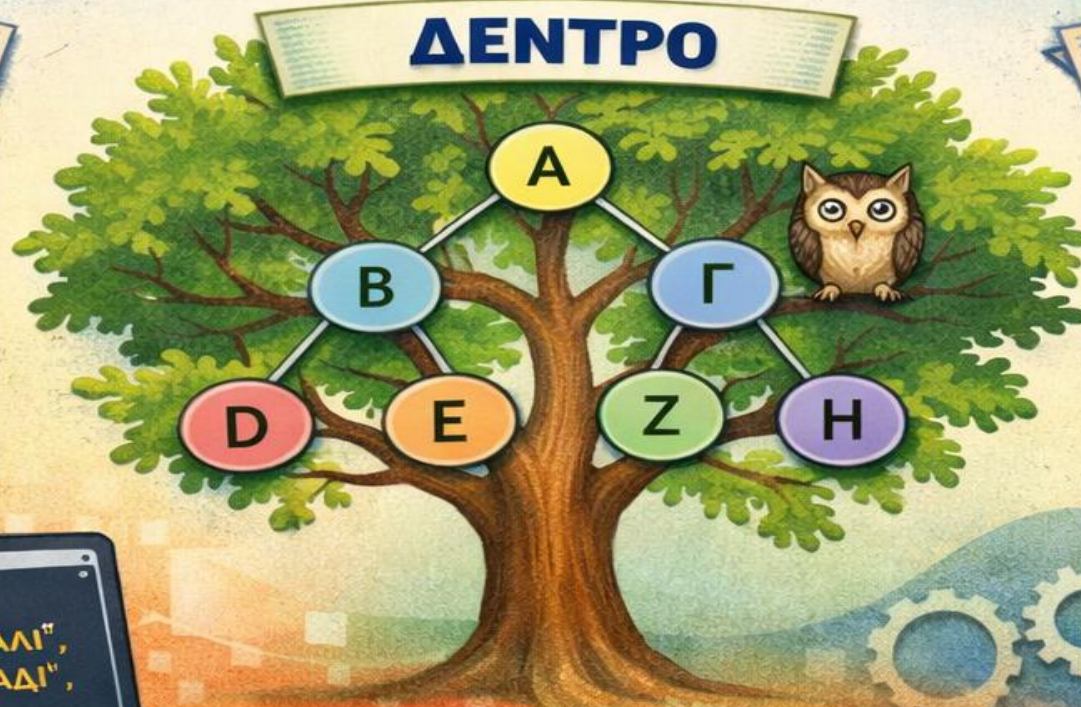


ΑΕΠΠ

ΛΙΣΤΑ

[0]	ΜΗΛΟ
[1]	ΠΟΡΤΟΚΑΛΙ
[2]	ΜΠΑΝΑΝΑ
[3]	ΑΧΛΑΔΙ
[4]	ΣΤΑΦΥΛΙ

ΔΕΝΤΡΟ



ΓΡΑΦΟΣ



```
fruits=  
  ="ΜΗΛΟ", "ΠΟΡΤΟΚΑΛΙ",  
  "ΜΠΑΝΑΝΑ", "ΑΧΛΑΔΙ",  
  "ΣΤΑΦΥΛΙ"]
```

ΛΙΣΤΕΣ, ΔΕΝΤΡΑ, ΓΡΑΦΟΙ ΕΝΟΤΗΤΑ 1.3

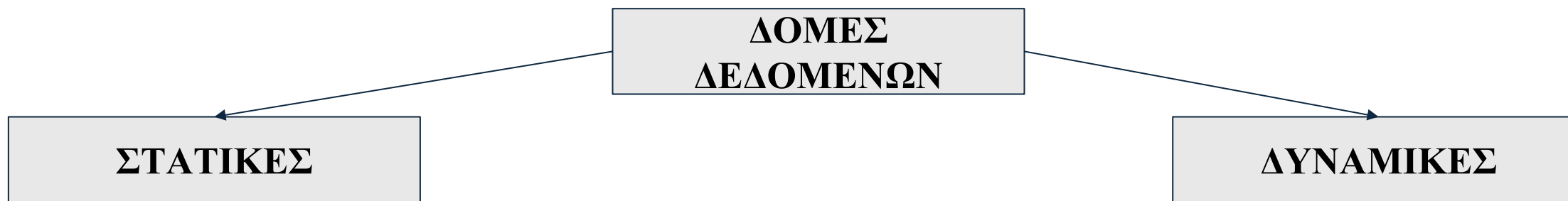
...ΑΛΓΟΡΙΘΜΟΙ + ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ = ΠΡΟΓΡΑΜΜΑΤΑ...

