεγίστων
  - Σωροί ελαχίστων



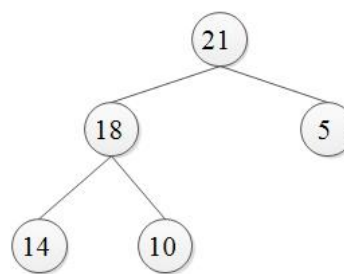
(α) Δένδρο μεγίστων



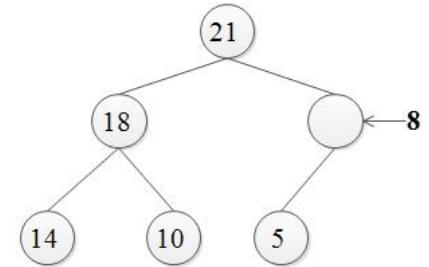
(β) Δένδρο ελαχίστων

# Λειτουργίες Σωρού Μεγίστων

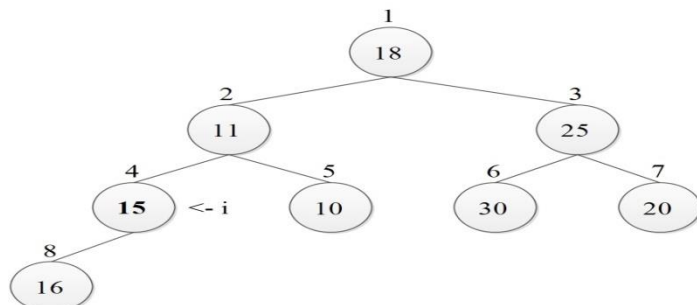
- ❑ Εισαγωγή
- ❑ Διαγραφή
- ❑ Αρχικοποίηση



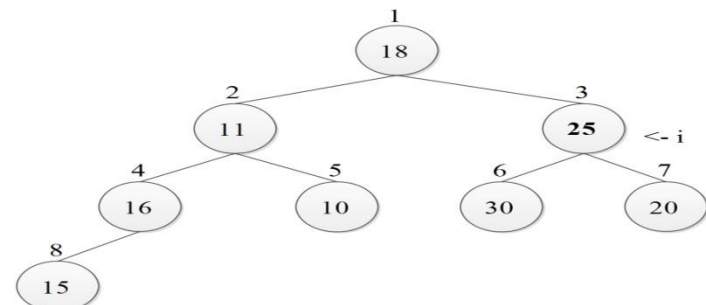
(α)



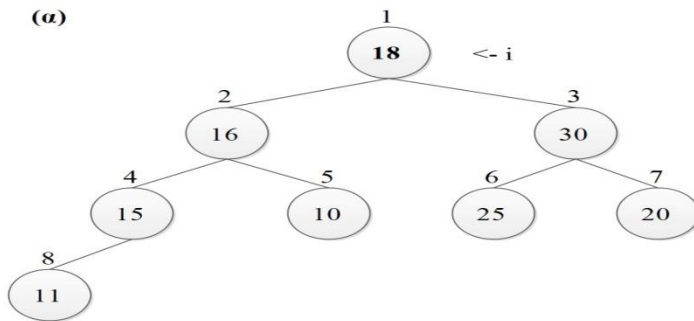
(β)



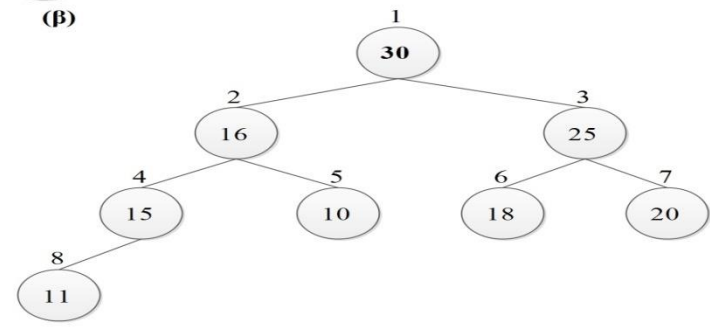
(α)



(β)



(γ)



(δ)

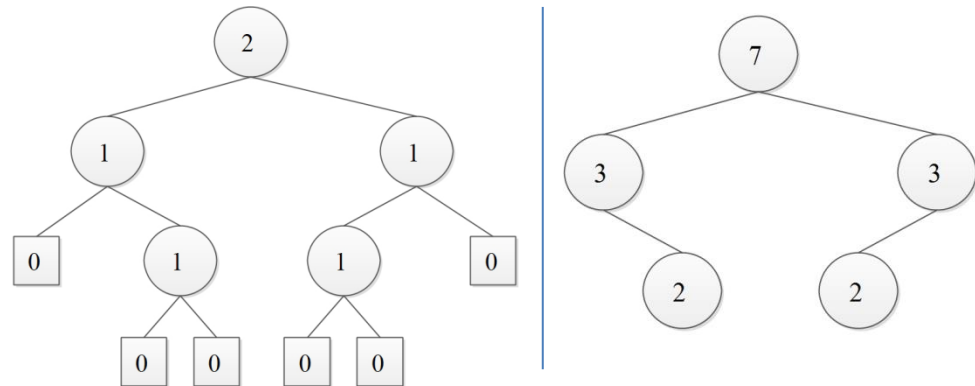
# Αριστερίστικα Δένδρα

Ένας αριστερίστικος σωρός αποτελεί ένα δυαδικό δένδρο διατάξεων σωρού οι κόμβοι του οποίο τηρούν δομικού είδους περιορισμό

□ Αριστερίστικη ιδιότητα

□ Αριστερίστικο δένδρο επηρεασμένο ως προς το ύψος (HBLT)

