

ΔΔΜΕΣ ΔΕΔΟΜΕΝΩΝ

Δυαδικά Δένδρα



Απόστολος Ν. Παπαδόπουλος
Αναπληρωτής Καθηγητής
Τμήμα Πληροφορικής Α.Π.Θ.

Βασικές Έννοιες

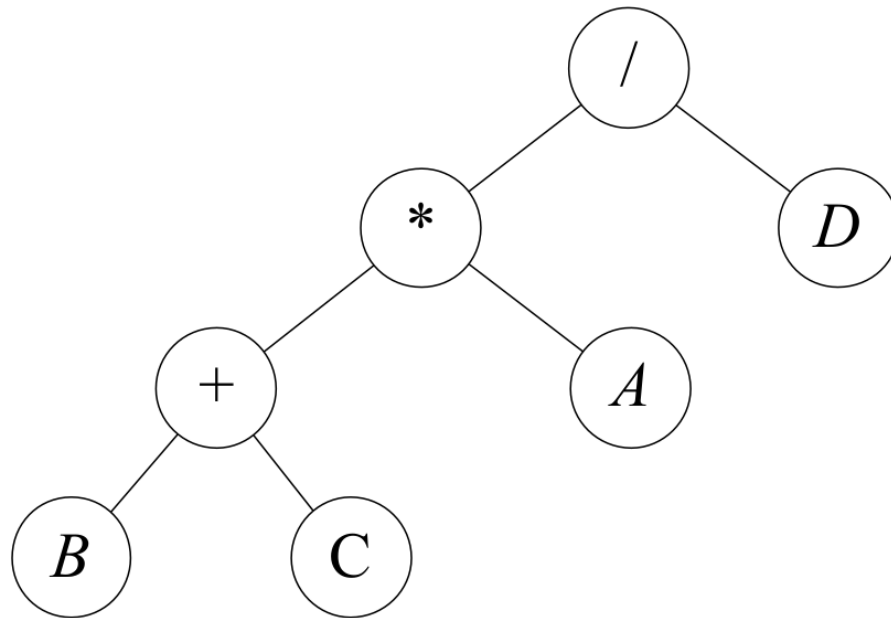
Ένα **δένδρο** αποτελείται από κόμβους και ακμές που τους συνδέουν. Κάθε κόμβος έχει παιδιά (πλην των φύλλων) και έναν μόνο γονέα (πλην της ρίζας).

Θα ξεκινήσουμε με τα **δυναμικά δένδρα**. Κάθε κόμβος έχει το πολύ δύο παιδιά. Υπάρχουν και δένδρα με μεγαλύτερο αριθμό παιδιών ανά κόμβο.

Η δομή του δένδρου γενικεύεται και για περισσότερα παιδιά ανά κόμβο.

