ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΕΣ ΤΕΧΝΙΚΕΣ

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

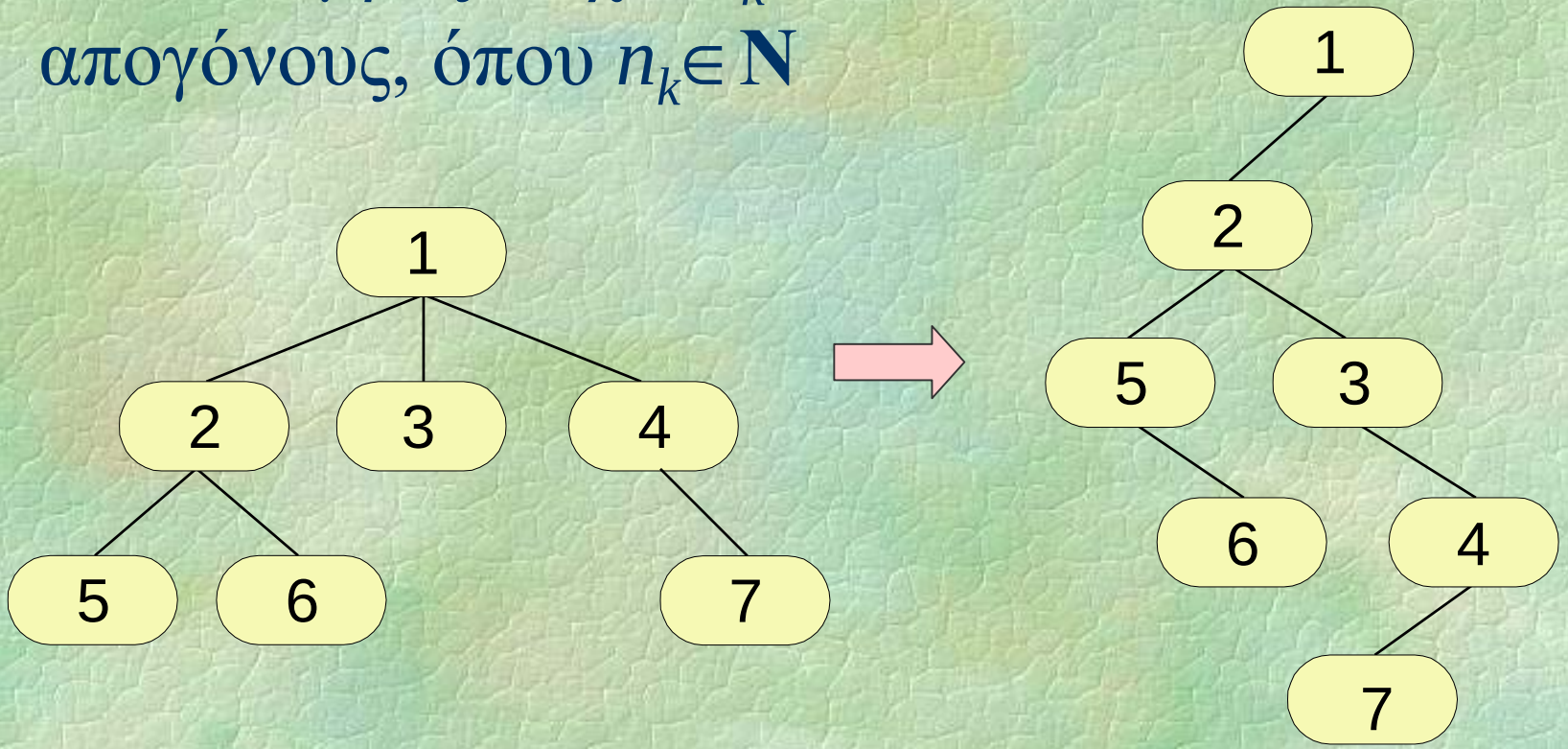
Διδάσκοντες: Γιάννης Μαΐστρος (maistros@cs.ntua.gr)
Στάθης Ζάχος (zachos@cs.ntua.gr)
Νίκος Παπασπύρου (nickie@softlab.ntua.gr)

Διαφάνειες παρουσίασης #8 (β)

- ✓Δένδρα γενικής μορφής
- ✓Διάσχιση δυαδικών δένδρων
- ✓Αριθμητικές εκφράσεις
- ✓Δυαδικά δένδρα αναζήτησης