A) Στατικές δομές: Το ακριβές μέγεθος της απαιτούμενης κύριας μνήμης είναι συγκεκριμένο (σταθερό μέγεθος) και καθορίζεται κατά τη στιγμή του προγραμματισμού τους, και κατά συνέπεια κατά τη στιγμή της μετάφρασης τους και όχι κατά τη στιγμή της εκτέλεσης του προγράμματος. Τα στοιχεία των στατικών δομών αποθηκεύονται σε συνεχόμενες θέσεις μνήμης.

Προσοχή! Η λειτουργία της εισαγωγής και διαγραφής απαγορεύεται στις στατικές δομές δεδομένων καθώς αυξομειώνει το μέγεθος του πίνακα.

B) Δυναμικές δομές: Δεν αποθηκεύονται σε συνεχόμενες θέσεις μνήμης αλλά στηρίζονται στην τεχνική της δυναμικής παραχώρησης μνήμης. Με άλλα λόγια, οι δομές αυτές δεν έχουν σταθερό μέγεθος, αλλά ο αριθμός των κόμβων τους μεγαλώνει και μικραίνει καθώς στη δομή εισάγονται νέα δεδομένα ή διαγράφονται κάποια δεδομένα αντίστοιχα (το μέγεθος καθορίζεται κατά την εκτέλεση του προγράμματος). Όλες οι σύγχρονες γλώσσες προγραμματισμού προσφέρουν τη δυνατότητα δυναμικής παραχώρησης μνήμης.

➡ Για το μάθημά μας στη...ΓΛΩΣΣΑ



- ΠΙΝΑΚΕΣ
- ΣΤΟΙΒΑ
- ΟΥΡΑ

- ΛΙΣΤΕΣ
- ΔΕΝΔΡΑ
- ΓΡΑΦΟΙ

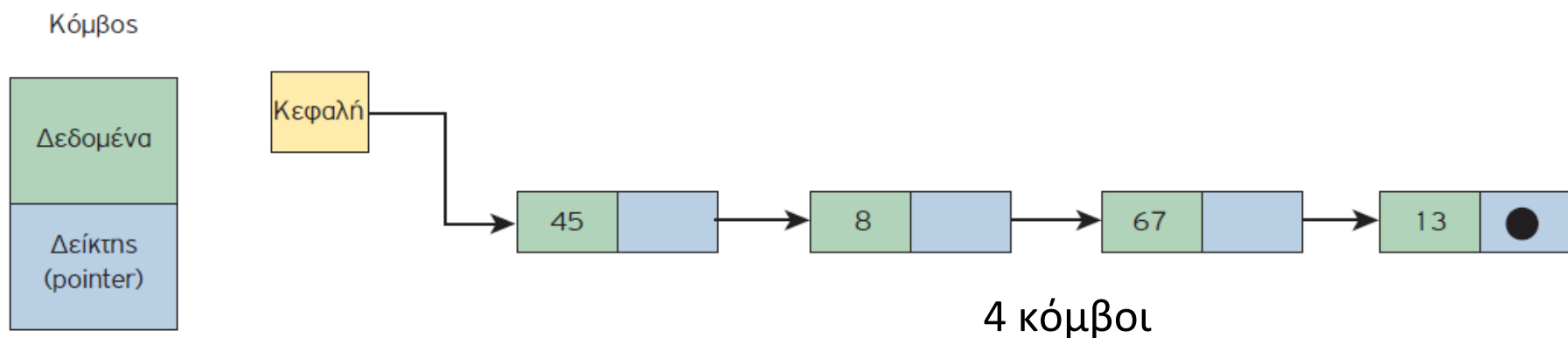
1.3.1 ΛΙΣΤΕΣ

- Η έννοια της λίστας συναντάται αρκετά συχνά στην καθημερινότητά μας όπως τραγούδια, τις επαφές μας, τα ψώνια που θέλουμε να κάνουμε.
- Η λίστα δεν είναι τίποτα άλλο παρά μία *συλλογή από αντικείμενα του ίδιου τύπου*. Μπορούμε να έχουμε δηλαδή λίστες από λέξεις, από ονόματα αλλά και από αριθμούς.
- Μπορούμε οποιαδήποτε στιγμή να *προσθέσουμε* ή να *αφαιρέσουμε* χωρίς να μας απασχολεί το μέγεθος ή κενές θέσεις.

1.3.1 ΛΙΣΤΕΣ

Η συνδεδεμένη λίστα αποτελείται από μία σειρά από κόμβους, που συνήθως βρίσκονται σε απομακρυσμένες θέσεις μνήμης. Κάθε κόμβος αποτελείται από δύο κύρια τμήματα. Το πρώτο τμήμα περιέχει τα δεδομένα και το δεύτερο τμήμα φιλοξενεί τη διεύθυνση του επόμενου κόμβου με τον οποίο συνδέεται ή όπως αλλιώς θα λέγαμε στη γλώσσα των δομών δεδομένων, το δεύτερο τμήμα περιέχει έναν δείκτη (pointer) που δείχνει στον επόμενο κόμβο.

