



Δομές Δεδομένων

Δένδρα

Α. Συμβώνης

Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών

Τομέας Μαθηματικών

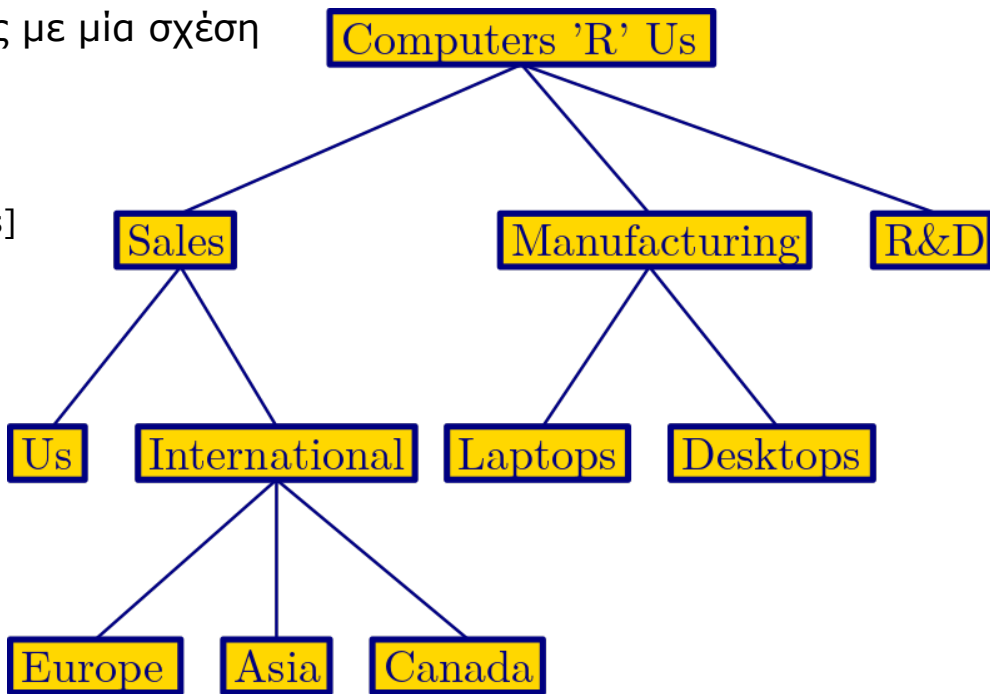


Δένδρο [Tree]



Δένδρο [Tree]

- Στην επιστήμη των υπολογιστών, ένα **δένδρο** είναι ένα αφηρημένο μοντέλο μιας ιεραρχικής δομής
- Ένα δένδρο αποτελείται από κόμβους με μία σχέση γονέα – παιδιού
- Εφαρμογές
 - Οργανογράμματα [organizational charts]
 - Συστήματα αρχείων
 - Περιβάλλοντα προγραμματισμού

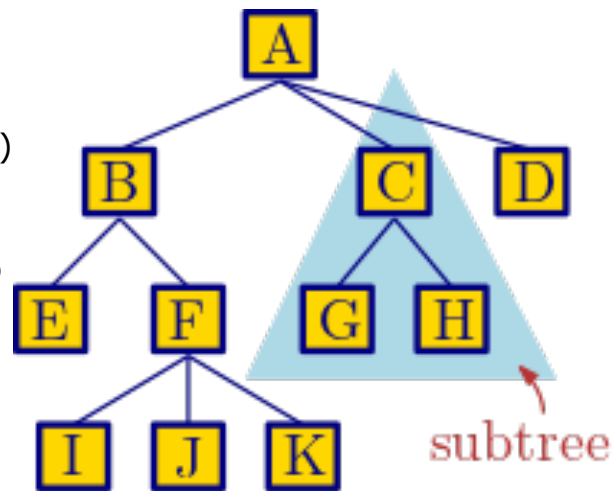




Δένδρο [Tree]

- Ορολογία

- Ρίζα [root]: ο κόμβος χωρίς γονέα (A)
- Εσωτερικός κόμβος [internal node]: κόμβος με τουλάχιστον ένα παιδί (A,B,C,F)
- Εξωτερικός κόμβος – φύλλο [external node - leaf]: κόμβος χωρίς παιδιά (E,I,J,K,G,H,D)
- Πρόγονοι [ancestors]: γονέας, παππούς, προ-πάππους, κλπ.
- Βάθος κόμβου [depth]: το πλήθος των προγόνων του κόμβου
- Ύψος δένδρου [height]: το μέγιστο βάθος όλων των κόμβων (3)
- Απόγονοι κόμβου [descendants]: παιδί, εγγόνι, κλπ.
- Υποδένδρο [subtree]: ένα δένδρο αποτελούμενο από ένα κόμβο και τους απογόνους του