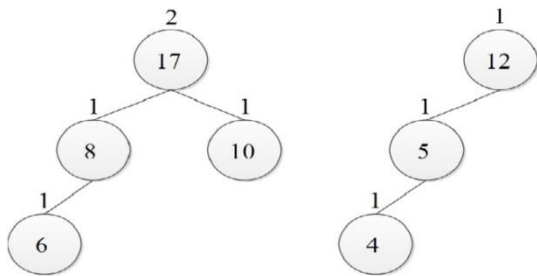
- HBLT μεγίστων
- HBLT ελαχίστων



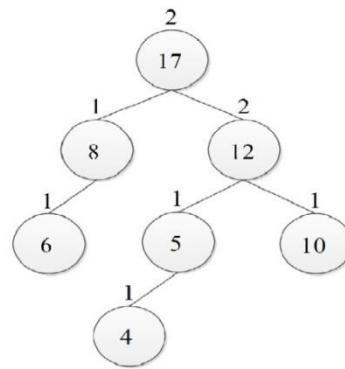
□ Αριστερίστικο δένδρο επηρεασμένο ως προς το βάρος (WBLT)

# Λειτουργίες HBLT

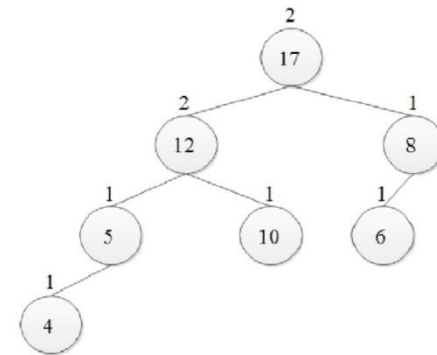
- Εισαγωγή
- Διαγραφή
- Αρχικοποίηση
- Συγχώνευση



(α) Δύο HBLT Μεγίστων



(β) Συγχώνευση HBLT Μεγίστων

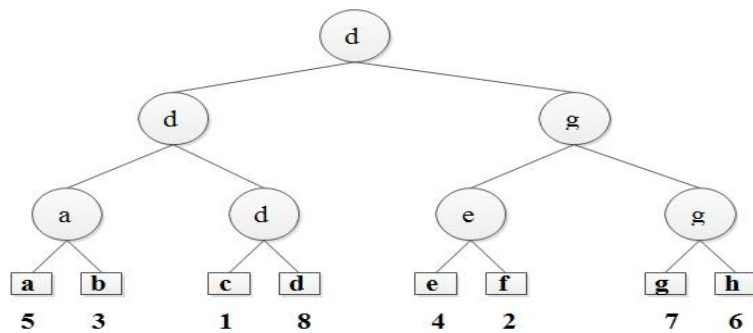


(γ) Διόρθωση των τιμών του HBLT Μεγίστων

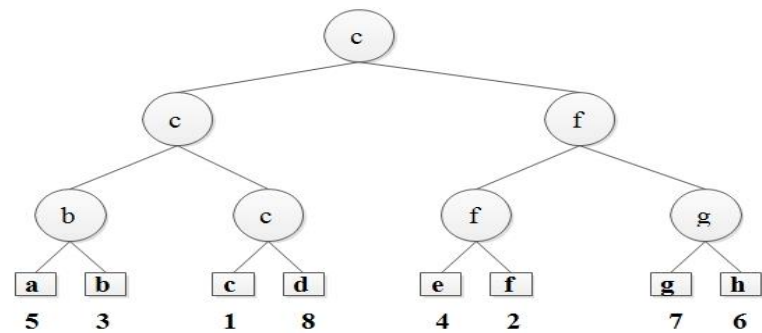


# Δένδρα Τουρνουά

- Είναι ένα πλήρες δυαδικό δένδρο το οποίο αποθηκεύεται αποτελεσματικότερα όταν χρησιμοποιούνται οι πίνακες δυαδικών δένδρων
- Χρησιμοποιούνται στην υλοποίηση του αλγορίθμου για το πρόβλημα της συσκευασίας κιβωτίων
- Δένδρο νικητών μεγίστων - ελαχίστων



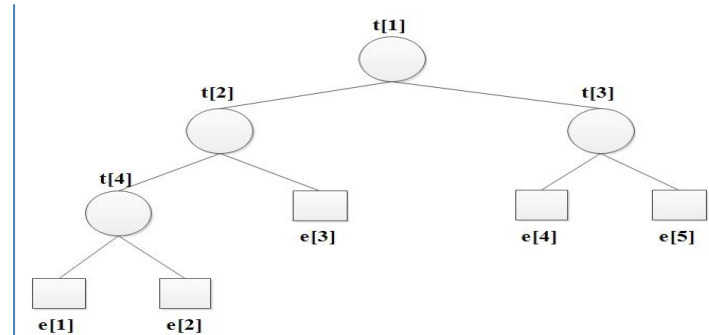
(α) Δένδρο Νικητών Μεγίστων.