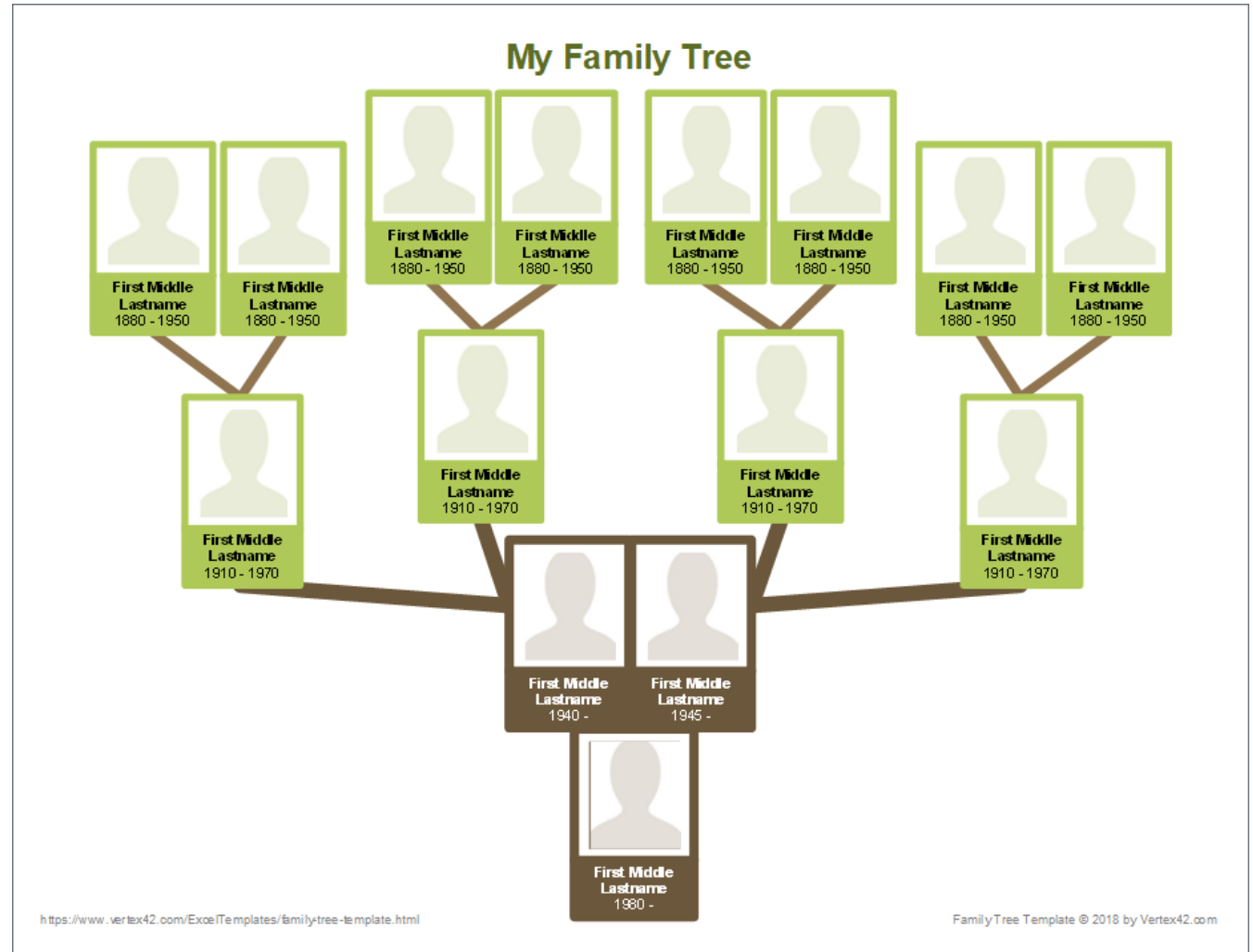
Βασικές Έννοιες

Δένδρο έκφρασης: $(B+C)*A/D$



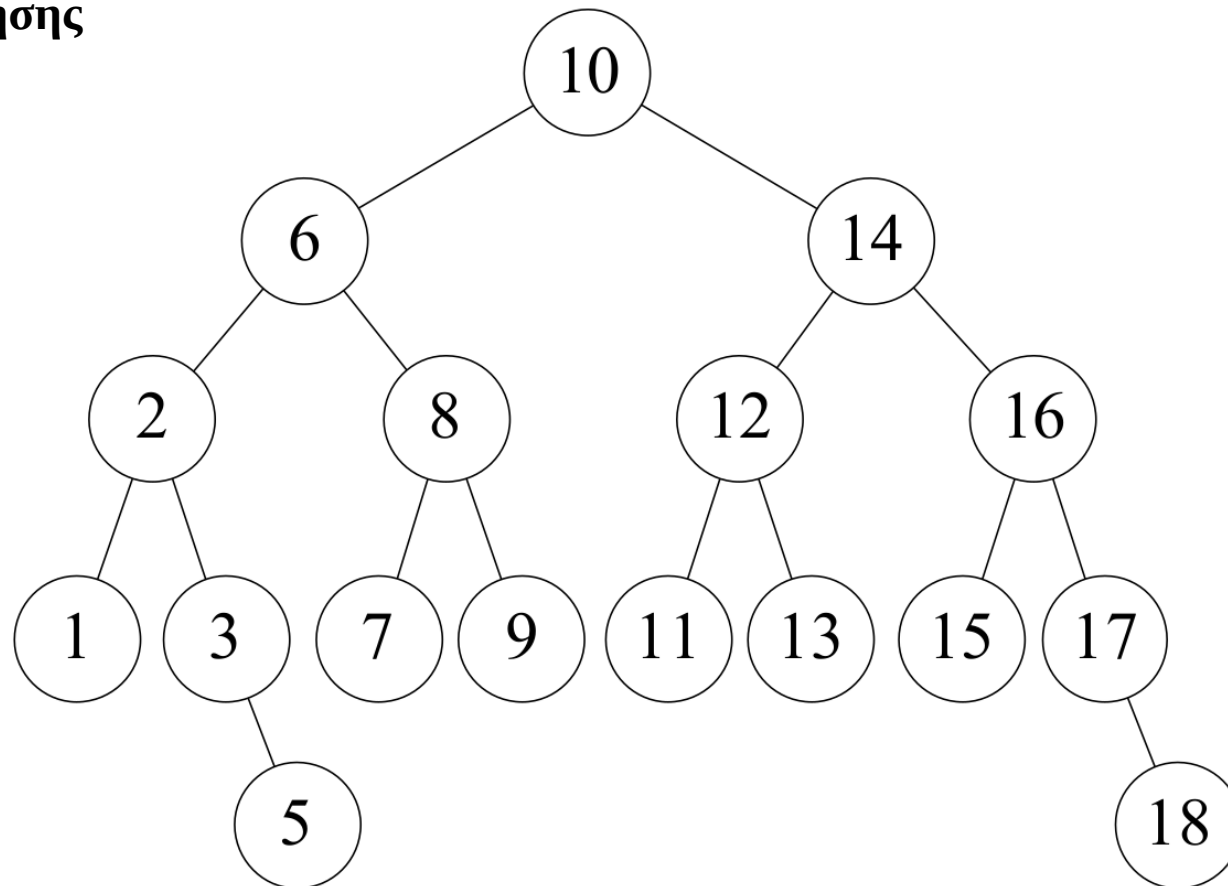