Δένδρο [Tree]

• ΑΤΔ Δένδρο

- ο Οι κόμβοι επεκτείνουν τη διαπροσωπία `Position`
- ο Γενικοί μέθοδοι:
 - `int size()`
 - `boolean isEmpty()`
 - `Iterator iterator()`
 - `Iterable positions()`
- ο Μέθοδοι πρόσβασης:
 - `Position root()`
 - `Position parent(p)`
 - `Iterable children(p)`
 - `int numChildren(p)`
- ο Μέθοδοι ερώτησης:
 - `boolean isInternal(p)`
 - `boolean isExternal(p)`
 - `boolean isRoot(p)`
- ο Επιπλέον μέθοδοι ενημέρωσης μπορούν να οριστούν από τις δομές δεδομένων που υλοποιούν τον ΑΤΔ Δένδρο



Δένδρο [Tree]

- Η διαπροσωπία Tree

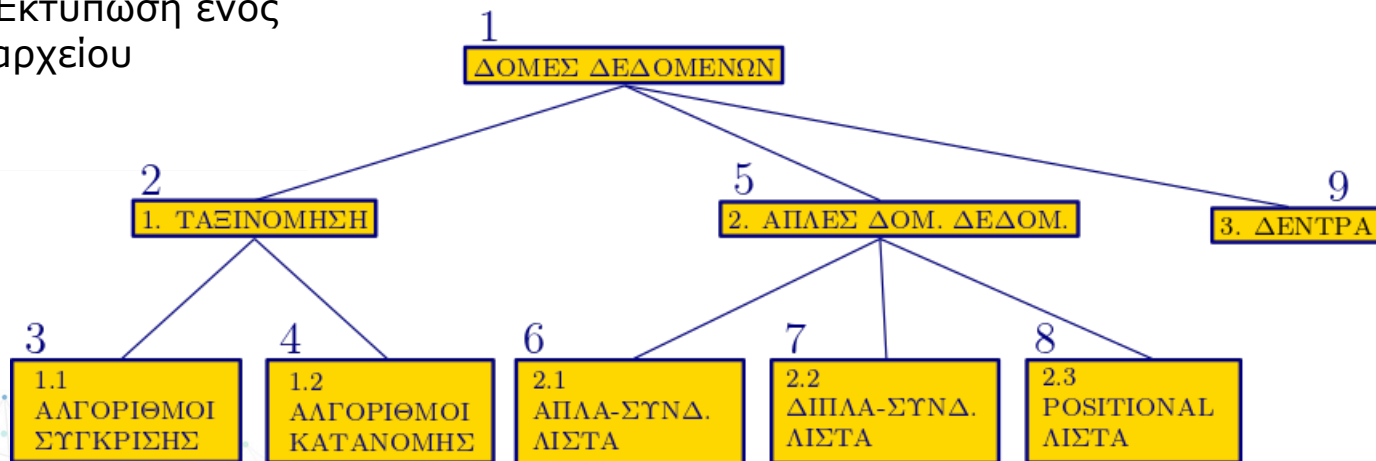
```
3 public interface Tree<E> extends Iterable<E> {  
4     Position<E> root();  
5     Position<E> parent(Position<E> p) throws IllegalArgumentException;  
6     Iterable<Position<E>> children(Position<E> p) throws IllegalArgumentException;  
7     int numChildren(Position<E> p) throws IllegalArgumentException;  
8     boolean isInternal(Position<E> p) throws IllegalArgumentException;  
9     boolean isExternal(Position<E> p) throws IllegalArgumentException;  
10    boolean isRoot(Position<E> p) throws IllegalArgumentException;  
11    int size();  
12    boolean isEmpty();  
13    Iterator<E> iterator();  
14    Iterable<Position<E>> positions();  
15 }
```



Προ-διατεταγμένη Διαπέραση Δένδρου [Preorder traversal]

