

ΔΕΝΤΡΑ

5.2 Εισαγωγή στα Δέντρα και Δυναμικά Δέντρα

Ένα **δέντρο** (**tree**) αποτελείται από:

- ένα πεπερασμένο σύνολο στοιχείων που ονομάζονται **κόμβοι** (**nodes**) ή **κορυφές** (**vertices**) και
- από ένα πεπερασμένο σύνολο **κατευθυνόμενων τόξων** (**directed arcs**), τα οποία συνδέουν ζεύγη κόμβων.

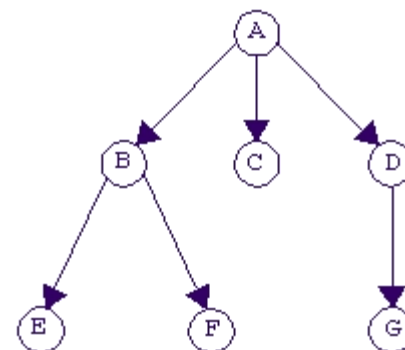
Αν το δέντρο δεν είναι άδειο, τότε υπάρχει ένας κόμβος, ο οποίος δεν έχει εισερχόμενα τόξα. Αυτός ο κόμβος ονομάζεται **ρίζα** (**root**) του δέντρου και ξεκινώντας από αυτόν μπορούμε να καταλήξουμε σε οποιονδήποτε από τους υπόλοιπους κόμβους ακολουθώντας μια μοναδική διαδρομή από τόξα.

Οι υπόλοιποι κόμβοι χωρίζονται σε $n \geq 0$ ξένα μεταξύ τους υποσύνολα, T_1, T_2, \dots, T_n , κάθε ένα από τα οποία είναι ένα δέντρο και ονομάζεται **υποδέντρο** (**subtree**) της ρίζας.

- Οι κόμβοι που δεν έχουν εξερχόμενα τόξα ονομάζονται **φύλλα (leaves)**, ενώ οι κόμβοι που δεν είναι ούτε φύλλα ούτε ρίζα ονομάζονται **εσωτερικοί κόμβοι (internal nodes)**.
- Επίσης, οι κόμβοι που είναι άμεσα προσπελάσιμοι από έναν κόμβο (δηλαδή με χρήση ενός μόνο κατευθυνόμενου τόξου) ονομάζονται **παιδιά (children)** του κόμβου αυτού και ο ίδιος ο κόμβος ονομάζεται **πατέρας (parent)** των παιδιών του.
- Οι κόμβοι που είναι παιδιά του ίδιου κόμβου-πατέρα ονομάζονται **αδέρφια (siblings)**.
- Τέλος, όσον αφορά στους κόμβους ενός δέντρου, ο αριθμός των υποδέντρων ενός κόμβου ονομάζεται **βαθμός του κόμβου** αυτού.
- Ο μέγιστος βαθμός των κόμβων του δέντρου καθορίζει και τον **βαθμό του δέντρου**.

Βασικές έννοιες: παράδειγμα

Οι έννοιες αυτές μπορούν να γίνουν περισσότερο κατανοητές με το διπλανό σχήμα:



Όπως φαίνεται από το σχήμα, τα δέντρα ζωγραφίζονται ανάποδα, με τη ρίζα στην κορυφή και τα φύλλα κάτω.

Στο συγκεκριμένο δέντρο που έχει βαθμό 3 ή αλλιώς είναι ένα **τριαδικό δέντρο**:

- ο κόμβος A είναι η ρίζα του δέντρου,
- οι κόμβοι C, E, F και G είναι φύλλα και
- οι κόμβοι B, D είναι εσωτερικοί κόμβοι.

Ο κόμβος A είναι πατέρας των B, C, D, ο B είναι πατέρας των E και F και ο D είναι πατέρας του κόμβου G.

Επίσης, οι κόμβοι B, C, D είναι αδέρφια μεταξύ τους και παιδιά του A, οι E και F είναι αδέρφια μεταξύ τους και παιδιά του B, ενώ ο G είναι παιδί του D.

Όταν αναφερόμαστε στους κόμβους ενός δέντρου συχνά χρησιμοποιούμε την έννοια του επιπέδου.

Το **επίπεδο** της ρίζας ενός δέντρου είναι 1, των παιδιών της 2 κ.ο.κ, δηλαδή αν ένας κόμβος βρίσκεται στο επίπεδο i τότε τα παιδιά του βρίσκονται στο επίπεδο $i+1$.

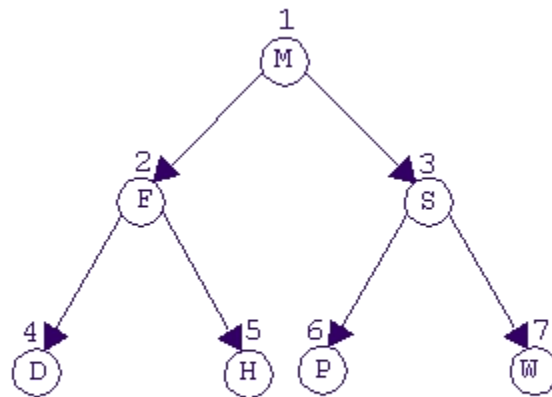
Το μεγαλύτερο επίπεδο που μπορεί να βρίσκεται ένας κόμβος του δέντρου ονομάζεται **ύψος (height)** ή **βάθος (depth)** του δέντρου.