Βασικές Έννοιες

Γενεαλογικό δένδρο



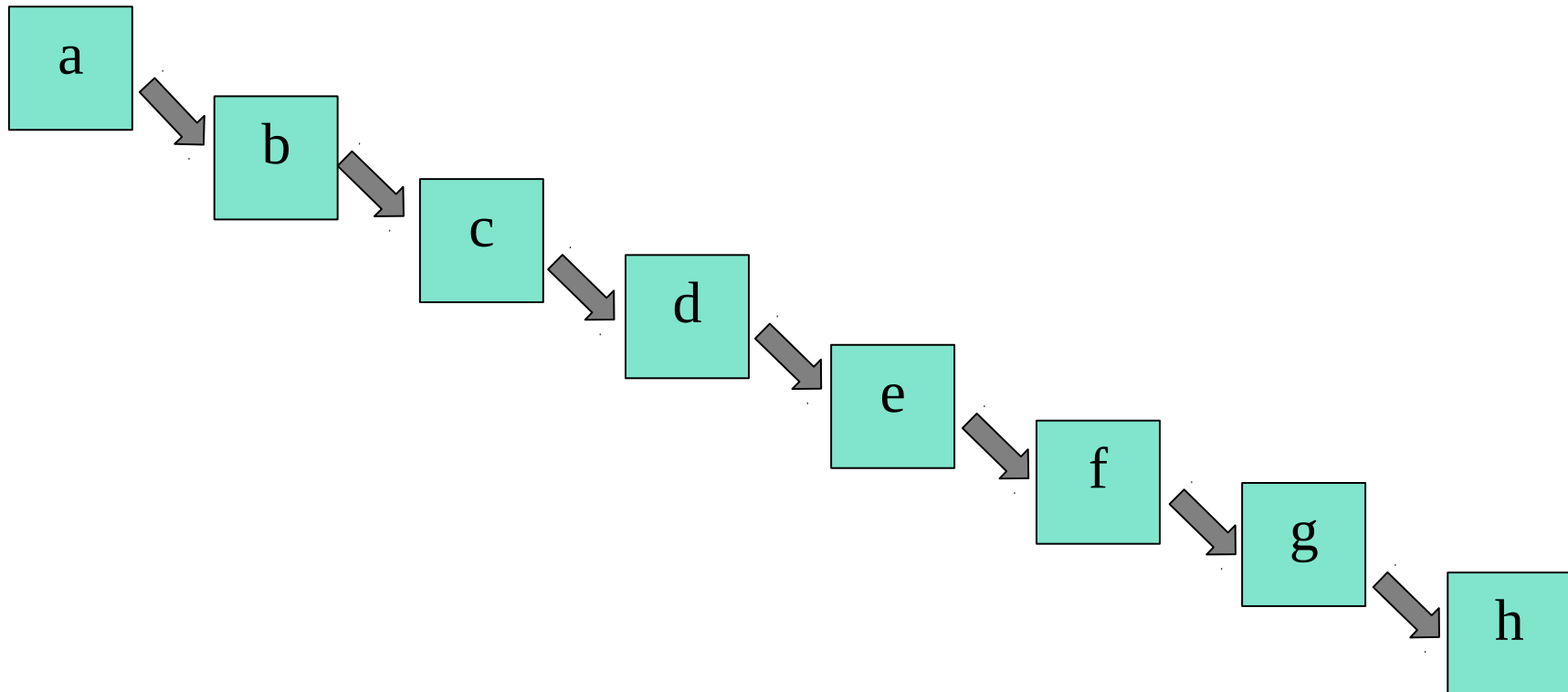
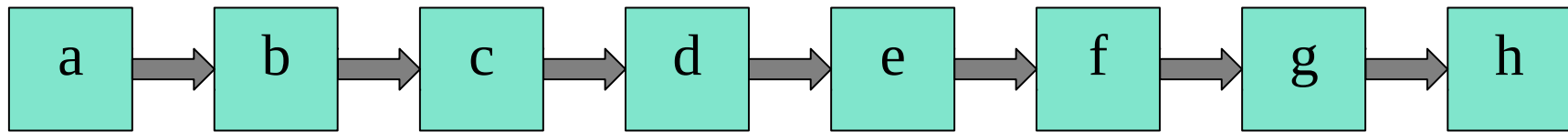
Βασικές Έννοιες