- Μία διαπέραση των κόμβων του δένδρου με ένα συστηματικό τρόπο
- Σε μία **preorder** διαπέραση, «επισκεπτόμαστε/επεξεργαζόμαστε» έναν κόμβο **πριν** από τους απογόνους του
- Εφαρμογή: Εκτύπωση ενός δομημένου αρχείου

```
Algorithm preOrder(v)
    visit(v)
    for each child w of v
        preOrder(w)
```

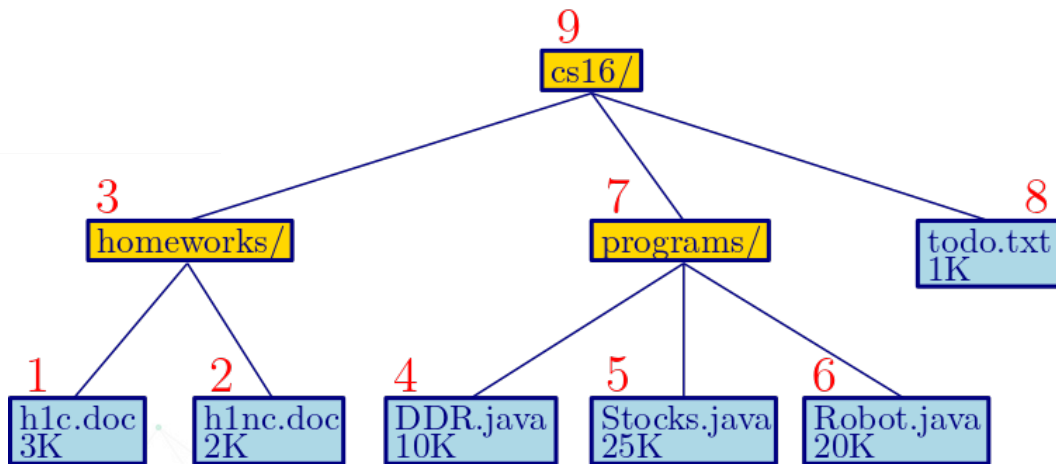




Μετα-διατεταγμένη Διαπέραση Δένδρου [Postorder traversal]

- Σε μία **postorder** διαπέραση, «επισκεπτόμαστε/επεξεργαζόμαστε» έναν κόμβο **μετά** από τους απογόνους του
- Εφαρμογή: Εκτίμηση μεγέθους μνήμης ενός φακέλου και των υποφακέλων του

```
Algorithm postOrder(v)
  for each child w of v
    postOrder(w)
  visit(v)
```





Δένδρο [Tree]

- Υλοποίηση της postOrder διαπέρασης

- abstract class AbstractTree<E>

```
192 * Adds positions of the subtree rooted at Position p to the given
193 * snapshot using a postorder traversal
194 * @param p      Position serving as the root of a subtree
195 * @param snapshot a list to which results are appended
196 */
197 private void postorderSubtree(Position<E> p, List<Position<E>> snapshot) {
198     for (Position<E> c : children(p))
199         postorderSubtree(c, snapshot);
200     snapshot.add(p); // for postorder, we add position p after exploring subtrees
201 }
202
203 /**
204 * Returns an iterable collection of positions of the tree, reported in postorder.
205 * @return iterable collection of the tree's positions in postorder
206 */
207 public Iterable<Position<E>> postorder() {
208     List<Position<E>> snapshot = new ArrayList<>();
209     if (!isEmpty())
210         postorderSubtree(root(), snapshot); // fill the snapshot recursively
211     return snapshot;
212 }
213
```



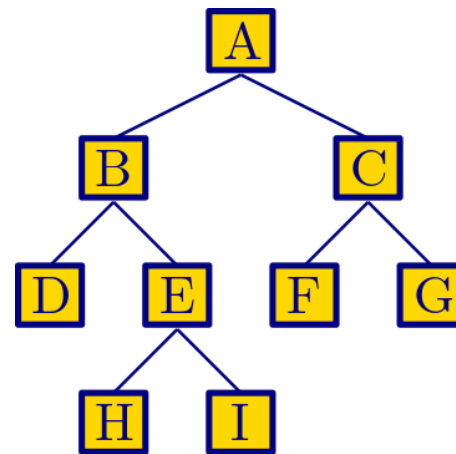
Δυαδικό Δένδρο

[Binary Tree]



Δυαδικό Δένδρο [Binary Tree]

