

# Δέντρα

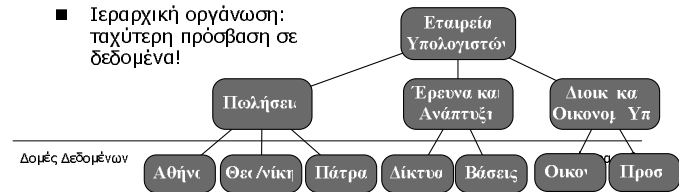
Δημήτρης Φωτάκης

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Πανεπιστήμιο Αιγαίου

# Δέντρα

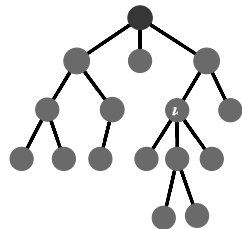
- **Δέντρο:** μοντέλο ιεραρχικής δομής.
  - Αναπαράσταση (ιεραρχικών) σχέσεων: προγόνου-απογόνου, προϊσταμένου-υφισταμένου, όλου-μέρους, ...
- Εφαρμογές:
  - Γενεαλογικά δέντρα.
  - Οργανόγραμμα επιχείρησης, ιεραρχία διοίκησης.
  - User interfaces, web sites, module hierarchy, δέντρα απόφασης, ...
  - Ιεραρχική οργάνωση: ταχύτερη πρόσβαση σε δεδομένα!



Δομές Δεδομένων

# Δέντρα: Ορολογία

- Γράφημα **ακυκλικό** και **συνεκτικό**.
- Δέντρο με **n κορυφές** έχει **m = n - 1 ακμές**.
- **Ρίζα** : κόμβος χωρίς πρόγονο.
  - Δέντρο με **ρίζα** : **ιεραρχία**
- **Φύλλο** : κόμβος χωρίς απογόνους.
- **Πρόγονοι u** : κόμβοι στο (μοναδικό) μονοπάτι *u* προς ρίζα.
- **Απόγονοι u** : κόμβοι σε μονοπάτια από *u* προς φύλλα.
- **Υποδέντρο u** : Δέντρο αποτελούμενο από *u* και απογόνους του.

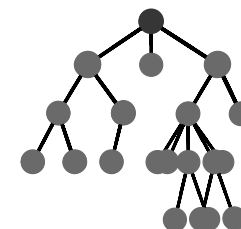


Δομές Δεδομένων

Δέντρα 3

# Δέντρα: Ορολογία

- **Επίπεδο u** : μήκος μονοπατιού από *u* προς ρίζα.
- **Ύψος** : μέγιστο επίπεδο κόμβου (φύλλου).
  - Μέγιστη απόσταση από ρίζα.
- **Βαθμός u** : αριθμός παιδιών *u*.
- **Δυαδικό δέντρο** : κάθε κορυφή  $\leq 2$  παιδιά
  - Αριστερό και δεξιά.
- Κάθε **υποδέντρο** είναι δυαδικό δέντρο.



Δομές Δεδομένων

Δέντρα 4

## Διαδικά Δέντρα

### □ Ύψος $h$ :

$$h+1 \leq \# \text{κορυφών} \leq 2^{h+1} - 1$$

- $h+1$  επίπεδα,  $\geq 1$  κορ. / επίπ.

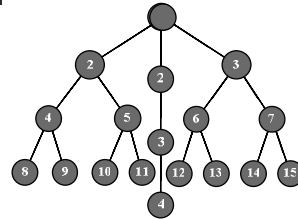
- $\leq 2^i$  κορυφές στο επίπεδο  $i$ .

$$1 + 2 + \dots + 2^h = 2^{h+1} - 1$$

### □ #κορυφών $n$ :

$$\log_2(n+1) - 1 \leq h \leq n - 1$$

### □ Πλήρες (full) : $n = 2^{h+1} - 1$



Δομές Δεδομένων

Δέντρα 5

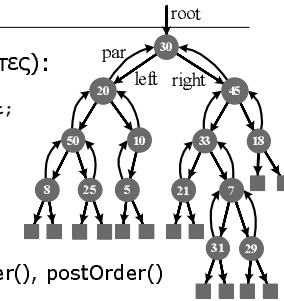
## Διαδικά Δέντρα

### □ Δυναμική αναπαράσταση (δείκτες):

```
typedef struct _tnode {
    struct _tnode *left, *right;
    struct _tnode *par;
    int key;
} tnode;
tnode root = NULL;
```

### □ Λειτουργίες:

- Διελεύσεις: inOrder(), preOrder(), postOrder()
- Μετρήσεις: height(), size()
- Βοηθητικές: createNode(), isEmpty(), ...