Δυαδικά δέντρα

Ένα ειδικό είδος δέντρων είναι τα **δυαδικά δέντρα (binary trees)**, δηλαδή τα δέντρα στα οποία κάθε κόμβος έχει το πολύ δύο παιδιά ή αλλιώς έχει βαθμό 2.

Για την αποθήκευση ενός δυαδικού δέντρου μπορεί να χρησιμοποιηθεί ένας πίνακας.

Αν αριθμήσουμε τους κόμβους του δέντρου από τη ρίζα προς τα κάτω και τους κόμβους του ίδιου επιπέδου από αριστερά προς δεξιά, όπως δείχνει το σχήμα:



τότε μπορούμε να αποθηκεύσουμε τα περιεχόμενα του i κόμβου στην i θέση του πίνακα:

i	1	2	3	4	5	6	7	...
$T[i]$	M	F	S	D	H	P	W	...

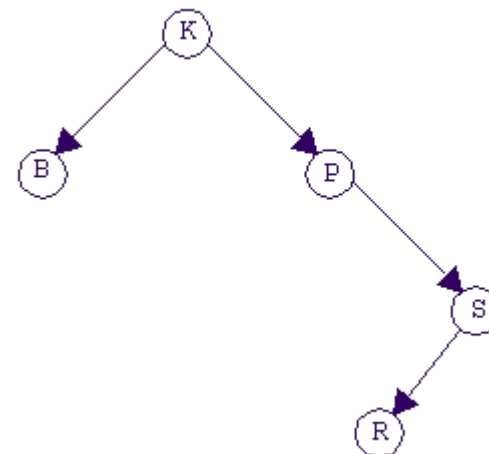
Πλήρη δέντρα

Η υλοποίηση δέντρων με πίνακα είναι πιο κατάλληλη για **πλήρη δέντρα** (**complete trees**), δηλαδή για δέντρα στα οποία κάθε επίπεδο είναι συμπληρωμένο πλήρως, εκτός ίσως από το τελευταίο, στο οποίο οι κόμβοι βρίσκονται στις πιο αριστερές θέσεις.

Το χαρακτηριστικό της πληρότητας εξασφαλίζει την αποθήκευση των στοιχείων σε διαδοχικές θέσεις στην αρχή του πίνακα.

Για άλλα είδη δέντρων, όμως, μια τέτοια υλοποίηση καταλαμβάνει πολύ χώρο.

Το διπλανό δέντρο, για παράδειγμα, χρειάζεται έναν πίνακα 15 θέσεων με τις 10 θέσεις κενές!



i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
T[i]	K	B	P				S							R		...

Χρήση συνδεδεμένων δομών

Ένας καλύτερος τρόπος αποθήκευσης δυαδικών δέντρων είναι η χρησιμοποίηση **συνδεδεμένων δομών**, όπου κάθε κόμβος έχει δύο δεσμούς:

- ο ένας δείχνει στο αριστερό παιδί του κόμβου αυτού ή είναι μηδενικός, στην περίπτωση που δεν υπάρχει αριστερό παιδί, και
- ο άλλος στο δεξί παιδί, αν υπάρχει.

Μπορούμε να έχουμε πρόσβαση σε οποιονδήποτε κόμβο του δέντρου αν **διατηρούμε έναν δείκτη στην ρίζα του**.

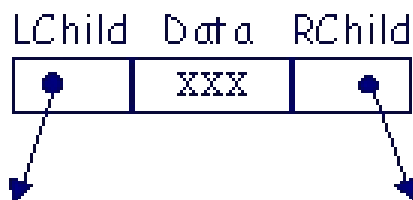
Δηλώσεις για την υλοποίηση ενός δυαδικού δέντρου:

```
typedef int BinTreeElementType;      /*ο τύπος των στοιχείων των  
                                         κόμβων ενδεικτικά τύπου int */  
  
typedef struct BinTreeNode *BinTreePointer;  
  
typedef struct BinTreeNode {  
    BinTreeElementType Data;  
    BinTreePointer LChild, RChild;  
}  
  
BinTreeNode;
```


Χρήση συνδεδεμένων δομών

Τα **πεδία δεσμού LChild και RChild** σε κάθε κόμβο είναι δείκτες που δείχνουν σε κόμβους που αντιπροσωπεύουν το αριστερό και δεξί παιδί αντίστοιχα.

Κάθε κόμβος έχει την ακόλουθη μορφή:

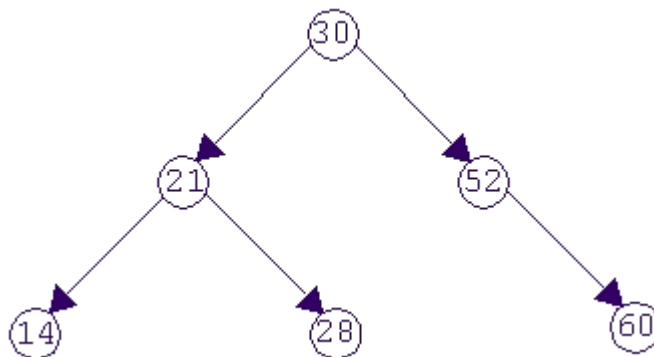


Αν δεν υπάρχει αριστερό ή δεξί παιδί, τότε ο αντίστοιχος δείκτης έχει τιμή nil.

Κατά συνέπεια, ένας κόμβος φύλλο θα έχει και τους δυο δείκτες μηδενικούς:



Έτσι, λοιπόν, για το δυαδικό δέντρο:



η αντίστοιχη συνδεδεμένη αναπαράσταση είναι η ακόλουθη:

