

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΕΣ ΤΕΧΝΙΚΕΣ

<http://www.softlab.ntua.gr/~nickie/Courses/progtech/>

Διδάσκοντες: Γιάννης Μαΐστρος (maistros@cs.ntua.gr)
Στάθης Ζάχος (zachos@cs.ntua.gr)
Νίκος Παπασπύρου (nickie@softlab.ntua.gr)

Διαφάνειες παρουσίασης #8 (α)

- ✓ Δυαδικά δένδρα
- ✓ Ισοζυγισμένα δυαδικά δένδρα

◆ Τύπος κόμβου TreeNode

```
typedef struct TreeNode_tag {  
    int data;  
    struct TreeNode_tag * left;  
    struct TreeNode_tag * right;  
} TreeNode;
```

◆ Τύπος tree

```
typedef TreeNode * tree;
```

◆ Κενό δένδρο

```
const tree treeEmpty = NULL;
```


◆ Εισαγωγή σε δένδρα

- Καθοριστική απόφαση: σε ποιο σημείο του δένδρου θα εισαχθεί ο νέος κόμβος
- **Ισοζυγισμένα δένδρα** (balanced trees): το βάθος δυο φύλλων διαφέρει το πολύ κατά 1

◆ Συνάρτηση μέγιστου βάθους

```
int treeDepth (tree t)
{
    if (t == NULL) return 0;
    return 1 + max(treeDepth(t->left),
                   treeDepth(t->right));
}
```


◆ Εισαγωγή σε ισοζυγισμένα δένδρα

```
void treeBalancedInsert (tree * t, int d)
{
    if (*t == NULL) {
        *t = (TreeNode *)
            malloc(sizeof(TreeNode));

        if (*t == NULL) {
            printf("Out of memory\n");
            exit(1);
        }

        (*t)->data = d;
        (*t)->left = (*t)->right = NULL;
    }
}
```


◆ Εισαγωγή σε ισοζυγισμένα δένδρα (συνέχεια)

```
else {  
    int d1 = treeDepth((*t)->left);  
    int d2 = treeDepth((*t)->right);  
  
    if (d1 <= d2)  
        treeBalancedInsert(  
            &((*t)->left), d);  
    else  
        treeBalancedInsert(  
            &((*t)->right), d);  
}  
}
```