(β) Δένδρο Νικητών Ελαχίστων

# Λειτουργίες Δένδρων νικητών μεγίστων

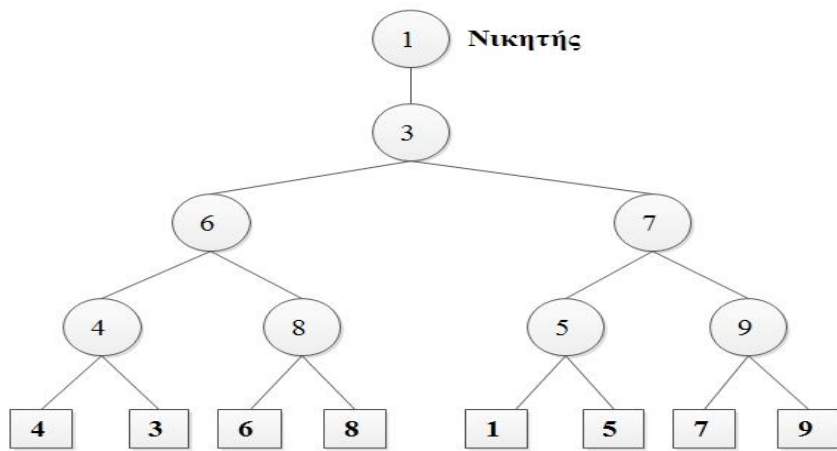
- ❑ Αρχικοποίηση
- ❑ Επιστροφή του νικητή του παιχνιδιού (ρίζα)



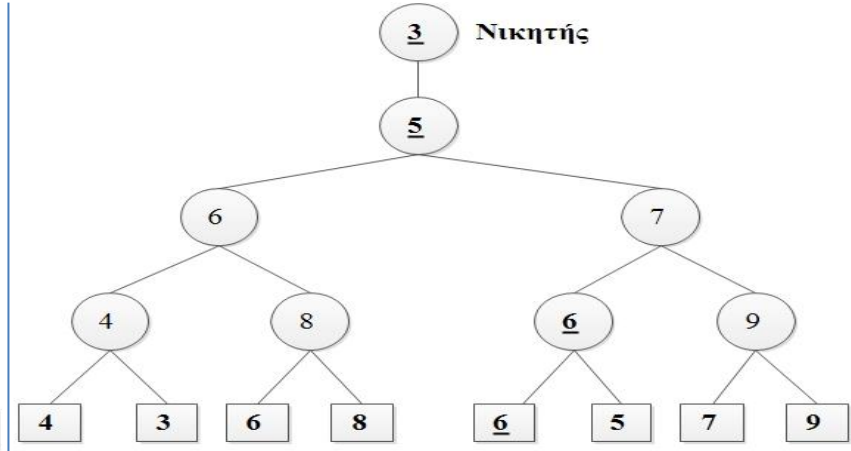
- ❑ Διεξαγωγή παιχνιδιών ξανά μετά από αλλαγή τιμής ενός παίκτη
- ❑ Αναπαράσταση δένδρων νικητών
  - αριστερότερος εσωτερικός κόμβος αριθμείται  $2^s$ , για  $s = \log_2(n-1)$ ,  $n-2^s$
  - ο αριθμός των εξωτερικών κόμβων  $e$  είναι  $LowExt = 2*(n-2^s)$  στο κατώτερο επίπεδο, αλλιώς  $LowExt+1$
  - Αν  $offset=2^{s+1}+1$ , τότε ο πατέρας του  $t[p]$  για κάθε εξωτερικό κόμβο  $e[i]$  προκύπτει από την σχέση  $p = (i+offset)/2$ , για  $i \leq LowExt$ , και  $p = (i - LowExt + n - 1)/2$ , για  $i > LowExt$

# Δένδρα ηττημένων ελαχίστων

- Ελάχιστες διαφορές μεταξύ δένδρων νικητών και των δένδρων ηττημένων
  - Κατά την αρχικοποίηση αντί να γράφεται ο νικητής του κάθε αγώνα γράφεται ο ηττημένος
  - Μόνο σε περίπτωση αλλαγής της τιμής νικητή του παιχνιδιού η διαδικασία της επανάληψης των αγώνων είναι πιο απλή από αυτήν των δένδρων νικητών



(α) Αρχικοποίηση



(β) Επανάληψη αγώνων





# Διαδικά Δένδρα Αναζήτησης (ΔΔΑ)

Τα διαδικά δένδρα αναζήτησης είναι δομές δεδομένων κατάλληλες για το πρόβλημα του λεξικού και για τη διαχείριση ουρών προτεραιότητας γιατί υποστηρίζουν τις λειτουργίες των διαδικών δένδρων

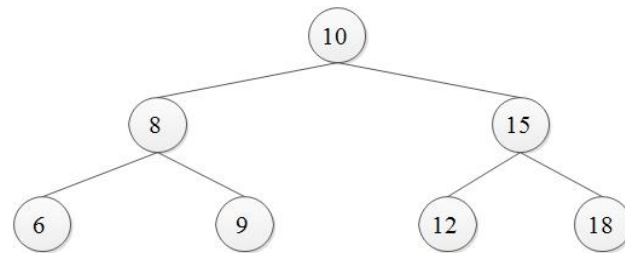
□ Το Πρόβλημα του Λεξικού

□ Βασικές Λειτουργίες ενός ΔΔΑ

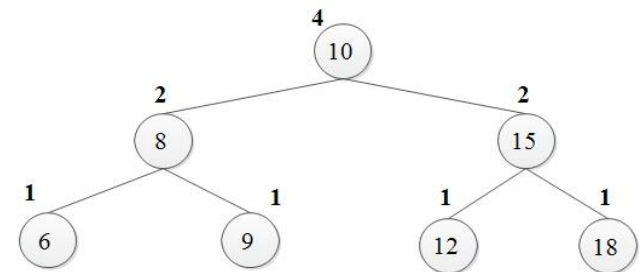
-Αναζήτηση

-Εισαγωγή

-Διαγραφή



(α) Διαδικό Δένδρο Αναζήτησης



(β) Δεικτοδοτούμενο διαδικό δένδρο αναζήτησης

# Περιπτώσεις ισορροπημένων ΔΔΑ

---

## □ AVL Δένδρα Αναζήτησης

- Ένα δυαδικό δένδρο αναζήτησης καλείται δένδρο AVL αν και μόνο αν τα ύψη των δύο υποδένδρων κάθε εσωτερικού κόμβου διαφέρουν το πολύ κατά 1
- 