Το πεδίο Δεδομένα μπορεί να περιέχει μία ή περισσότερες αλφαριθμητικές ή αριθμητικές πληροφορίες. Ο δείκτης (pointer) είναι ένας ιδιαίτερος τύπος δεδομένων που προσφέρεται από τις περισσότερες σύγχρονες γλώσσες προγραμματισμού. Ο δείκτης δε λαμβάνει αριθμητικές τιμές όπως ακέραιες, πραγματικές κ.ά., αλλά οι τιμές του είναι διευθύνσεις στην κύρια μνήμη και χρησιμοποιείται ακριβώς για τη σύνδεση των διαφόρων στοιχείων μιας δομής, που είναι αποθηκευμένα σε μη συνεχόμενες θέσεις μνήμης.



1.3.1 ΛΙΣΤΕΣ

Μία (**απλά**) **συνδεδεμένη λίστα** (linked list) είναι ένα σύνολο κόμβων διατεταγμένων γραμμικά (ο ένας μετά τον άλλο). Κάθε κόμβος περιέχει εκτός από τα **δεδομένα** του και έναν **δείκτη** που δείχνει προς τον επόμενο κόμβο. Ο δείκτης του τελευταίου κόμβου δε δείχνει σε κάποιον κόμβο (δείκτης στο κενό). Για να το δηλώσουμε αυτό λέμε ότι το πεδίο δείκτη του τελευταίου κόμβου έχει την τιμή **NULL**.

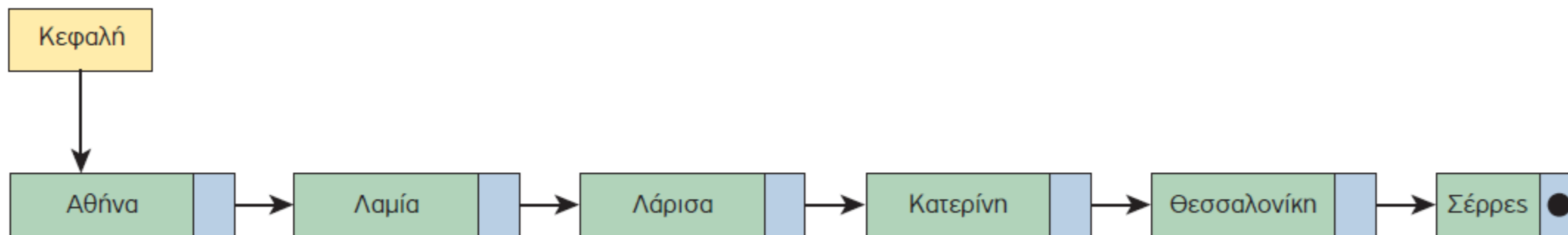
Για να προσπελάσουμε τους κόμβους της λίστας χρειάζεται να γνωρίζουμε τη διεύθυνση (θέση στη μνήμη) του πρώτου κόμβου της λίστας. Η διεύθυνση αυτή αποθηκεύεται σε μία ειδική μεταβλητή που την ονομάζουμε συνήθως **Κεφαλή** (Head).



5 κόμβοι

Πρόσβαση στους κόμβους μιας συνδεδεμένης λίστας

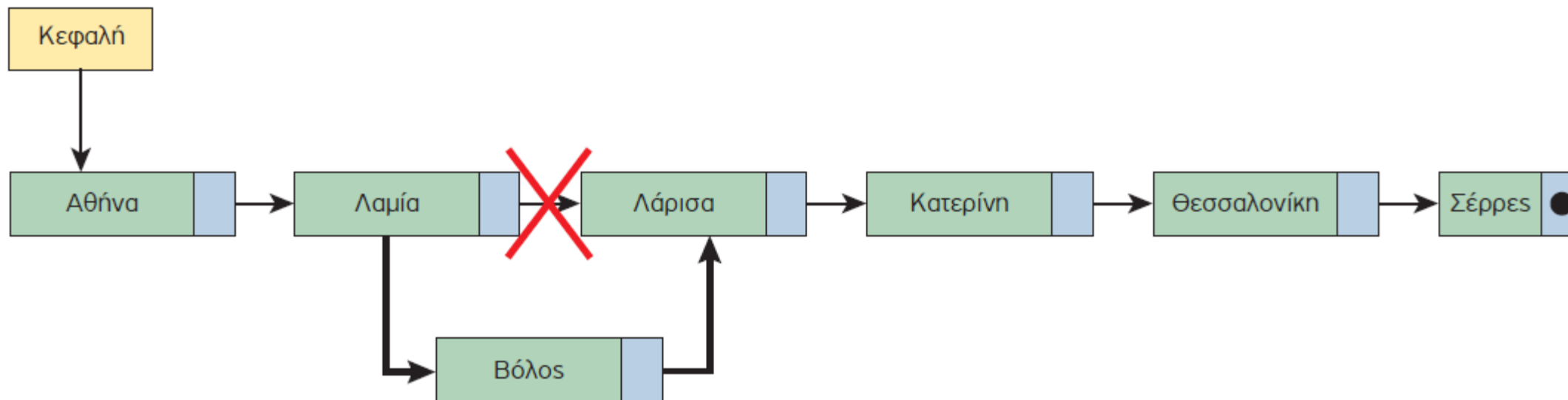
Οι κόμβοι μιας (απλά) συνδεδεμένης λίστας είναι διατεταγμένοι σε μια συγκεκριμένη σειρά, χωρίς αυτό να σημαίνει ότι αποθηκεύονται σε συνεχόμενες θέσεις στη μνήμη. Αντίθετα, είναι διασκορπισμένοι σε όλη τη μνήμη και η σύνδεση μεταξύ τους γίνεται μέσω των δεικτών. Έχουμε άμεση πρόσβαση μόνο στον πρώτο κόμβο της λίστας. Επομένως, για να εντοπίσουμε κάποιον από τους ενδιάμεσους κόμβους, **πρέπει να ξεκινήσουμε από τον πρώτο κόμβο της λίστας** και να ακολουθήσουμε τους δείκτες με τη σειρά, μέχρι να φτάσουμε στον επιθυμητό κόμβο.