Δένδρο αναζήτησης



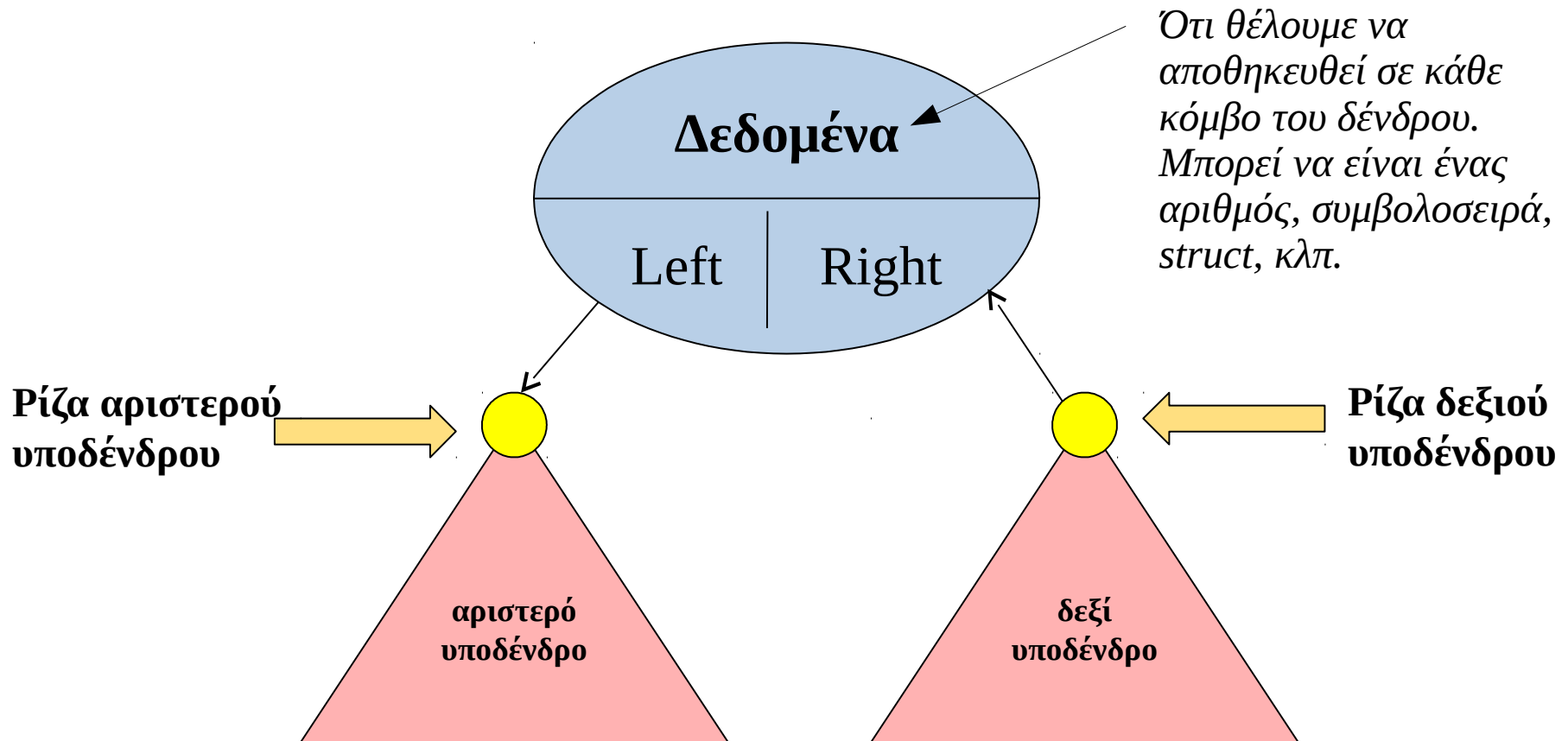
Βασικές Έννοιες

Μία συνδεδεμένη λίστα είναι ένα (εκφυλισμένο) δυαδικό δένδρο



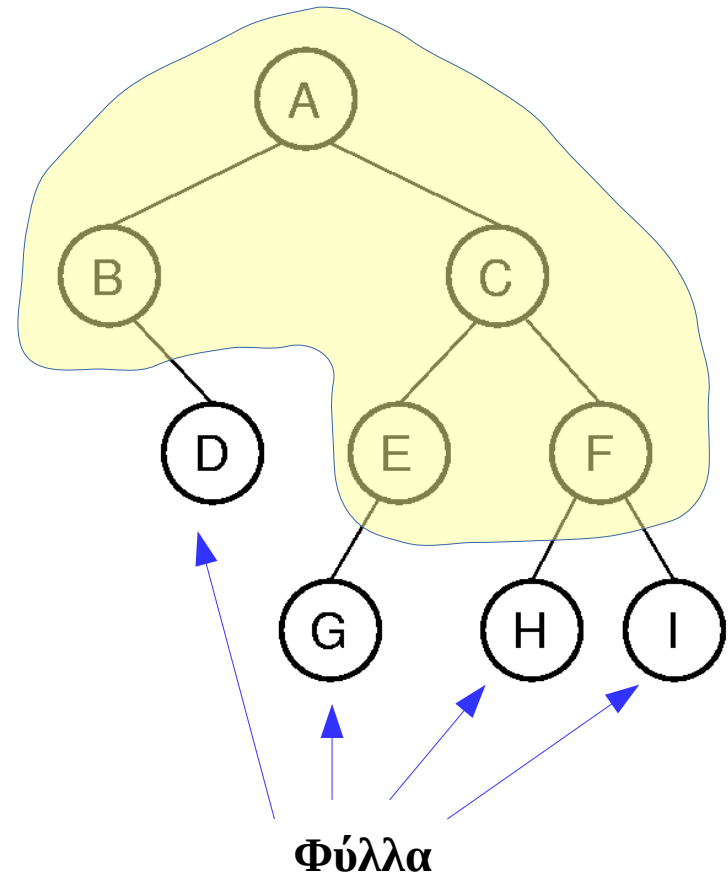
Βασικές Έννοιες

Δομή ενός κόμβου σε συνδεδεμένη αναπαράσταση



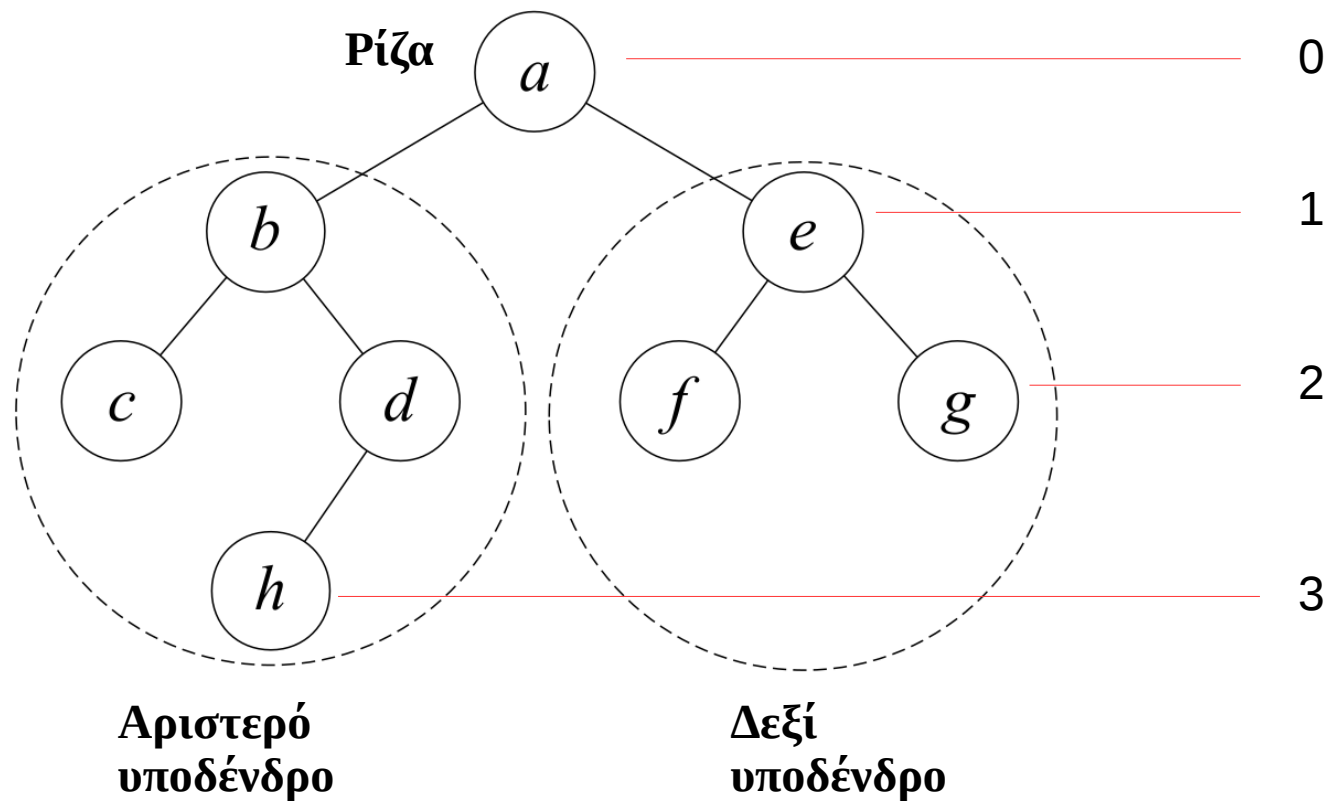
Βασικές Έννοιες

- στοιχείο (κόμβος)
- ακμή
- υποδένδρο
- παιδιά, γονιός, πρόγονος, απόγονος
- μονοπάτι
- ύψος (βάθος)
- επίπεδο
- φύλλα, εσωτερικοί κόμβοι



Βασικές Έννοιες

Δένδρο ύψους $h = 3$
Πλήθος επιπέδων = 4



Βασικές Έννοιες

Ένα δένδρο ύψους 0 έχει μόνο έναν κόμβο, τη ρίζα.