## □ Κόκκινα-Μαύρα Δένδρα Αναζήτησης

- Ένα κόκκινο-μαύρο (KM) δένδρο είναι ένα δυαδικό δένδρο αναζήτησης στο οποίο όλοι οι κόμβοι του είναι χρωματισμένοι με κόκκινο ή μαύρο χρώμα βάση κανόνων
- 

## □ B-Δένδρα Αναζήτησης

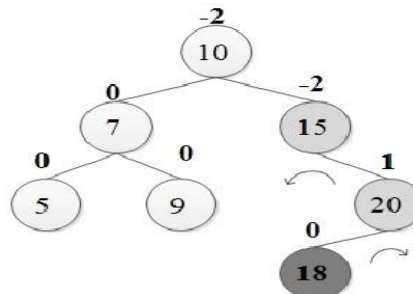
- Τα B-δένδρα είναι ισορροπημένα δένδρα αναζήτησης που έχουν σχεδιαστεί για να λειτουργούν αποδοτικά σε δίσκους ή άλλες συσκευές αποθήκευσης

# AVL Δένδρα Αναζήτησης

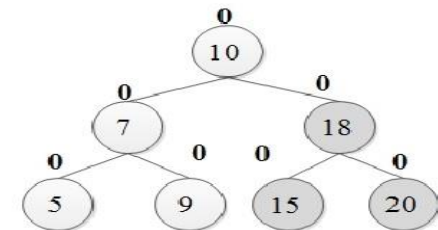
## □ Αναζήτηση

- Η αναζήτηση στοιχείου σε ένα δένδρο AVL γίνεται με τον ίδιο τρόπο, όπως και σε ένα δυαδικό δένδρο αναζήτησης

## □ Εισαγωγή

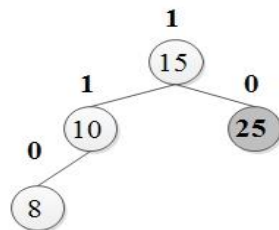


(α) Εισαγωγή κόμβου με τιμή 18

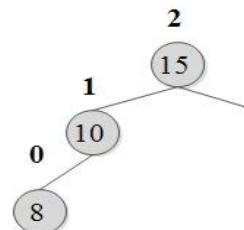


(β) RL περιστροφή

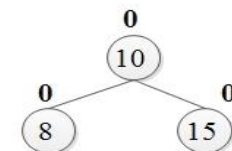
## □ Διαγραφή



(α) Διαγραφή στοιχείου 25



(β) Ανισορροπία δένδρου

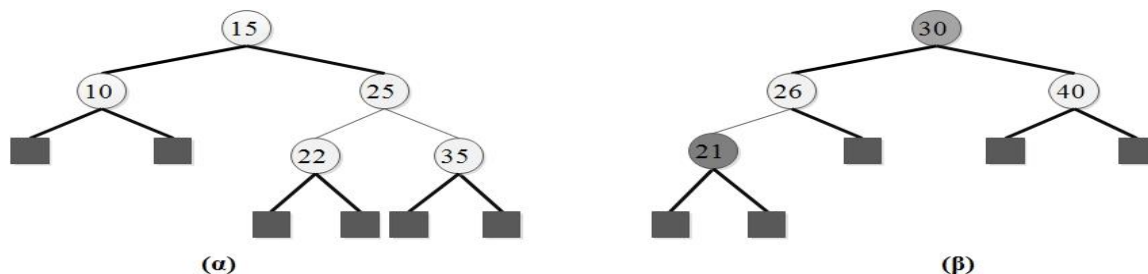


(γ) Δεξιά περιστροφή

# Κόκκινα-Μαύρα Δένδρα Αναζήτησης

Κανόνες χρωματισμού κόμβων και δεικτών:

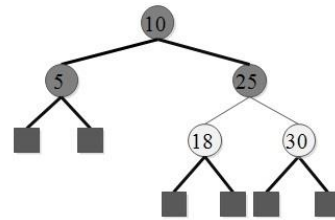
- ❑ Η ρίζα του δένδρου έχει μαύρο χρώμα
- ❑ Όλοι οι εξωτερικοί κόμβοι (φύλλα) έχουν μαύρο χρώμα
- ❑ Δεν υπάρχουν δύο διαδοχικοί κόκκινοι κόμβοι ή δύο διαδοχικοί κόκκινοι δείκτες προς έναν εξωτερικό κόμβο
- ❑ Όλα τα μονοπάτια από τη ρίζα προς τους εξωτερικούς κόμβους έχουν το ίδιο πλήθος μαύρων δεικτών
- ❑ Οι δείκτες από έναν εσωτερικό προς ένα εξωτερικό κόμβο είναι μαύροι



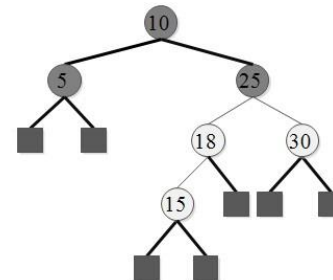
# Λειτουργίες ΚΜ Δένδρου Αναζήτησης

## □ Αναζήτηση

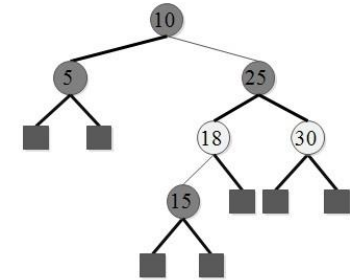
- Η αναζήτηση σε ένα ΚΜ δένδρο γίνεται ακριβώς όπως και σε ένα δυαδικό δένδρο αναζήτησης