Αν θέλω να επισκεφτώ τη Λάρισα θα πρέπει να προσπελάσω υποχρεωτικά την Αθήνα και τη Λαμία, δηλαδή τους 2 πρώτους κόμβους πριν φτάσω στον τρίτο.

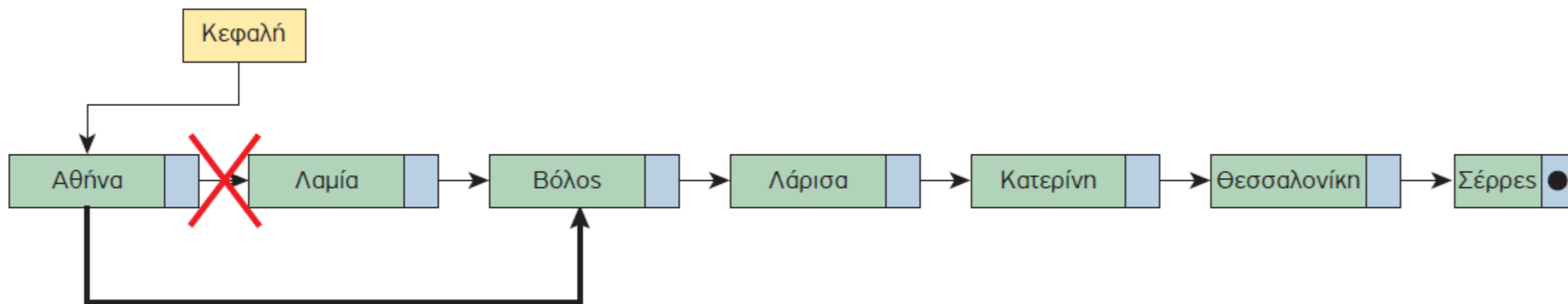
Εισαγωγή νέου κόμβου στη λίστα

Για την εισαγωγή (παρεμβολή) του νέου κόμβου ο δείκτης του δεύτερου κόμβου δείχνει τον νέο κόμβο και ο δείκτης του νέου κόμβου δείχνει τον τέταρτο κόμβο (δηλαδή, παίρνει την τιμή που είχε πριν την εισαγωγή ο δείκτης του δεύτερου κόμβου).



Διαγραφή κόμβου από τη λίστα

Για τη διαγραφή ενός κόμβου αρκεί να αλλάξει τιμή ο δείκτης του προηγούμενου κόμβου και να δείχνει πλέον στον επόμενο αυτού που διαγράφεται. Ο κόμβος που διαγράφηκε (ο δεύτερος) αποτελεί «άχρηστο δεδομένο» και ο χώρος μνήμης που καταλάμβανε, παραχωρείται για άλλη χρήση.

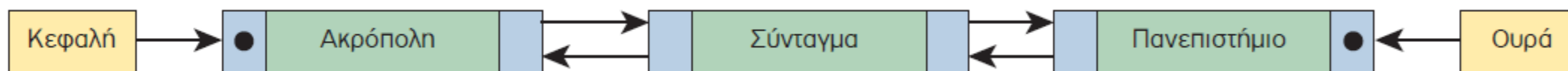


Διπλά συνδεδεμένη λίστα

Μια **διπλά συνδεδεμένη λίστα** (doubly linked list) μπορούμε να τη διατρέξουμε και προς τις δύο κατευθύνσεις. Η χρήση του δεύτερου δείκτη προσφέρει τη δυνατότητα ξεκινώντας από οποιοδήποτε κόμβο της λίστας να μπορούμε να διαβάσουμε τη λίστα και προς τις δυο κατευθύνσεις.