- Ένα **δυαδικό δένδρο** είναι ένα δένδρο για το οποίο ισχύουν τα παρακάτω:
 - Κάθε εσωτερικός κόμβος έχει το πολύ **δύο** παιδιά (ακριβώς δύο για **κανονικά/πλήρη** δυαδικά δένδρα)
 - Τα (το πολύ δύο) παιδιά ενός εσωτερικού κόμβου είναι διατεταγμένα
- Τα παιδιά ενός εσωτερικού κόμβου ονομάζονται **αριστερό** και **δεξί** παιδί
- [Εναλλακτικός αναδρομικός ορισμός] Ένα **δυαδικό δένδρο** είναι ένα από τα παρακάτω:
 - Ένα δένδρο αποτελούμενο από ένα μόνο κόμβο
 - Ένα δένδρο του οποίου η ρίζα έχει το πολύ δύο διατεταγμένα παιδιά, καθένα από τα οποία είναι ρίζα σε ένα δυαδικό υποδένδρο
- Εφαρμογές:
 - Αριθμητικές εκφράσεις
 - διαδικασίες απόφασης
 - αναζητήσεις



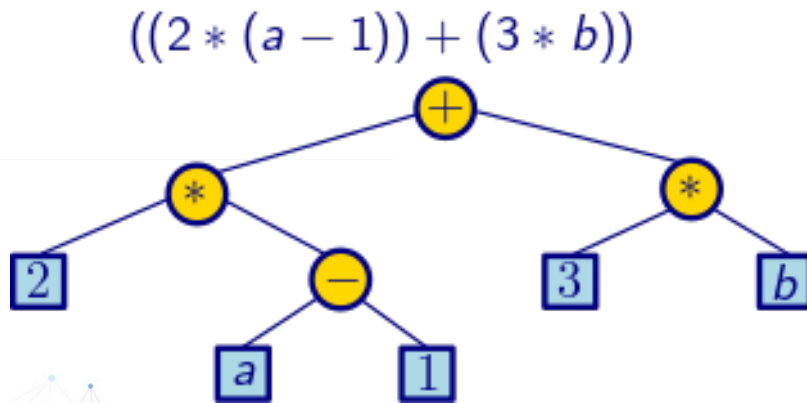


Δυαδικό Δένδρο [Binary Tree]

- Εφαρμογή: Δένδρο Αριθμητικών Εκφράσεων

- Ένα δυαδικό δένδρο το οποίο αναπαριστά μία αριθμητική έκφραση
 - εσωτερικοί κόμβοι: τελεστές (σύμβολα)
 - εξωτερικοί κόμβοι: αριθμοί, μεταβλητές

Παράδειγμα:



- Η postorder διαπέραση δίνει την postfix μορφή της έκφρασης
- Γνωστή και ως **Polish form**