Ένα δένδρο ύψους h έχει το πολύ $2^{h+1} - 1$ κόμβους.

Κάθε κόμβος έχει το πολύ 2 παιδιά.

Κάθε κόμβος έχει έναν πρόγονο (εκτός από τη ρίζα).

Αν υποθέσουμε ότι η ρίζα έχει ύψος 1, τότε:

ένα δένδρο ύψους h έχει το πολύ $2^h - 1$ κόμβους.

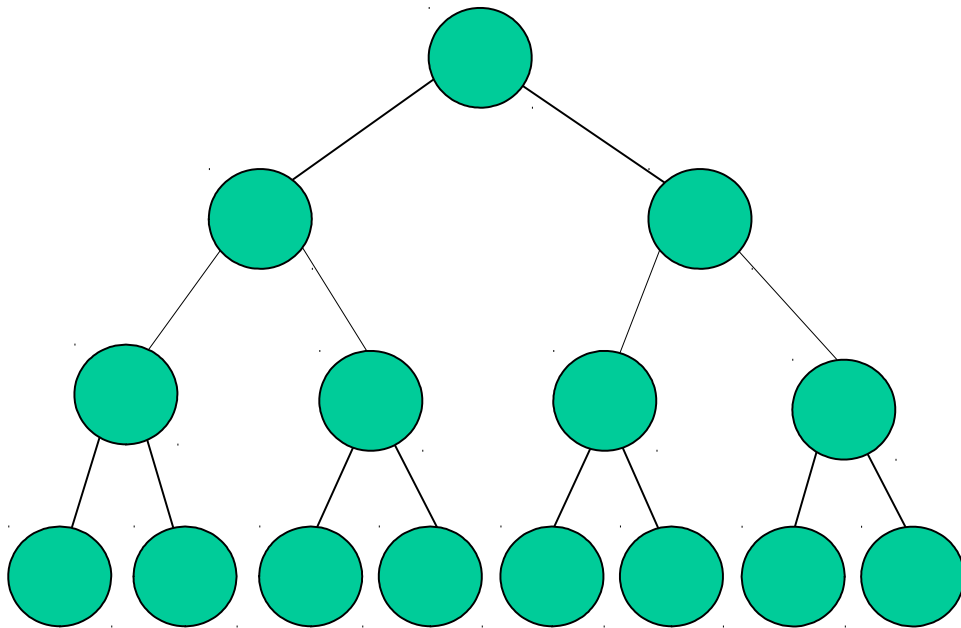
Βασικές Έννοιες

Πλήρες (full) δυαδικό δένδρο ύψους h είναι εκείνο το δυαδικό δένδρο, το οποίο περιέχει ακριβώς $2^h - 1$ στοιχεία

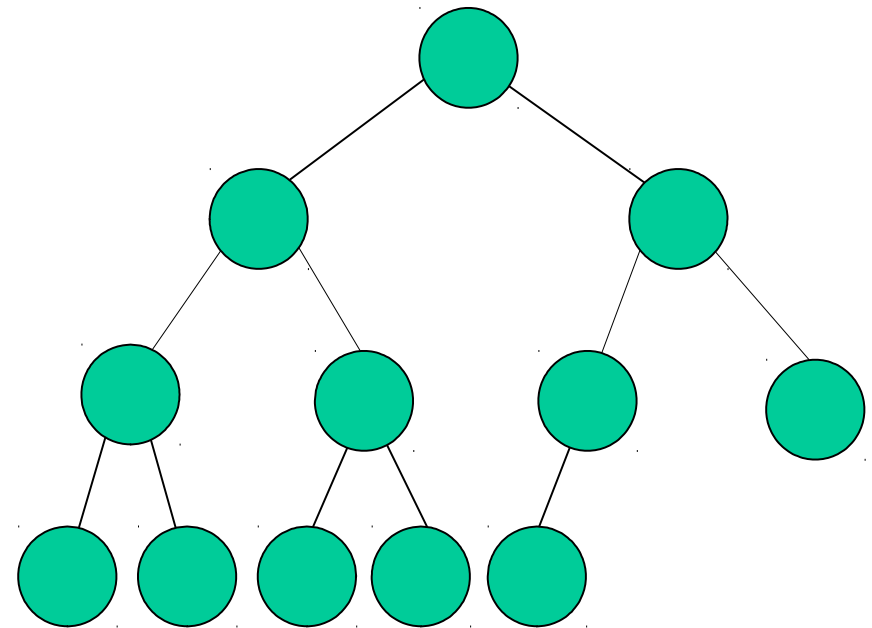
Μπορούμε να αριθμήσουμε από 1 έως $2^h - 1$ τα στοιχεία ενός πλήρους δυαδικού δένδρου ύψους h , ξεκινώντας από το επίπεδο 1 προς το επίπεδο h και, μέσα σε κάθε επίπεδο, από αριστερά προς τα δεξιά.