(α) Αρχικό κόκκινο-μαύρο δένδρο



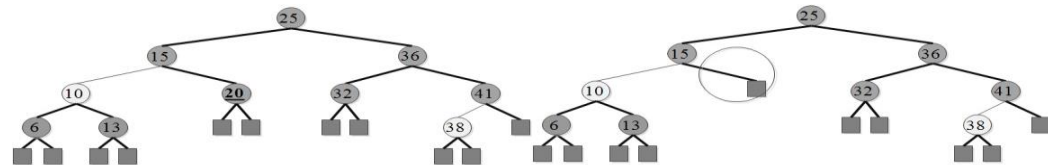
(β) Εισαγωγή στοιχείου 15



(γ) Αντιστροφή χρωμάτων

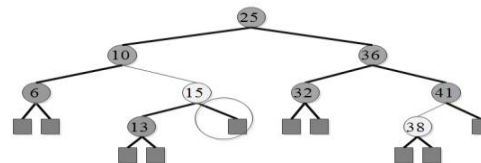
## □ Εισαγωγή

## □ Διαγραφή

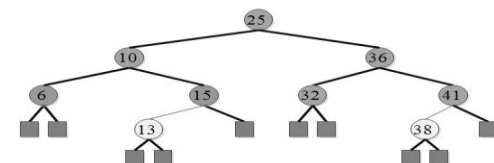


Διαγραφή στοιχείου με κλειδί 20

Παραβίαση μαύρου βάθους



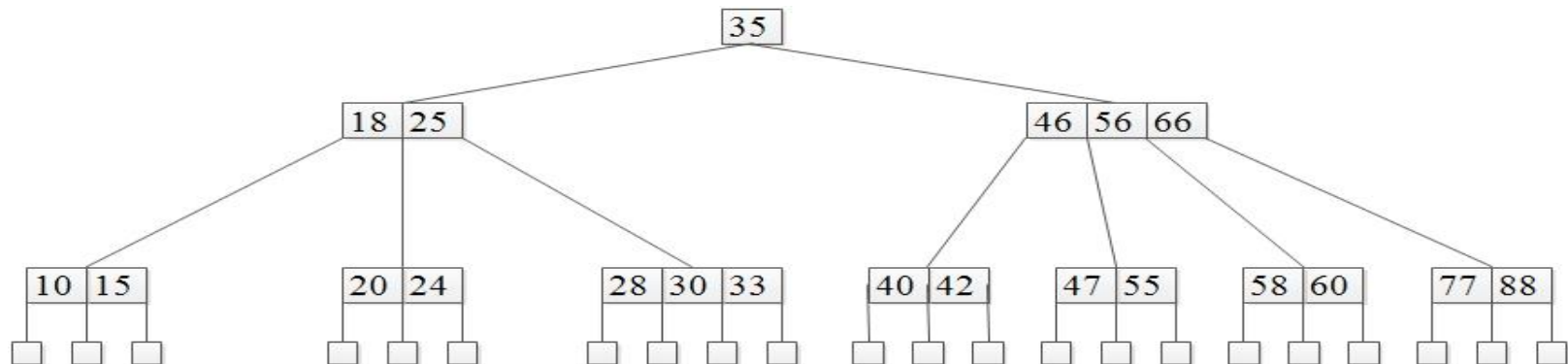
Δεξιά περιστροφή



Αντιστροφή χρωμάτων

# B-Δένδρα Αναζήτησης

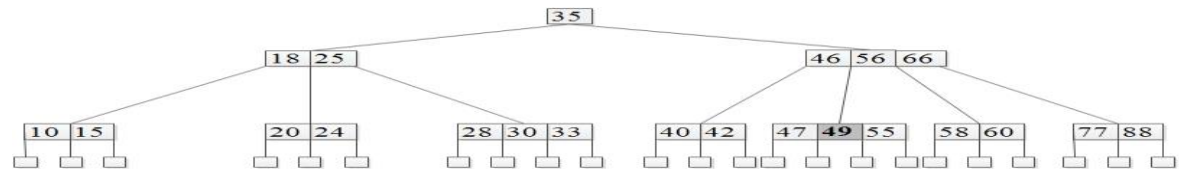
- Είναι παρόμοια με τα κόκκινα-μαύρα δένδρα, αλλά είναι καλύτερα στο να ελαχιστοποιούν τις λειτουργίες I/O ενός δίσκου
- Πολλά συστήματα βάσεων δεδομένων χρησιμοποιούν τα B-δένδρα ή τις παραλλαγές τους για την αποθήκευση των πληροφοριών
- Έστω ότι ένα μη κενό B-δένδρο και  $a$ ,  $\beta$  ακέραιοι ( $a \geq 2$  και  $\beta \geq 2^a$ ). Το δένδρο αυτό ονομάζεται  $(a-\beta)$  δένδρο αν και μόνο αν:
  - Όλα τα φύλλα έχουν το ίδιο βάθος
  - Η ρίζα έχει τουλάχιστον δύο παιδιά και το πολύ  $\beta$
  - Το πλήθος των εσωτερικών κόμβων (εκτός της ρίζας) είναι μεταξύ  $a$  και  $\beta$



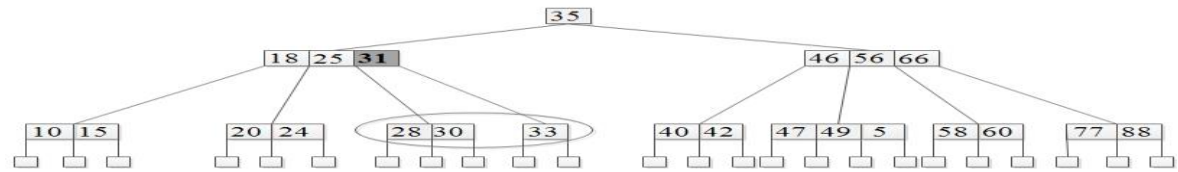
# Λειτουργίες (2, 4)-Δένδρου

## □ Αναζήτηση

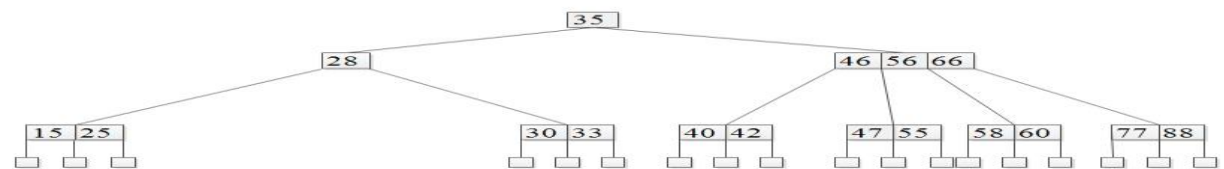
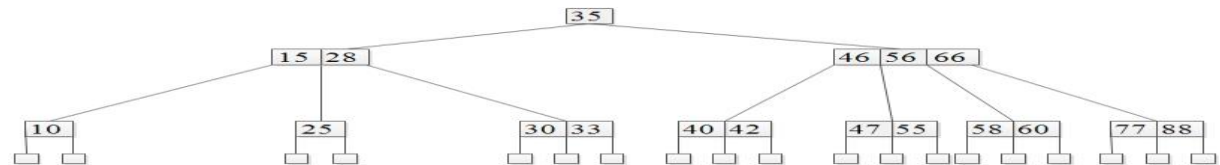
- Η αναζήτηση σε ένα  $(\alpha, \beta)$ -δένδρο γίνεται ακριβώς όπως και σε ένα δυαδικό δένδρο αναζήτησης



## □ Εισαγωγή



## □ Διαγραφή



# ΕΦΑΡΜΟΓΕΣ

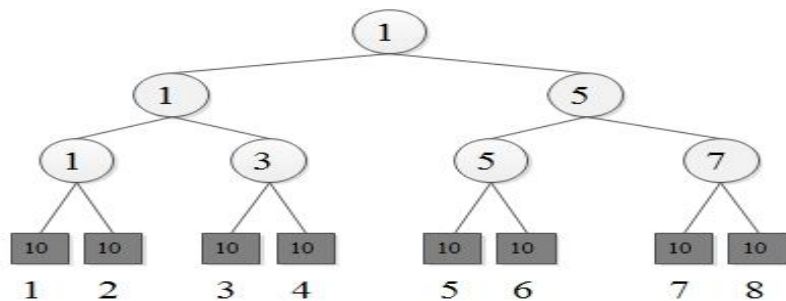
---

- Πρόβλημα Συσκευασίας Κιβωτίων
  - Πρώτη Εφαρμογή (First Fit)
  - Καλύτερη Εφαρμογή (Best Fit)
  - Πρώτη Φθίνουσα Εφαρμογή (First Fit Decreasing)
  - Καλύτερη Φθίνουσα Εφαρμογή (Best Fit Decreasing)
- Πρόβλημα Κατανομής Διασταυρώσεων
  - Ελαχιστοποίηση του πλήθους των διασταυρώσεων.

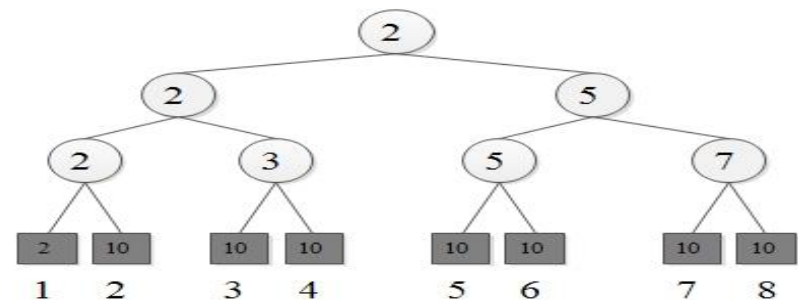


# Χρήση Πρώτης Εφαρμογής

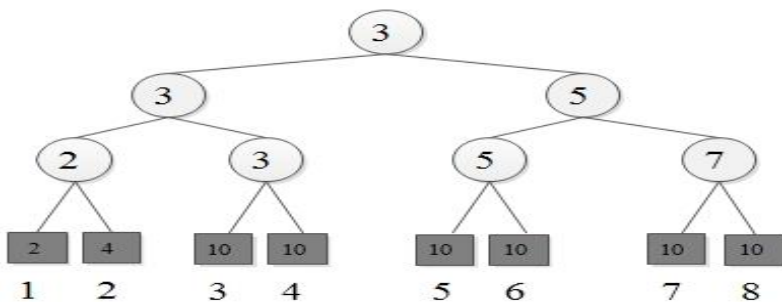
- Υλοποιείται με την χρήση ενός δένδρου νικητών μεγίστων  
-  $s[1] = 8, s[2] = 6, s[3] = 7$