Κάθε κόμβος της διπλά συνδεδεμένης λίστας, συνδέεται με τον αμέσως επόμενο και τον αμέσως προηγούμενο κόμβο της λίστας. Εκτός, βέβαια, από τον αρχικό και τον τελευταίο κόμβο της λίστας.

Σε μια διπλά συνδεδεμένη λίστα διευκολύνεται η ταξινόμηση και η αναζήτηση, ωστόσο, αυξάνεται η πολυπλοκότητα στη διαχείριση των κόμβων, καθώς απαιτείται επιπλέον χώρος για τον δεύτερο δείκτη (επιπρόσθετη μνήμη για κάθε κόμβο).



Βασικές πράξεις των συνδεδεμένων λιστών

- **Εισαγωγή** κόμβου στη λίστα (εισαγωγή κόμβου στην αρχή, στο τέλος της λίστας ή ενδιάμεσα).
- **Διαγραφή** κόμβου από τη λίστα (διαγραφή από την αρχή, το τέλος της λίστας ή ενδιάμεσα).
- **Έλεγχος** για το αν η λίστα είναι κενή.
- **Αναζήτηση** κόμβου για την εύρεση συγκεκριμένου στοιχείου.
- **Διάσχιση** της λίστας και προσπέλαση των στοιχείων της (π.χ. εκτύπωση των δεδομένων που περιέχονται σε όλους τους κόμβους της λίστας).

Διαφορές Λίστας σε σχέση με τον Πίνακα

- Ο πίνακας θεωρείται μια δομή τυχαίας προσπέλασης, σε αντίθεση με μια λίστα που είναι στην ουσία μια δομή ακολουθιακής ή σειριακής προσπέλασης. Για να φθάσουμε, δηλαδή, σ' έναν κόμβο μιας λίστας πρέπει να περάσουμε από όλους τους προηγούμενους ξεκινώντας από τον πρώτο.
- Ο πίνακας έχει σταθερό μέγεθος, το οποίο δηλώνεται εξαρχής κατά την υλοποίηση. Αυτό γίνεται, διότι ο πίνακας είναι στατική δομή δεδομένων σε αντίθεση με τη λίστα που είναι δυναμική δομή και το μέγεθός της μπορεί να μεταβάλλεται καθώς εισέρχονται νέοι κόμβοι στη λίστα ή διαγράφονται κάποιοι άλλοι.
- Οι κόμβοι της λίστας αποθηκεύονται σε μη συνεχόμενες θέσεις μνήμης σε αντιδιαστολή με τους πίνακες, όπου τα στοιχεία αποθηκεύονται σε συνεχόμενες θέσεις μνήμης.

Πλεονεκτήματα- Μειονεκτήματα

Στα πλεονεκτήματα των λιστών (έναντι των πινάκων) συγκαταλέγονται τα εξής:

- Το δυναμικό τους μέγεθος,
- η ευκολία εισαγωγής και διαγραφής από οποιοδήποτε μέρος της λίστας, καθώς και
- η μη αναγκαιότητα δήλωσης του μεγέθους τους.

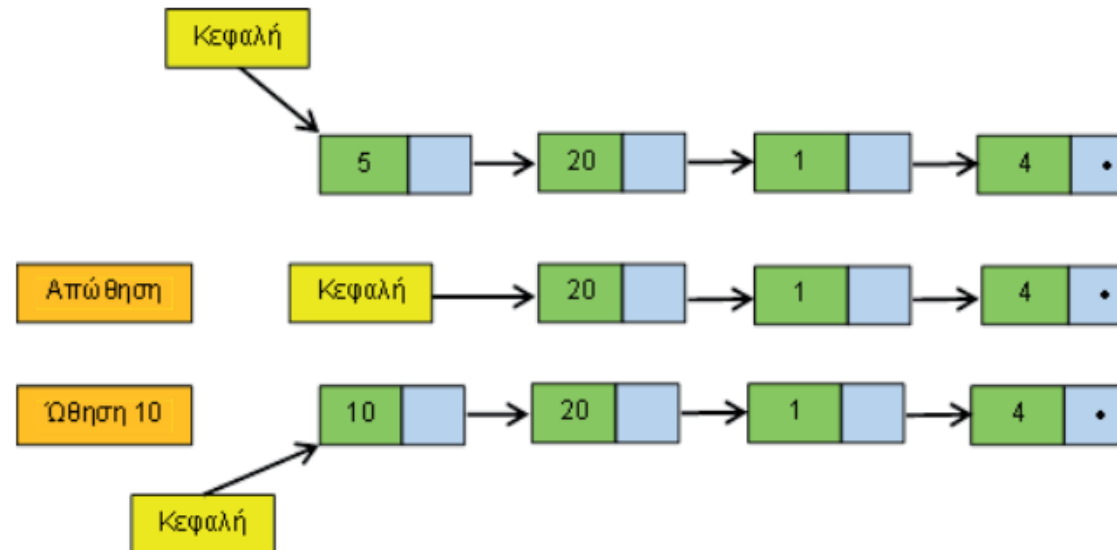
Στα μειονεκτήματα των λιστών (έναντι των πινάκων) περιλαμβάνονται τα εξής:

- Η τυχαία πρόσβαση στη λίστα δεν επιτρέπεται. Είναι αδύνατο να φτάσετε στον n-οστό κόμβο μιας απλά συνδεδεμένης λίστας χωρίς πρώτα να περάσετε από όλους τους κόμβους διαδοχικά μέχρι τον συγκεκριμένο κόμβο ξεκινώντας από τον πρώτο κόμβο. Εναλλακτικά, στην περίπτωση της διπλά συνδεδεμένης λίστας μπορείτε να ξεκινήσετε και από τον τελευταίο κόμβο. Επομένως, δεν μπορούμε να πραγματοποιήσουμε με αποτελεσματικό τρόπο δυαδική αναζήτηση σε συνδεδεμένες λίστες.
- Οι συνδεδεμένες λίστες έχουν πολύ μεγαλύτερη επιβάρυνση από τους πίνακες, αφού οι συνδεδεμένοι κόμβοι της λίστας είναι δυναμικά κατανομημένοι (οι οποίοι είναι λιγότερο αποτελεσματικοί στη χρήση της μνήμης) και κάθε κόμβος στη λίστα πρέπει, επιπλέον, να αποθηκεύσει έναν πρόσθετο δείκτη που θα δείχνει στον επόμενο κόμβο. Στην περίπτωση των διπλά συνδεδεμένων λιστών χρειαζόμαστε επιπλέον έναν δεύτερο δείκτη που θα δείχνει στον προηγούμενο κόμβο.

ΑΣΚΗΣΗ 1. Οι συνδεδεμένες λίστες αξιοποιούνται για την υλοποίηση της στοίβας και της ουράς, λόγω της δυνατότητας αυξομείωσης του μεγέθους τους. Μπορείτε να εξηγήσετε γιατί;

Η στοίβα μπορεί να υλοποιηθεί με μία απλά συνδεδεμένη λίστα.

Εάν υποθέσουμε ότι οι κόμβοι (στοιχεία) εισέρχονται από την αρχή της απλά συνδεδεμένης λίστας και εξέρχονται πάλι από την αρχή της λίστας τότε ο κόμβος (στοιχείο) που εισήλθε τελευταίος εξέρχεται και πρώτος (LIFO). Επομένως η στοίβα μπορεί να υλοποιηθεί με μία απλά συνδεδεμένη λίστα αφού μπορεί να γίνει εισαγωγή κόμβου (στοιχείου) στην αρχή της λίστας και διαγραφή κόμβου (στοιχείου) από την αρχή της λίστας. Καλύτερη είναι η υλοποίηση στοίβας με συνδεδεμένη λίστα εκτός αν γνωρίζουμε εκ των προτέρων τον μέγιστο αριθμό στοιχείων και θέλουμε μια εύκολη και γρήγορη λύση, οπότε επιλέγουμε την υλοποίηση με πίνακα.





Συνέχεια...

Στην ουρά, από την άλλη πλευρά, έχουμε τις εξής δύο κύριες λειτουργίες. Εισαγωγή στοιχείου στο πίσω άκρο της ουράς και εξαγωγή στοιχείου από το εμπρός άκρο της ουράς. Η ουρά μπορεί να υλοποιηθεί με μία διπλά συνδεδεμένη λίστα, αφού μπορούμε να κάνουμε εισαγωγή κόμβου στο τέλος της διπλά συνδεδεμένης λίστας (εισαγωγή στοιχείου στο πίσω άκρο της ουράς). Επίσης μπορούμε να κάνουμε διαγραφή κόμβου στην αρχή της διπλά συνδεδεμένης λίστας (εξαγωγή στοιχείου από το εμπρός άκρο της ουράς).

ΑΣΚΗΣΗ 2. Εντοπίστε τις διαφορές στην προσπέλαση όσον αφορά τη λίστα, τη στοίβα και την ουρά. Για την απάντησή σας σκεφτείτε αν η εισαγωγή ενός στοιχείου σε μία στοίβα μπορεί να γίνει σε οποιαδήποτε θέση της.

Στη στοίβα μπορεί να εισέλθει (ώθηση) στοιχείο μόνο στην κορυφή.

Στην ουρά ένα στοιχείο μπορεί να εισέλθει μόνο στο πίσω άκρο.

Στη λίστα ένα στοιχείο (κόμβος) μπορεί να εισέλθει σε οποιαδήποτε θέση (αρχή, τέλος ή ενδιάμεσα).