Συμπληρωμένο (complete) δυαδικό δένδρο ύψους h είναι εκείνο το δυαδικό δένδρο, το οποίο προκύπτει από ένα πλήρες δυαδικό δένδρο ύψους h εάν διαγράψουμε k στοιχεία με αρίθμηση $2^h - i$, για $1 \leq i \leq k$ ($k \geq 0$).

Βασικές Έννοιες



Πλήρες δυαδικό δένδρο



Συμπληρωμένο δυαδικό δένδρο

Λειτουργίες

Προσδιορισμός ύψους, πλήθους στοιχείων

Δημιουργία αντιγράφου

Παρουσίαση δένδρου σε μονάδα εξόδου

Διαγραφή δένδρου

Υπολογισμός έκφρασης (αν είναι δένδρο έκφρασης)

Όλα τα παραπάνω εκτελούνται με συστηματικό τρόπο με τη λειτουργία **διάσχισης του δένδρου**.

Διαπεράσεις

Κάθε διαδικασία επίσκεψης όλων των κόμβων ενός δένδρου, ακριβώς μια φορά τον καθένα, ονομάζεται διάσχιση (traversal).

Προδιατεταγμένη διάσχιση (preorder): Για κάθε κόμβο, επισκεπτόμαστε πρώτα τον ίδιο τον κόμβο, έπειτα τους κόμβους του αριστερού του υποδένδρου και στη συνέχεια τους κόμβους του δεξιού του υποδένδρου.

Μεταδιατεταγμένη διάσχιση (postorder): Για κάθε κόμβο, επισκεπτόμαστε πρώτα τους κόμβους του αριστερού του υποδένδρου, έπειτα τους κόμβους του δεξιού του υποδένδρου και στη συνέχεια τον ίδιο τον κόμβο.

Διαπεράσεις

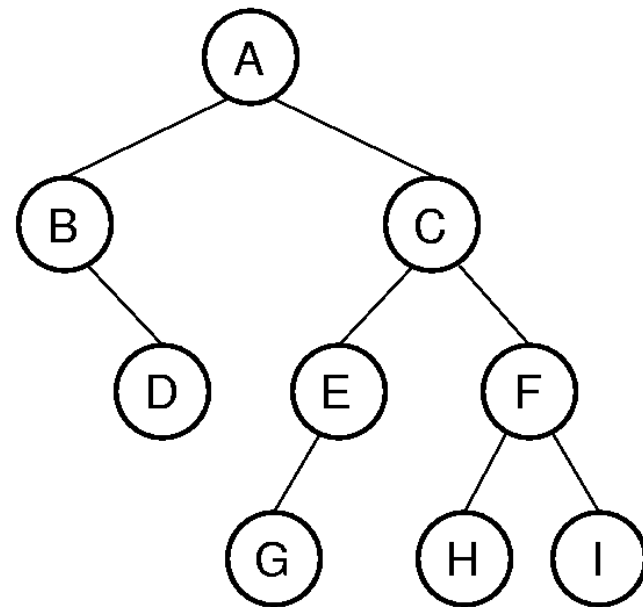
Ενδοδιατεταγμένη διάσχιση (inorder): Για κάθε κόμβο, επισκεπτόμαστε πρώτα τους κόμβους του αριστερού του υποδένδρου, έπειτα τον ίδιο τον κόμβο και στη συνέχεια τους κόμβους του δεξιού του υποδένδρου.

Κατά σειρά επιπέδων (level order): Επισκεπτόμαστε τους κόμβους κατά επίπεδα, από πάνω (τη ρίζα) προς τα κάτω, και μέσα σε ένα επίπεδο από αριστερά προς τα δεξιά.

Οι 3 πρώτες μέθοδοι είναι αναδρομικές ενώ η 4η είναι επαναληπτική.

Διαπεράσεις

- Προδιατεταγμένη:
A, B, D, C, E, G, F, H, I
- Μεταδιατεταγμένη:
D, B, G, E, H, I, F, C, A
- Ενδοδιατεταγμένη:
B, D, A, G, E, C, H, F, I
- Κατά σειρά επιπέδων:
A, B, C, D, E, F, G, H, I



Διαπεράσεις

```
void PreOrder (BinaryTreeNode *t) { // Preorder traversal of *t.  
    if (t) { Visit(t); PreOrder(t->LeftChild);  
        PreOrder(t->RightChild); }  
}
```

```
void InOrder (BinaryTreeNode *t) { // Inorder traversal of *t.  
    if (t) { InOrder(t->LeftChild);  
        Visit(t); InOrder(t->RightChild); }  
}
```

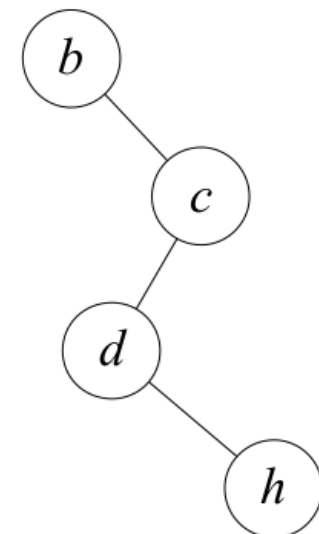
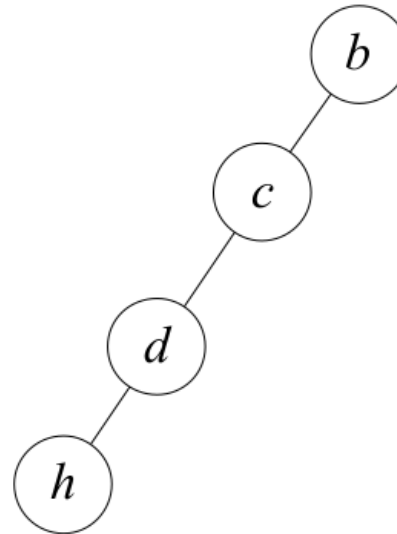
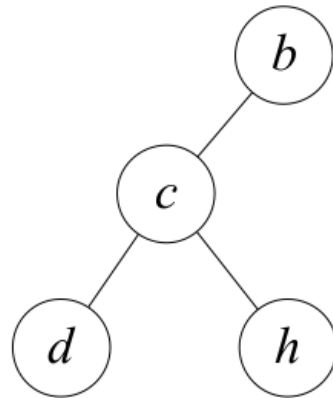
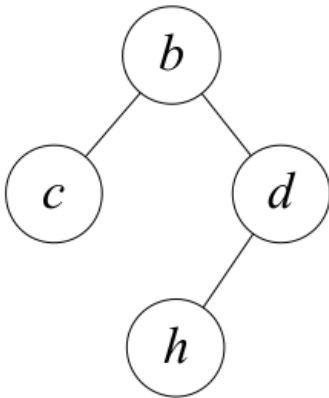
```
void PostOrder (BinaryTreeNode *t) { // Postorder traversal of *t.  
    if (t) { PostOrder(t->LeftChild);  
        PostOrder(t->RightChild); Visit(t); }  
}
```

Διαπεράσεις

```
void LevelOrder (BinaryTreeNode *t) { // Level-order traversal of *t.  
    LinkedQueue<BinaryTreeNode*> Q;  
    while (t) {  
        Visit(t); // visit t  
        // put t's children on queue  
        if (t->LeftChild) Q.Add(t->LeftChild);  
        if (t->RightChild) Q.Add(t->RightChild);  
        // get next node to visit  
        try {Q.Delete(t);}   
        catch (OutOfBounds) {return;}  
    }  
}
```

Διαπεράσεις

Δένδρα με την ίδια προδιάταξη (preorder): b, c, d, h



Ασκήσεις

Άσκηση 1

Με βάση το προηγούμενο αντιπαράδειγμα, αν μας δώσει κάποιος την προδιάταξη δεν μπορούμε να κατασκευάσουμε το δένδρο. Το ίδιο ισχύει για ενδοδιάταξη και μεταδιάταξη.

Ερώτηση: πόσες και ποιές διαπεράσεις χρειαζόμαστε για να προσδιορίσουμε μοναδικά το δένδρο; Να σχεδιαστεί ο αντίστοιχος αλγόριθμος.

Ασκήσεις

Άσκηση 2

Δίνεται δυαδικό δένδρο και έστω root ο pointer στη ρίζα του. Να σχεδιάσετε μία συνάρτηση που να βρίσκει το ύψος του δένδρου (το μήκος του μεγαλύτερου μονοπατιού από τη ρίζα προς κάποιο φύλλο).

Ασκήσεις

Άσκηση 3

Δίνεται δυαδικό δένδρο και έστω root ο pointer στη ρίζα του. Να σχεδιάσετε μία συνάρτηση που να ελέγχει αν το δένδρο είναι **πλήρες** ή όχι.

Ασκήσεις

Άσκηση 4

Δίνεται δυαδικό δένδρο T που αναπαριστά μία μαθηματική έκφραση. Πως θα πάρουμε την postfix αναπαράσταση της έκφρασης;

Ασκήσεις

Άσκηση 5

Δίδονται δύο δυαδικά δένδρα $T1$ και $T2$, και το $T1$ είναι κατά πολύ μεγαλύτερο από το $T2$. Να σχεδιάσετε αλγόριθμο που να αποφασίζει αν το $T2$ είναι υποδένδρο του $T1$.

THANKS FOR LISTENING



ANY QUESTIONS?

makeameme.org