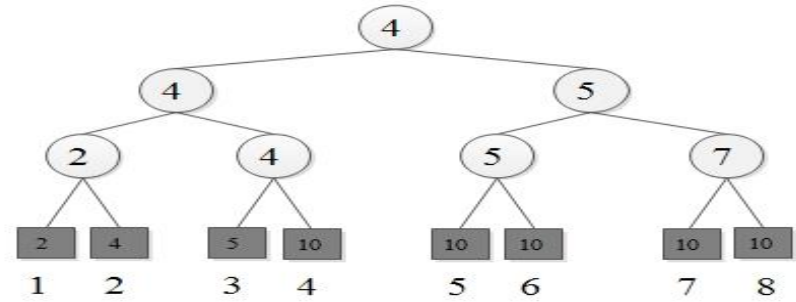
(α) Αρχικό Δένδρο Νικητών



(β) Πακετάρισμα  $s[1]$  στο πρώτο κιβώτιο



(γ) Πακετάρισμα  $s[2]$  στο δεύτερο κιβώτιο

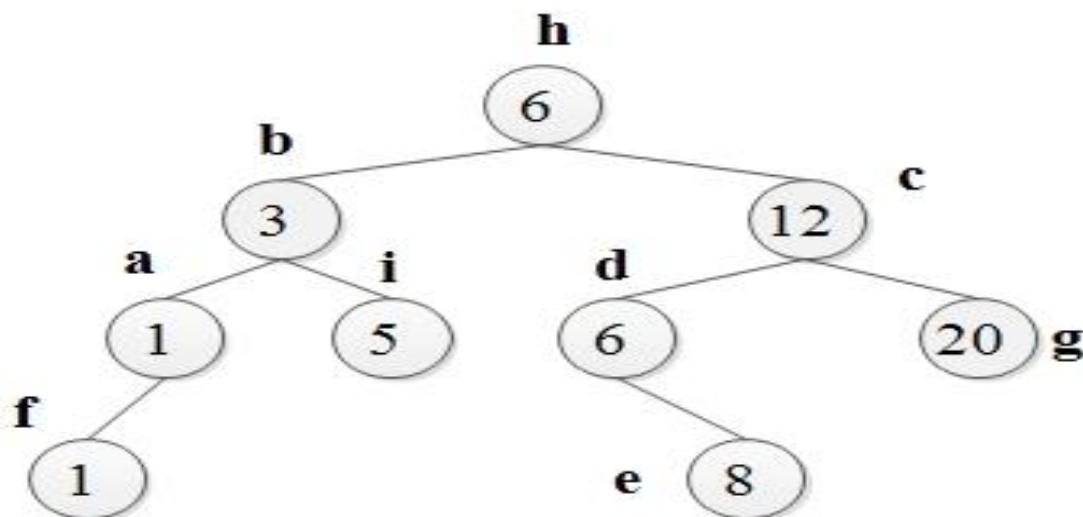


(δ) Πακετάρισμα  $s[3]$  στο τρίτο κιβώτιο

# Χρήση Καλύτερης Εφαρμογής

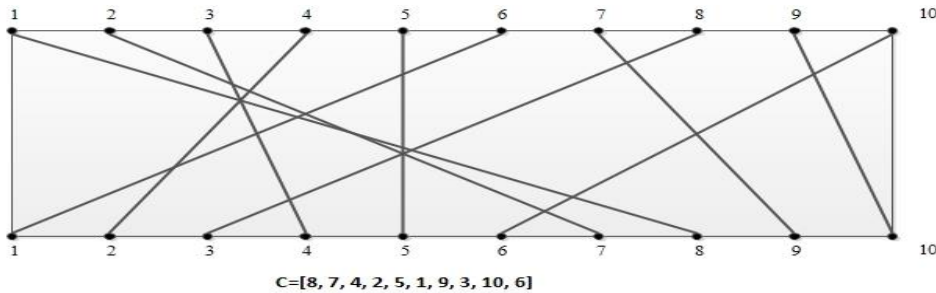
---

- Δεν διαφέρει πολύ από την μέθοδο της Πρώτης Εφαρμογής
- Υλοποιείται με την χρήση δυαδικού δένδρου αναζήτησης με διπλότυπα
  - $s[1] = 4, s[2] = 8, s[3] = 12$

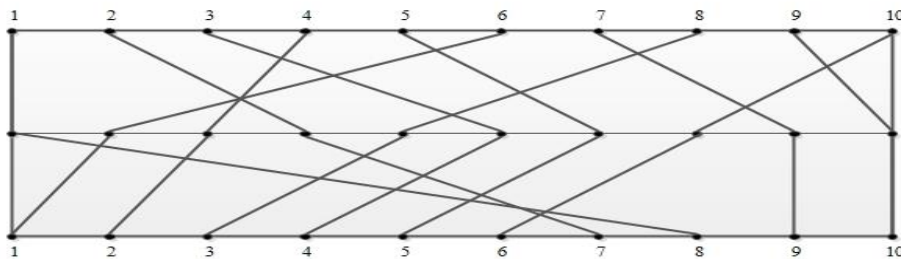


# Κατανομής Διασταυρώσεων

- Υλοποίηση με χρήση γραμμικής λίστας
- Περίπτωση  $C=[8, 7, 4, 2, 5, 1, 9, 3, 10, 6]$   
 -  $X = [1, 6, 4, 2, 8, 3, 5, 10, 7, 9]$