Στη στοίβα ένα στοιχείο μπορεί να εξέλθει μόνο από την κορυφή.

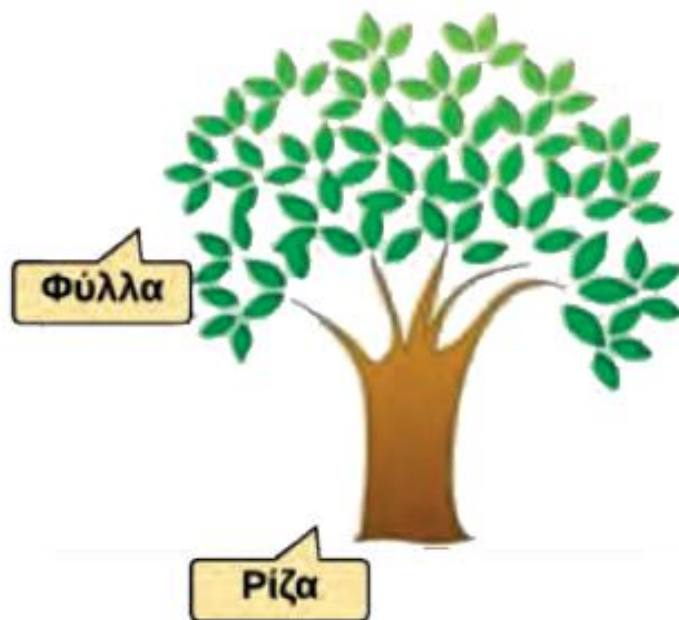
Στην ουρά ένα στοιχείο μπορεί να εξέλθει μόνο από την αρχή της.

Στη λίστα ένα στοιχείο (κόμβος) μπορεί να διαγραφεί από οποιοδήποτε σημείο της (αρχή, τέλος ή ενδιάμεσα).

1.3.2 ΔΕΝΔΡΑ

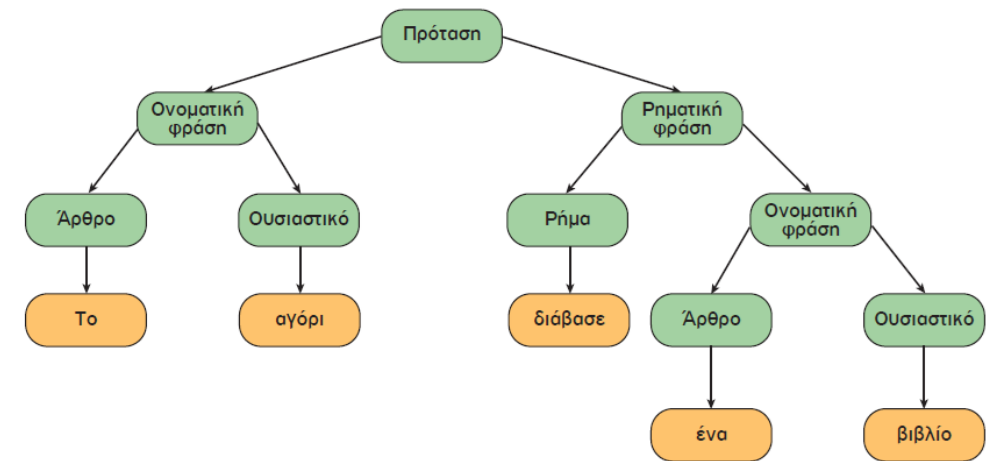
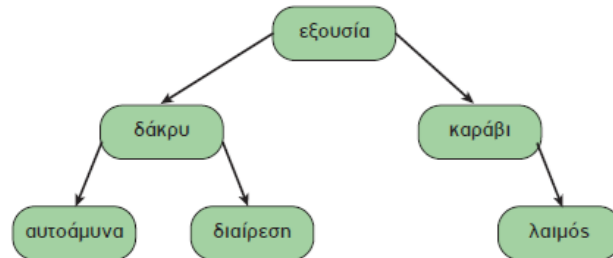
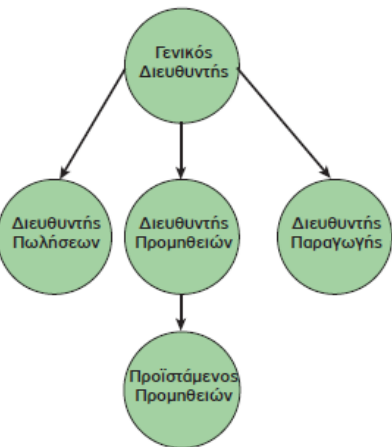
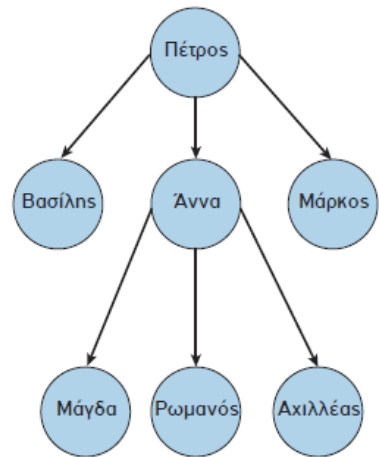
-Η μη γραμμική δομή των δένδρων χρησιμοποιείται σε περιπτώσεις όπου μετά από ένα στοιχείο ακολουθεί όχι ένα, αλλά δύο, τρία ή και περισσότερα στοιχεία.