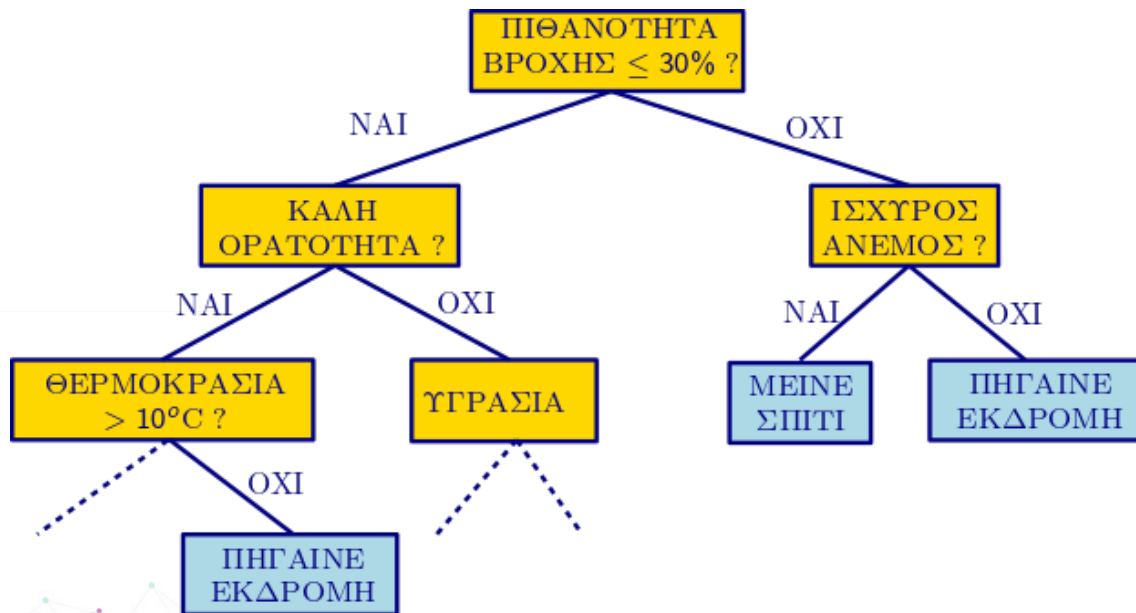
Παράδειγμα: $2\ a\ 1\ -\ *\ 3\ b\ *\ +$



Διαδικό Δένδρο [Binary Tree]

- Εφαρμογή: Δένδρο λήψης αποφάσεων

- Ένα διαδικό δένδρο το οποίο αναπαριστά μια διαδικασία λήψης αποφάσεων
 - εσωτερικοί κόμβοι: ερωτήσεις με ναι/όχι απάντηση
 - εξωτερικοί κόμβοι: αποφάσεις
- Παράδειγμα:





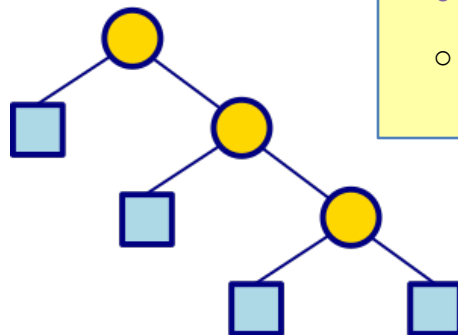
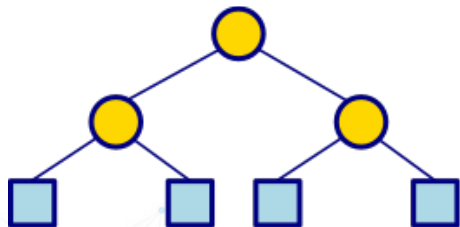
Δυαδικό Δένδρο [Binary Tree]

- Ιδιότητες Κανονικού/πλήρους Δυαδικού Δένδρου

- Συμβολισμός
 - n : αριθμός κόμβων
 - e : αριθμός εξωτερικών κόμβων
 - i : αριθμός εσωτερικών κόμβων
 - h : ύψος

- Ιδιότητες

- $e = i + 1$
- $n = 2e - 1$
- $h \leq i$
- $h \leq (n - 1)/2$
- $e \leq 2^h$
- $h \geq \log_2 e$
- $h \geq \log_2(n + 1) - 1$





Δυαδικό Δένδρο [Binary Tree]

- Ο ΑΤΔ Δυαδικό Δένδρο

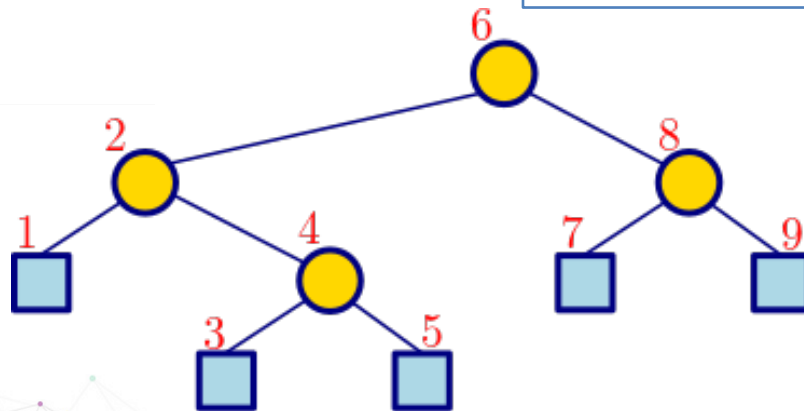
- ο Ο ΑΤΔ Δυαδικό Δένδρο επεκτείνει τον ΑΤΔ Δένδρο
- ο Κληρονομεί όλες τις μεθόδους του ΑΤΔ Δένδρο
- ο Επιπρόσθετοι μέθοδοι:
 - `Position left(p)`
 - `Position right(p)`
 - `Position sibling(p)`
- ο Οι παραπάνω μέθοδοι επιστρέφουν `null` αν ο `p` δεν έχει αριστερό παιδί, δεξί παιδί ή αδελφό, αντίστοιχα
- ο Επιπλέον μέθοδοι ενημέρωσης μπορούν να οριστούν από τις δομές δεδομένων που υλοποιούν τον ΑΤΔ Δυαδικό Δένδρο