Δένδρα γενικής μορφής

- Κάθε κόμβος k έχει n_k απογόνους, όπου $n_k \in \mathbf{N}$



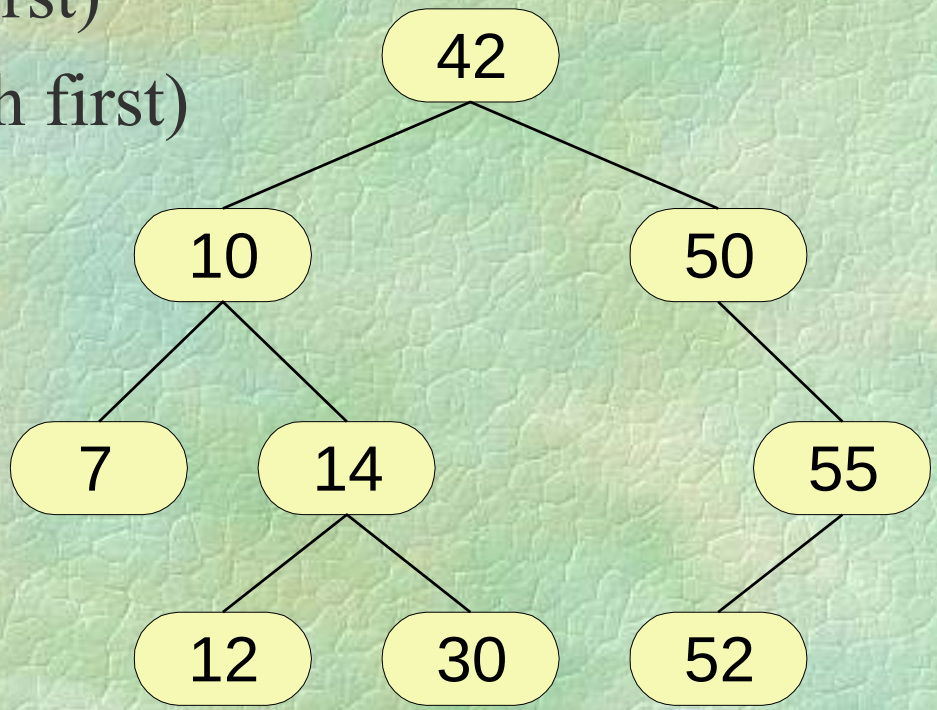
- Κωδικοποίηση με δυαδικά δένδρα

Διάσχιση δυαδικών δένδρων (i)

- Σειρά με την οποία διασχίζονται οι κόμβοι
  - κατά βάθος (depth first)
  - κατά πλάτος (breadth first)
- Κατά βάθος

  42, 10, 7, 14, 12, 30, 50, 55, 52
- Κατά πλάτος

  42, 10, 50, 7, 14, 55, 12, 30, 52



Διάσχιση δυαδικών δένδρων (ii)

- Εκτύπωση κατά βάθος
  - πολύ απλά, με χρήση αναδρομής
- Υλοποίηση

```
void treePrintDF (tree t)
{
   if (t != NULL) {
      printf("%d ", t->data);
      treePrintDF(t->left);
      treePrintDF(t->right);
   }
}
```

Διάσχιση δυαδικών δένδρων (iii)

- Εκτύπωση κατά πλάτος
  - με τη βοήθεια ουράς για την αποθήκευση δεικτών προς τους κόμβους που δεν έχουμε επισκεφθεί
- Υλοποίηση

```
void treePrintBF (tree t)
{
   queue q = queueEmpty;

   if (t != NULL)
      queueInsert(&q, t);
```

Διάσχιση δυαδικών δένδρων (iv)

- Εκτύπωση κατά πλάτος, υλοποίηση (συνέχεια)