Δομές Δεδομένων

Δέντρα 6

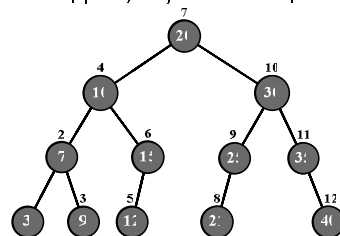
## Inorder

### □ Ενδο-διατεταγμένη (inorder) διέλευση:

- Αριστερό - Ρίζα - Δεξί.
- Κόμβος εξετάζεται μετά από κόμβους αριστερού υποδέντρου και πριν από κόμβους δεξιού υποδέντρου.

```
void inOrder(tnode *x) {
    if (x == NULL) return;
    inOrder(x->left);
    printf("%d ", x->key);
    inOrder(x->right);
}
```

inOrder(root);



Δομές Δεδομένων

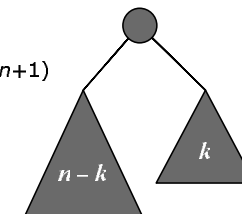
Δέντρα 7

## Inorder: Χρονική Πολυπλοκότητα

### □ Inorder σε δέντρο με $n$ στοιχεία : χρόνος $\Theta(n)$

### □ Απόδειξη με επαγωγή:

- Χρόνο  $\Theta(1)$  σε δέντρο με 1 στοιχείο.
- Επαγωγική υπόθεση για δέντρα με  $\leq n$  στοιχεία.
- Δέντρο με  $n+1$  στοιχεία:
  - Χρόνος:  $\Theta(n-k) + \Theta(1) + \Theta(k) = \Theta(n+1)$



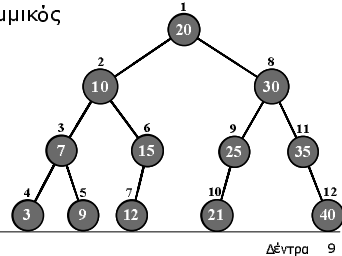
Δομές Δεδομένων

Δέντρα 8

## Preorder

- Προ-διατεταγμένη (preorder) διέλευση:
  - Ρίζα – Αριστερό – Δεξί.
  - Κόμβος εξετάζεται πριν από κόμβους αριστερού και δεξιού υποδέντρου.
  - Χρόνος :  $\Theta(n)$  – γραμμικός

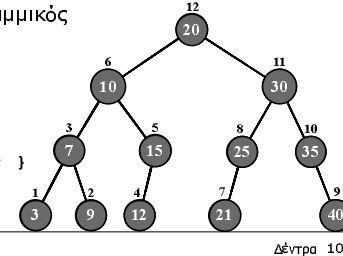
```
void preOrder(tnode *x) {
    if (x == NULL) return;
    printf(" %d ", x->key);
    preOrder(x->left);
    preOrder(x->right); }
preOrder(root);
```



## Postorder

- Μετα-διατεταγμένη (preorder) διέλευση:
  - Αριστερό – Δεξί – Ρίζα
  - Κόμβος εξετάζεται μετά από κόμβους αριστερού και δεξιού υποδέντρου.
  - Χρόνος :  $\Theta(n)$  – γραμμικός

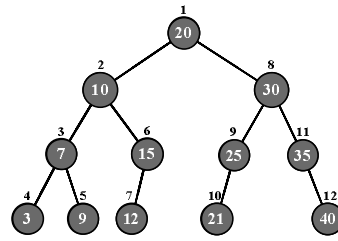
```
void postOrder(tnode *x) {
    if (x == NULL) return;
    postOrder(x->left);
    postOrder(x->right);
    printf(" %d ", x->key); }
postOrder(root);
```



## Μέγεθος

- size(): μέτρηση #κόμβων.
  - Τροποποίηση διέλευσης σε γραμμικό χρόνο.

```
int cnt;
int size() {
    cnt = 0;
    preOrderCount(root);
    return (cnt); }
void preOrderCount(tnode *x) {
    if (x == NULL) return;
    cnt++;
    preOrderCount(x->left);
    preOrderCount(x->right); }
```



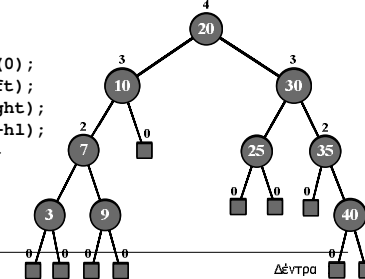
## Ύψος

- height(): μέτρηση ύψους.