Ένδο-διατεταγμένη Διαπέραση Δένδρου [Inorder traversal]

- Σε μία **inorder** διαπέραση, «**επισκεπτόμαστε/επεξεργαζόμαστε**» έναν κόμβο **μετά** από το αριστερό υποδένδρο και **πριν** από το δεξί υποδένδρο
- Εφαρμογή: Απεικόνιση δυαδικού δένδρου
 - $x(v)$ = inorder rank of v
 - $y(v)$ = depth of v

```
Algorithm inOrder( $v$ )  
  if  $\text{left}(v) \neq \text{null}$   
    inOrder( $\text{left}(v)$ )  
  visit( $v$ )  
  if  $\text{right}(v) \neq \text{null}$   
    inOrder( $\text{right}(v)$ )
```



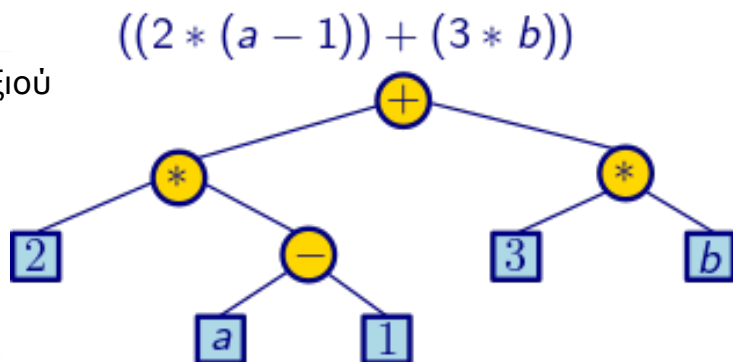


Ένδο-διατεταγμένη Διαπέραση Δένδρου [Inorder traversal]

- Εφαρμογή: Εκτύπωση Αριθμητικής Έκφρασης
 - Εξειδίκευση της inorder διαπέρασης
 - Εκτύπωση τελεστή ή σταθεράς/μεταβλητής όταν επισκεπτόμαστε τον κόμβο
 - Εκτύπωση παρένθεσης '(' πριν τη διαπέραση του αριστερού παιδιού
 - Εκτύπωση παρένθεσης ')' μετά τη διαπέραση του δεξιού παιδιού

Algorithm `printExpression(v)`

```
if left(v) != null
    print("(")
    printExpression(left(v))
print(v.element())
if right(v) != null
    printExpression(right(v))
print(") ")
```



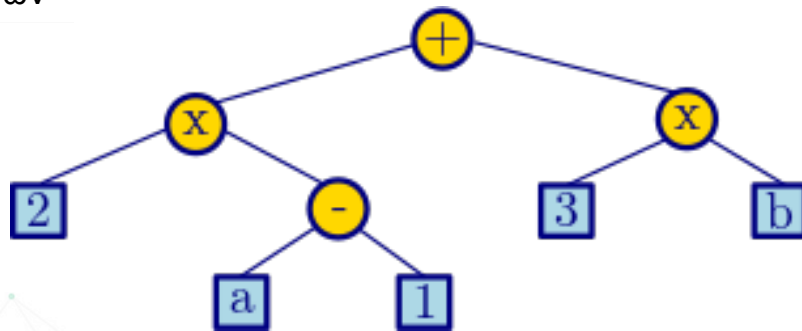


Ένδο-διατεταγμένη Διαπέραση Δένδρου [Inorder traversal]

- Εφαρμογή: Υπολογισμός Αριθμητικής Έκφρασης
 - Εξειδίκευση της postorder διαπέρασης
 - Αναδρομική μέθοδος η οποία επιστρέφει την αριθμητική έκφραση που αντιστοιχεί στο υποδένδρο
 - Όταν επισκέπτεται εσωτερικό κόμβο συνδυάζει τις τιμές των υποδέντρων