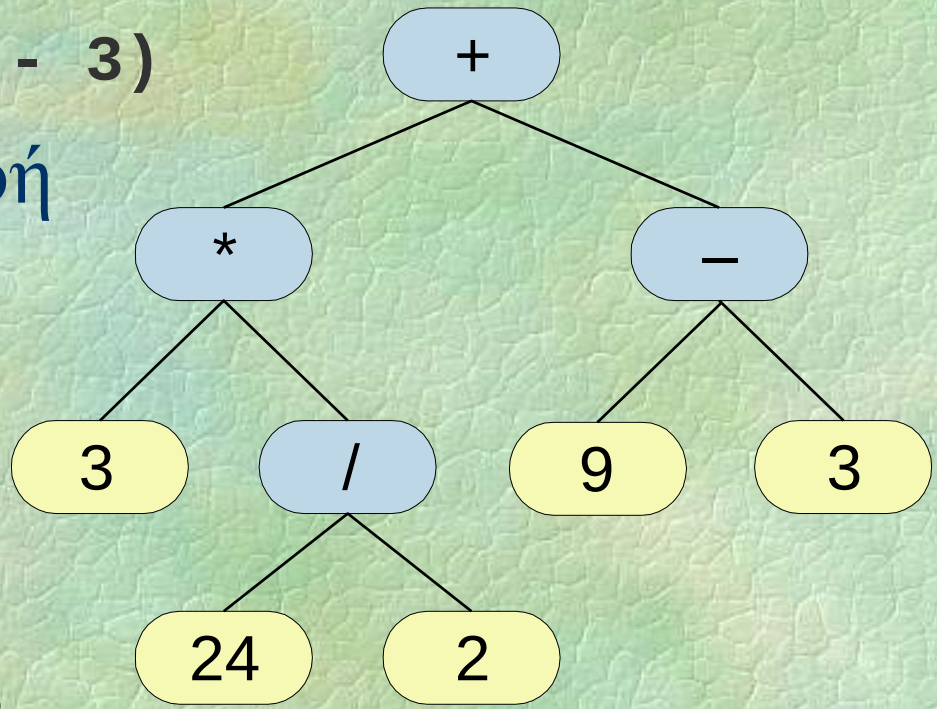
```
while (!queueIsEmpty(q)) {
    TreeNode * n = queueRemove(&q);

    printf("%d ", n->data);
    if (n->left != NULL)
        queueInsert(&q, n->left);
    if (n->right != NULL)
        queueInsert(&q, n->right);
  }
}
```

Αριθμητικές εκφράσεις (i)

- Παράδειγμα

```
3 * (24 / 2) + (9 - 3)
```

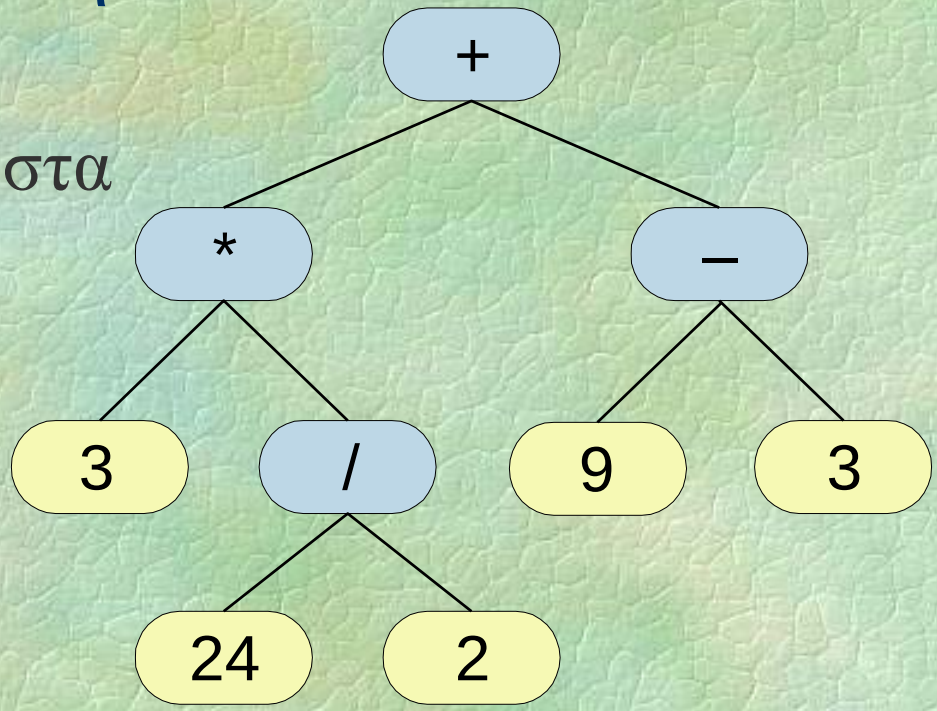
- Παράσταση σε μορφή δυαδικού δένδρου
  - οι αριθμοί στα φύλλα
  - οι τελεστές στους υπόλοιπους κόμβους



- Διάσχιση κατά βάθος

Αριθμητικές εκφράσεις (ii)

- Ενθεματική παράσταση
  - infix notation
  - ο τελεστής ανάμεσα στα τελούμενα
  - διφορούμενη: χρειάζονται παρενθέσεις
  - η συνήθης μορφή για τον άνθρωπο

