



Αλγόριθμοι και Δομές Δεδομένων(Θ)

Ενότητα 12: ΔΕΝΔΡΑ (2)

Ευάγγελος Γ. Ούτσιος

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Κεντρικής Μακεδονίας» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



ΔΕΝΔΡΑ

ΔΥΑΔΙΚΑ ΔΕΝΔΡΑ (*Binary trees*)

Ορισμός

Ένα δυαδικό δένδρο αποτελείται από ένα πεπερασμένο σύνολο κόμβων. Το δένδρο είναι είτε άδειο, είτε αποτελείται από δύο άλλα δυαδικά δένδρα που ονομάζονται αριστερό και δεξιό υποδένδρο.

Ορισμός στην C:

```
struct treenode
{
    char data;
    struct treenode * left;
    struct treenode * right;
};
typedef struct treenode * PTR;
```

ΔΕΝΔΡΑ

Μία σπουδαία λειτουργία σε ένα δένδρο είναι η διάσχισή του (traversal) ή διέλευση, η επίσκεψη δηλ. όλων των κόμβων του μία φορά. Υπάρχουν τρεις διαφορετικές μέθοδοι διάσχισης:

- α) **Προδιατεταγμένη** μέθοδος (*preorder traversal*)
- β) **Ενδοδιατεταγμένη** μέθοδος (*inorder traversal*)
- γ) **Μεταδιατεταγμένη** μέθοδος (*postorder traversal*)

ΔΕΝΔΡΑ

preorder

1. Επίσκεψη της ρίζας
2. Επίσκεψη του αριστερού υποδένδρου
3. Επίσκεψη του δεξιού υποδένδρου

inorder

1. Επίσκεψη του αριστερού υποδένδρου
2. Επίσκεψη της ρίζας
3. Επίσκεψη του δεξιού υποδένδρου

postorder

1. Επίσκεψη του αριστερού υποδένδρου
2. Επίσκεψη του δεξιού υποδένδρου
3. Επίσκεψη της ρίζας

ΔΕΝΔΡΑ

Υλοποίηση στην C:

```
void preorder_traversal(PTR t)
{
    if (t!=NULL)
    {
        printf("%c ",t->data);
        preorder_traversal(t->left);
        preorder_traversal(t->right);
    }
}

void inorder_traversal(PTR t)
{
    if (t!=NULL)
    {
        inorder_traversal(t->left);
        printf("%c ",t->data);
        inorder_traversal(t->right);
    }
}
```

```
void postorder_traversal(PTR t)
{
    if (t!=NULL)
    {
        postorder_traversal(t->left);
        postorder_traversal(t->right);
        printf("%c ",t->data);
    }
}
```

ΔΕΝΔΡΑ

Εισαγωγή κόμβου σε Δυαδικό Δένδρο Αναζήτησης:

```
void insert_node(PTR *pt, char x)
{
    PTR t;
    t = *pt;
    if (t == NULL)
    {
        t = malloc(sizeof (struct treenode));
        t->data = x;
        t->left = NULL;
        t->right = NULL;
    }
    else
        if (x < t->data)
            insert_node(&(t->left), x);
        else
            insert_node(&(t->right), x);
    *pt = t;
}
```


ΔΕΝΔΡΑ

Διαγραφή κόμβου από Δυαδικό Δένδρο Αναζήτησης:

Η διαδικασία της διαγραφής είναι πιο σύνθετη από τη διαδικασία της εισαγωγής.

Αν ο κόμβος είναι τερματικός, τότε είναι εύκολο. Επίσης εύκολη είναι η περίπτωση που ο διαγραφόμενος κόμβος έχει μόνον ένα απόγονο.

Δύσκολη είναι η περίπτωση που ο κόμβος έχει δύο απογόνους. Σ' αυτή την περίπτωση ο κόμβος πρέπει να αντικατασταθεί είτε από τον πιο δεξιό κόμβο του αριστερού υποδένδρου, ή από τον πιο αριστερό κόμβο του δεξιού υποδένδρου.

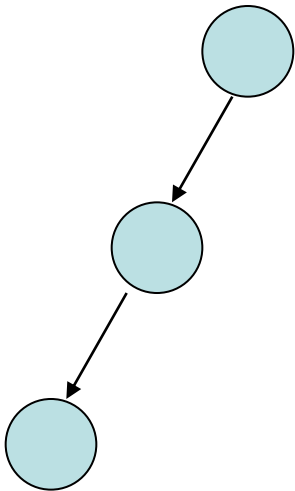
ΔΕΝΔΡΑ

Ισορροπημένα δένδρα

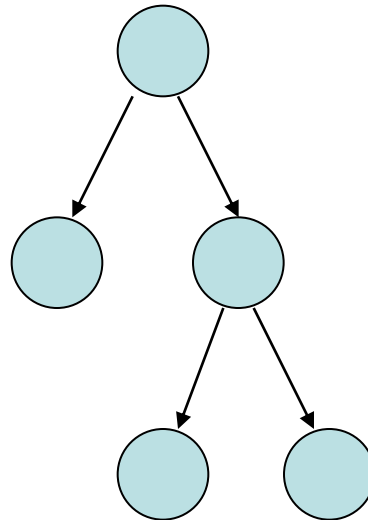
- Ένα δυαδικό δένδρο λέγεται **ισορροπημένο κατά ύψος** (height balanced tree) αν το ύψος του αριστερού και του δεξιού υποδένδρου κάθε κόμβου διαφέρει το πολύ κατά ένα.
- Ένα δυαδικό δένδρο λέγεται **τέλεια ισορροπημένο** (perfectly balanced tree) αν το πλήθος των κόμβων του αριστερού και του δεξιού υποδένδρου κάθε κόμβου διαφέρει το πολύ κατά ένα.

ΔΕΝΔΡΑ

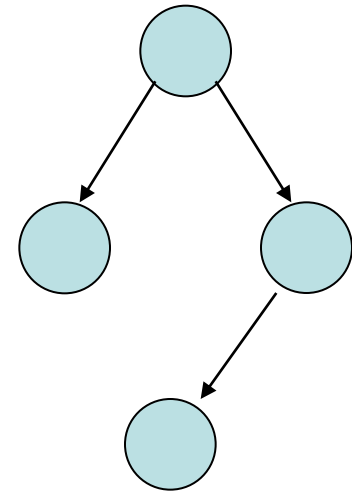
Π.χ.



α) Εκφυλισμένο δένδρο



β) Ισορροπημένο κατά ύψος δένδρο



γ) Τέλεια ισορροπημένο δένδρο