Algorithm `evalExpr(v)`

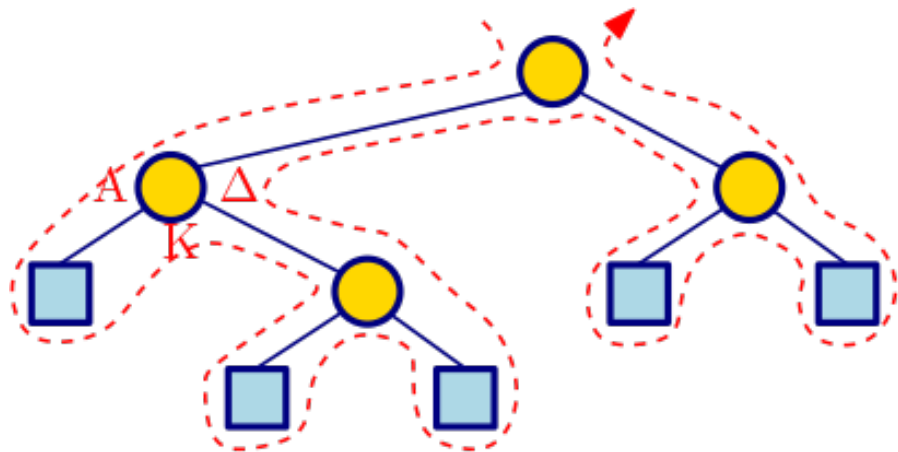
```
if isExternal(v)
    return v.element()
else
    x = evalExpr(left(v))
    y = evalExpr(right(v))
     $\diamond$  = v.element()
    return x $\diamond$ y
```





Euler Tour Διαπέραση

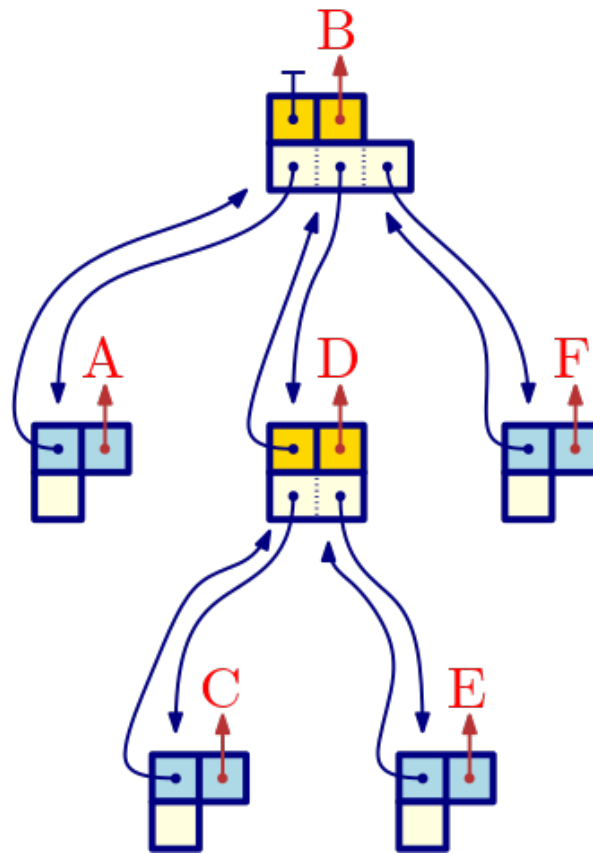
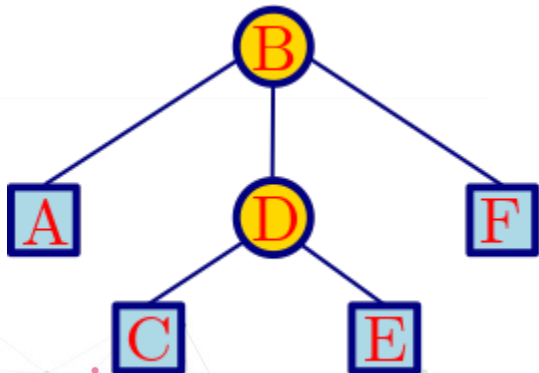
- Γενικευμένη διαπέραση ενός δυαδικού δένδρου
- Περιλαμβάνει μια preorder, μια postorder και μια inorder διαπέραση
- “Περπάτησε” περιμετρικά το δένδρο και επισκέψου/επεξεργάσου κάθε κόμβο τρεις φορές:
 - Όταν βρισκόμαστε αριστερά του(preorder)
 - Όταν βρισκόμαστε από κάτω του (inorder)
 - Όταν βρισκόμαστε δεξιά του (postorder)