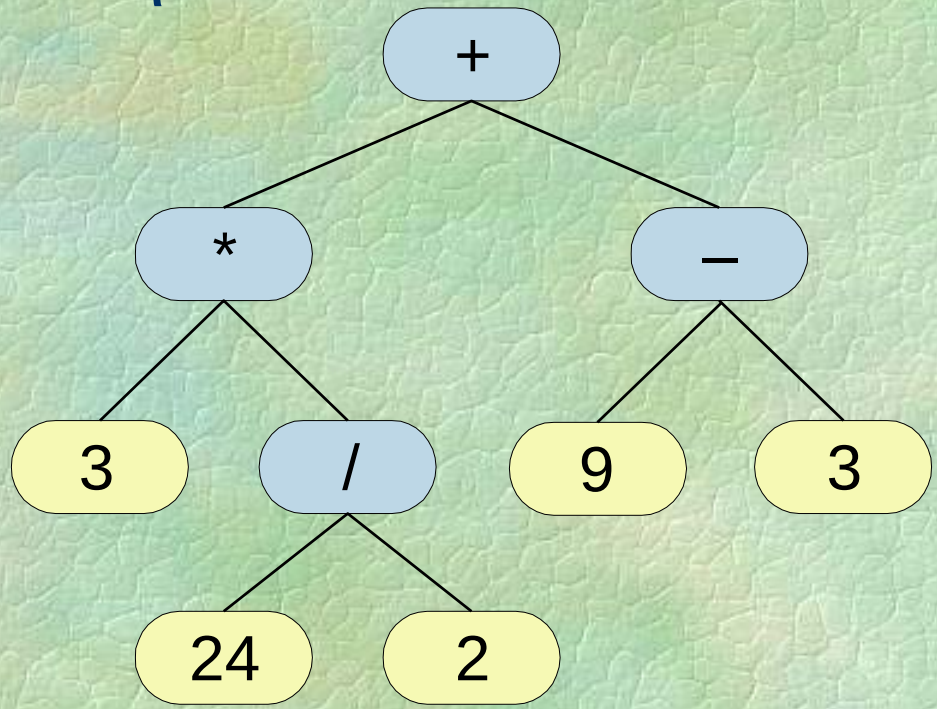


- Αποτέλεσμα

```
(3 * (24 / 2)) + (9 - 3)
```

Αριθμητικές εκφράσεις (iii)

- Προθεματική παράσταση
  - prefix notation
  - ο τελεστής πριν τα τελούμενα
  - όχι διφορούμενη, δε χρειάζονται παρενθέσεις
  - απλή μηχανική ανάγνωση

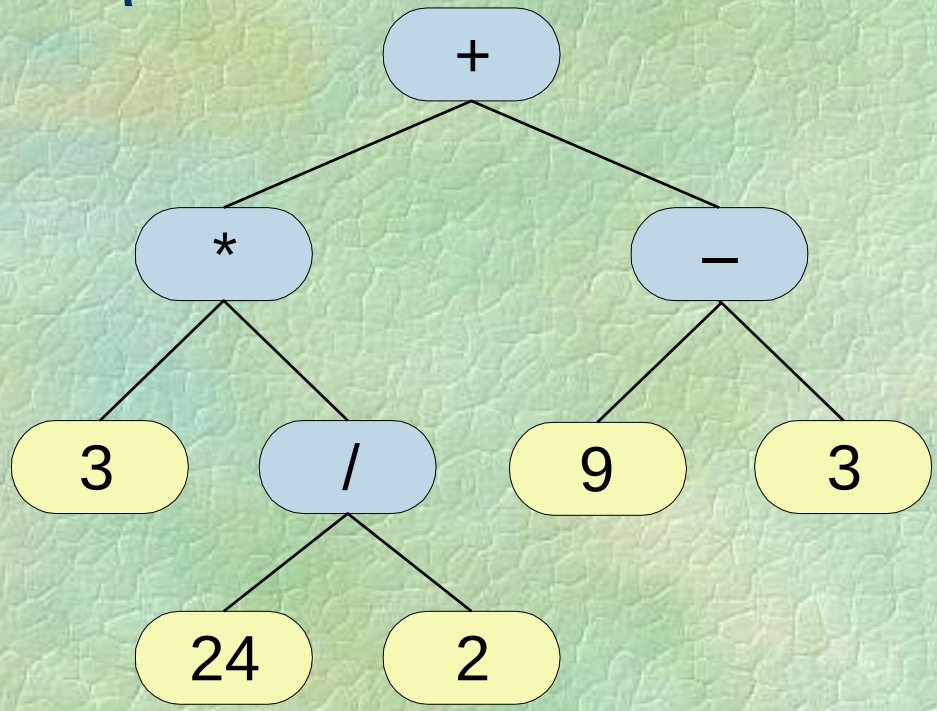


- Αποτέλεσμα

```
+ * 3 / 24 2 - 9 3
```

Αριθμητικές εκφράσεις (iv)