(α) Ένα στιγμιότυπο με καλώδια



(γ) Διάσπαση διασταυρώσεων

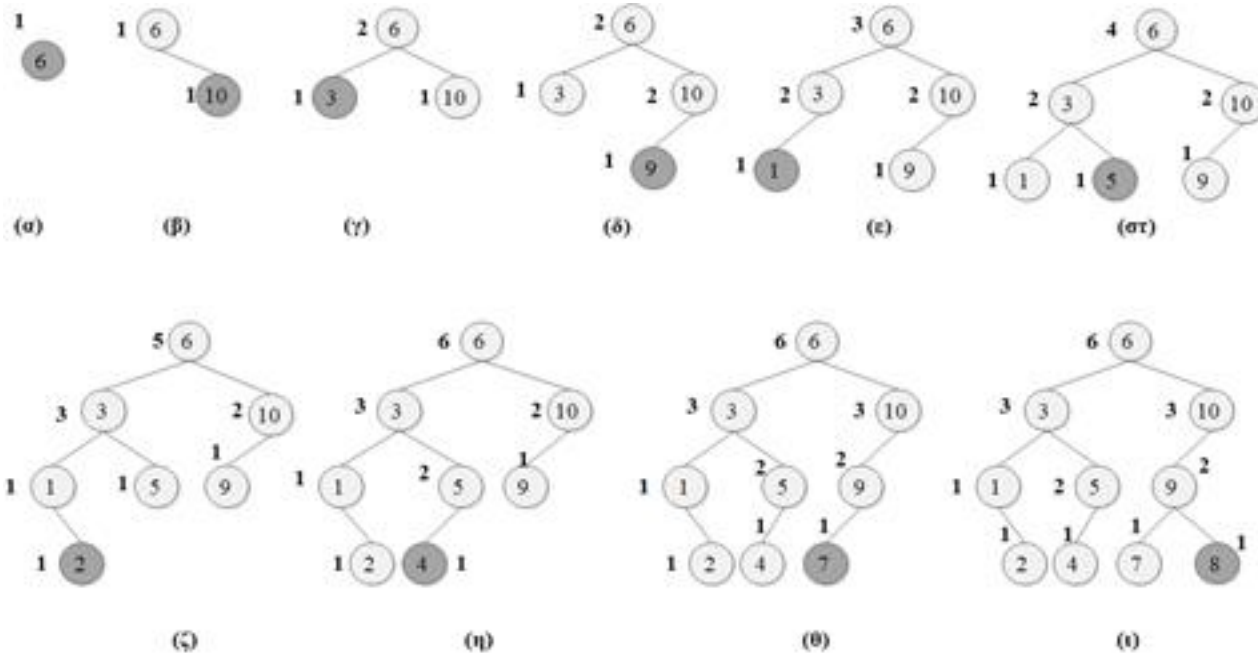
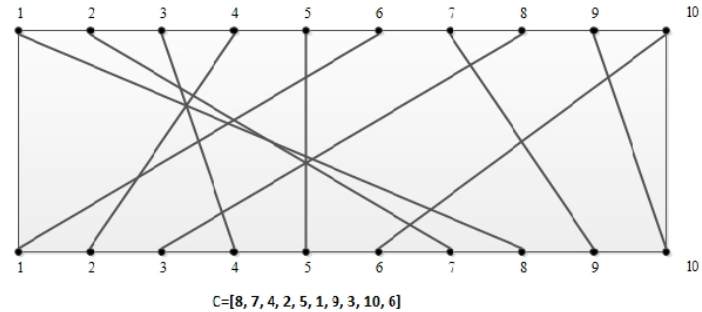
i	$k_i$	Crossings							
1	7	2	3	4	5	6	8	10	
2	6	3	4	5	6	8	10		
3	3	4	6	8					
4	1	6							
5	2	6	7						
6	0								
7	2	8	10						
8	0								
9	1	10							
10	0								

(β) Πίνακας διασταυρώσεων

# Κατανομής Διασταυρώσεων

- Διαδικό δένδρο αναζήτησης με κατάλογο
- Περίπτωση  $C=[8, 7, 4, 2, 5, 1, 9, 3, 10, 6]$

-  $X = [1, 6, 4, 2, 8, 3, 5, 10, 7, 9]$



# Παρουσίαση των Υλοποιήσεων

---

- ❑ Συσκευασία κιβωτίων με χρήση Πρώτης Εφαρμογής
- ❑ Συσκευασία κιβωτίων με χρήση Καλύτερης Εφαρμογής
- ❑ Κατανομή διασταυρώσεων με χρήση γραμμικής λίστας
- ❑ Οπτικοποίηση δυαδικού δένδρου αναζήτησης

# Σύγκριση Πρώτης και Καλύτερης Εφαρμογής

- Για την περίπτωση όπου τα αντικείμενα προς κατανομή είναι  $n = 10$  με χωρητικότητα  $c = 10$   
-  $n = \{5, 7, 6, 8, 9, 1, 4, 3, 5, 10\}$

Πρώτη Εφαρμογή	Καλύτερη Εφαρμογή
Ψάχνει την πρώτη καλύτερη θέση για κάθε αντικείμενο	Ψάχνει την κατάλληλη θέση για κάθε αντικείμενο
Χρήση 7 κιβωτίων	Χρήση 6 κιβωτίων
Χρήση δένδρων νικητών μεγίστων	Χρήση δυαδικού δένδρου αναζήτησης

# Κατανομή διασταυρώσεων

---

- Για  $C: \{0, 8, 7, 4, 2, 5, 1, 9, 3, 10, 6\}$  και  $X = [1, 6, 4, 2, 8, 3, 5, 10, 7, 9]$  προκύπτει:
  - Οι ενώσεις στο κέντρο είναι:  $\{0, 7, 6, 3, 1, 2, 0, 2, 0, 1, 0\}$
  - Οι ενώσεις στο πάνω μέρος είναι:  $\{1, 4, 6, 3, 7, 2, 9, 5, 10, 8\}$
  - Οι ενώσεις στο κάτω μέρος είναι:  $\{8, 1, 2, 7, 3, 4, 5, 6, 9, 10\}$



Ευχαριστώ πολύ  
για την  
προσοχή σας !