Συνδεδεμένη Δομή για Δένδρα

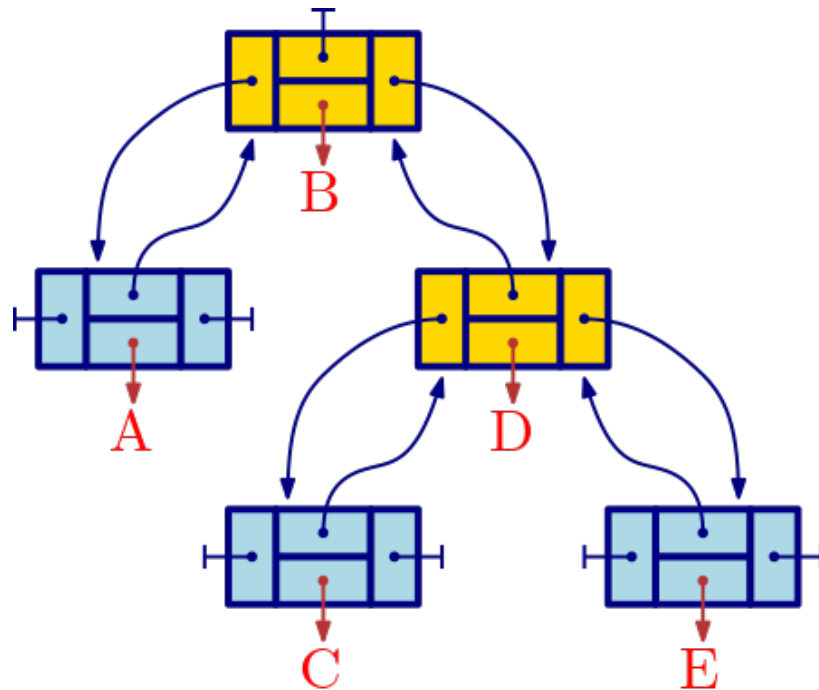
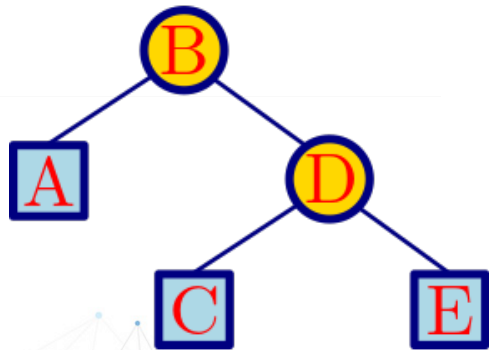
- Ένας κόμβος [Node] υλοποιείται από ένα αντικείμενο, το οποίο αποθηκεύει:
 - ο Τα δεδομένα
 - ο Τον κόμβο γονέα
 - ο Μια ακολουθία από τους κόμβους παιδιά
- Το αντικείμενο Node υλοποιεί τον ΑΤΔ Position





Συνδεδεμένη Δομή για Δυαδικά Δένδρα

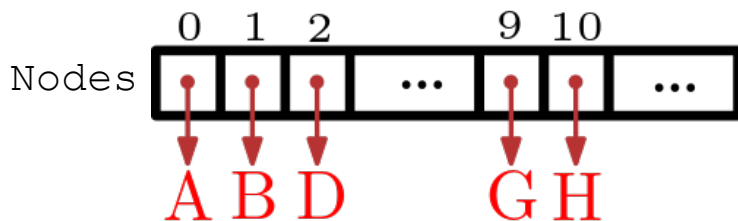
- Ένας κόμβος [Node] υλοποιείται από ένα αντικείμενο, το οποίο αποθηκεύει:
 - ο Τα δεδομένα
 - ο Τον κόμβο γονέα
 - ο Τον κόμβο αριστερό παιδί
 - ο Τον κόμβο δεξί παιδί
- Το αντικείμενο Node υλοποιεί τον ΑΤΔ Position





Διανυσματική Απεικόνιση Δυαδικών Δέντρων

- Οι κόμβοι αποθηκεύονται σε ένα διάνυσμα Nodes



Ο κόμβος v αποθηκεύεται στη θέση $\text{Nodes}[\text{rank}(v)]$

- $\text{rank}(\text{root}) = 0$
- Εάν ο κόμβος nd είναι αριστερό παιδί, τότε:
$$\text{rank}(nd) = 2 * \text{rank}(\text{parent}(nd)) + 1$$
- Εάν ο κόμβος nd είναι δεξί παιδί, τότε:
$$\text{rank}(nd) = 2 * \text{rank}(\text{parent}(nd)) + 2$$