$$h(u) = \begin{cases} 0 & \text{αν } u \text{ NULL-φύλλο} \\ 1 + \max\{h(u \rightarrow \text{left}), h(u \rightarrow \text{right})\} & \text{διαφορετικά} \end{cases}$$

```
int height(tnode *x) {
    if (x == NULL) return(0);
    int hl = height(x->left);
    int hr = height(x->right);
    if (hl > hr) return(++hl);
    else return(++hr); }
```

h = height(root);

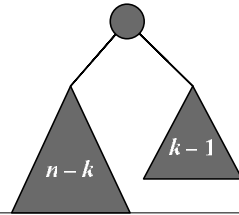


## Ύψος: Χρόνος Εκτέλεσης

- $T(n)$  : χρόνος εκτέλεσης  $\text{height}()$  για δέντρο με  $n$  στοιχεία.

$$T(n) = T(n - k) + T(k - 1) + \Theta(1)$$

- Γραμμικός χρόνος  $\Theta(n)$ .



Δομές Δεδομένων

Δέντρα 13

## Πρόβλημα (ΑΤΔ) Λεξικού

- Δυναμικά μεταβαλλόμενη συλλογή αντικειμένων που αναζητούνται με κλειδί (π.χ. κατάλογοι).
- **Λεξικό** : συλλογή αντικειμένων με μοναδικό κλειδί.
  - Κλειδί: αριθμός ή τύπος δεδομένων με ολική διάταξη.
  - Γενίκευση και για μη-μοναδικά κλειδιά.
- Λεξικό υποστηρίζει λειτουργίες:
  - Αναζήτηση στοιχείου με κλειδί  $x$
  - Εισαγωγή στοιχείου με κλειδί  $x$
  - Διαγραφή στοιχείου με κλειδί  $x$

Δομές Δεδομένων

Δέντρα 14

## Λειτουργίες Λεξικού

- Λεξικό υποστηρίζει λειτουργίες:
  - Αναζήτηση/εισαγωγή/διαγραφή στοιχείου με κλειδί  $x$
  - Εκτύπωση στοιχείων σε αύξουσα / φθίνουσα σειρά
  - Προηγούμενο και επόμενο στοιχείο.
  - Μέγιστο και ελάχιστο στοιχείο.
  - $k$ -οστό μικρότερο στοιχείο
  - Βοηθητικές λειτουργίες ...

Δομές Δεδομένων

Δέντρα 15

## Υλοποιήσεις Λεξικού

- Μη-ταξινομημένη διασυνδεδεμένη λίστα:
  - Εισαγωγή:  $O(1)$
  - Αναζήτηση / διαγραφή:  $O(n)$
  - Κατάλληλη όταν συχνές εισαγωγές, σπάνιες αναζητήσεις / μεμονωμένες διαγραφές (π.χ.  $\log$  file).
- Ταξινομημένος πίνακας:
  - (Δυαδική) αναζήτηση:  $O(\log n)$
  - Στατική συλλογή : «εισαγωγή»  $O(\log n)$  / στοιχείο Χρόνος ταξινόμησης :  $O(n \log n)$
  - Δυναμική συλλογή : εισαγωγή / διαγραφή  $O(n)$
  - Κατάλληλη όταν συχνές αναζητήσεις και δεδομένα μεταβάλλονται σπάνια (π.χ. αγγλο-ελληνικό λεξικό).

Δομές Δεδομένων

Δέντρα 16

## Υλοποιήσεις Λεξικού

---

- Ζυγισμένο (Διαδικό) Δέντρο Αναζήτησης:
  - Αναζήτηση / εισαγωγή / διαγραφή:  $O(\log n)$
  - Μέγιστο / ελάχιστο / προηγούμενο / επόμενο /  $k$ -οστό:  $O(\log n)$
  - Range queries σε γραμμικό χρόνο.
  - Πλήρως δυναμική – επιπλέον χώρος για δείκτες!
- Πίνακας Κατακερματισμού:
  - Αναζήτηση / διαγραφή :  $O(1)$
  - Εισαγωγή :  $O(1)$  expected amortized,  $O(\log n)$  whp,  $O(n)$  χ.π.
  - Δεν υποστηρίζει αποδοτικά άλλες λειτουργίες.
  - Δυναμική – επιπλέον χώρος στον πίνακα (util  $\approx 50\%$ )

Δομές Δεδομένων

Δέντρα 17