- Επιθεματική παράσταση
  - postfix notation
  - ο τελεστής μετά τα τελούμενα
  - όχι διφορούμενη, δε χρειάζονται παρενθέσεις
  - απλή μηχανική αποτίμηση

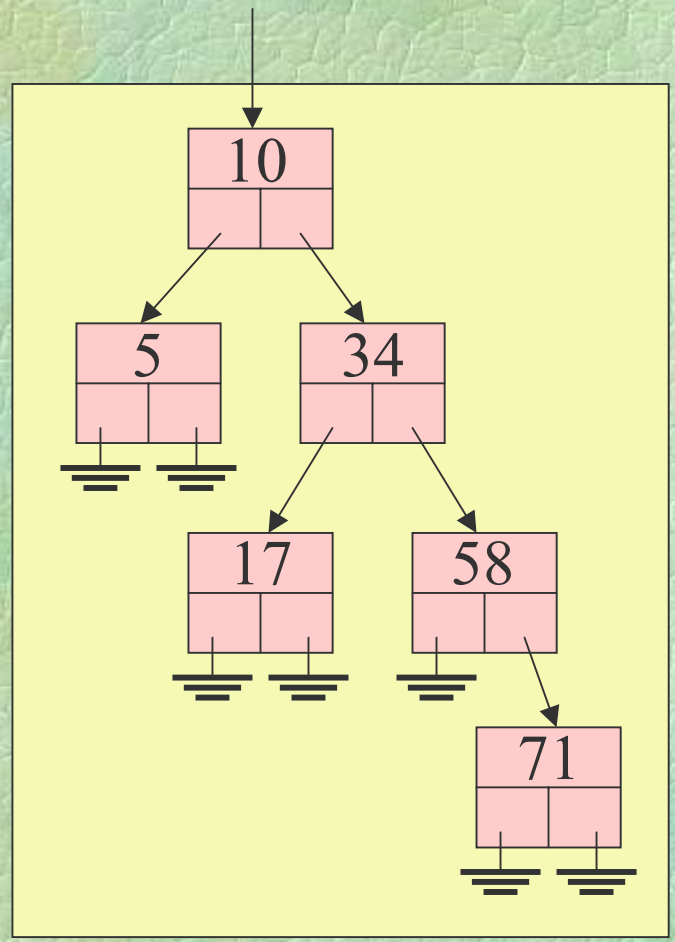


- Αποτέλεσμα

```
3 24 2 / * 9 3 - +
```

Δυαδικά δένδρα αναζήτησης (i)

- Binary search trees
- Δυαδικά δένδρα με τις παρακάτω ιδιότητες για κάθε κόμβο:
  - όλοι οι κόμβοι του αριστερού παιδιού έχουν τιμές μικρότερες ή ίσες της τιμής του κόμβου
  - όλοι οι κόμβοι του δεξιού παιδιού έχουν τιμές μεγαλύτερες ή ίσες της τιμής του κόμβου



Δυαδικά δένδρα αναζήτησης (ii)

- Τα δυαδικά δένδρα αναζήτησης διευκολύνουν την αναζήτηση στοιχείων
- Αναδρομική αναζήτηση
  - αν η τιμή που ζητείται είναι στη ρίζα, βρέθηκε
  - αν είναι μικρότερη από την τιμή της ρίζας, αρκεί να αναζητηθεί στο αριστερό παιδί
  - αν είναι μεγαλύτερη από την τιμή της ρίζας, αρκεί να αναζητηθεί στο δεξί παιδί
- Κόστος αναζήτησης: $O(\log n)$
  - υπό την προϋπόθεση το δένδρο να είναι ισοζυγισμένο

Δυαδικά δένδρα αναζήτησης (iii)

- Αναζήτηση

```
TreeNode * treeSearch (tree t, int key)
{
    if (t == NULL)
        return NULL;        /* not found */

    if (t->data == key)
        return t;           /* found */

    if (t->data > key)
        return treeSearch(t->left, key);
    else
        return treeSearch(t->right, key);
}
```