-Στην επιστήμη της Πληροφορικής, όμως, βλέπουμε τα δένδρα ανάποδα. Τα φύλλα είναι στο κάτω μέρος και η ρίζα στο επάνω μέρος.



Παραδείγματα δένδρων

- ✓ Τα δένδρα χρησιμοποιούνται σε πολλούς τομείς της επιστήμης των υπολογιστών, συμπεριλαμβανομένων των λειτουργικών συστημάτων, των γραφικών, των συστημάτων βάσεων δεδομένων, των παιχνιδιών, της τεχνητής νοημοσύνης και της δικτύωσης υπολογιστών.
- ✓ Παραδείγματα δένδρων: το οικογενειακό δένδρο, η δομή ενός οργανισμού ή μιας εταιρείας, ο πίνακας περιεχομένων ενός βιβλίου, τα αρχεία και οι φάκελοι ενός υπολογιστή, ένα λεξικό, τα μέρη που απαρτίζουν την μηχανή ενός αυτοκινήτου, τα συστατικά μιας πρότασης.



α. Οικογενειακό δένδρο

β. Οργανόγραμμα μιας εταιρείας

γ. Οργάνωση ενός λεξικού

Η δομή μιας πρότασης

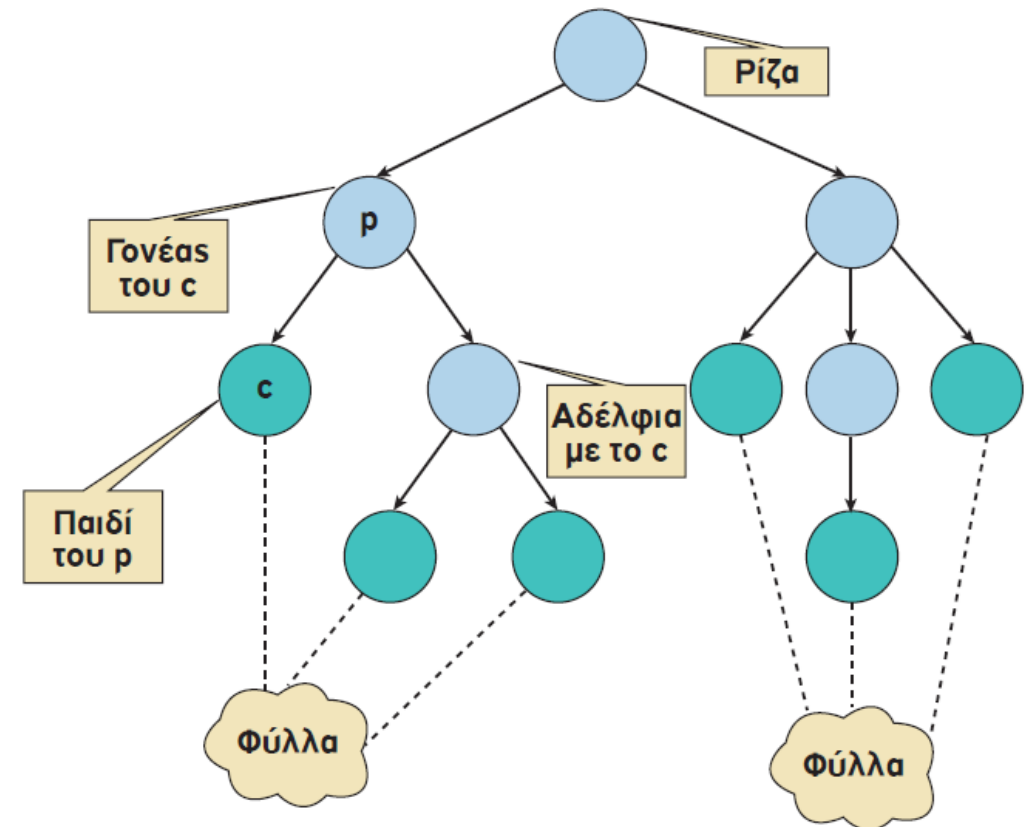
1.3.2 ΔΕΝΔΡΑ

Ένα δένδρο αποτελείται από **κόμβους**, οι οποίοι συνδέονται μεταξύ τους με **ακμές**. Όταν δύο κόμβοι συνδέονται μεταξύ τους με μία ακμή, τότε ονομάζουμε «**γονέα**» τον κόμβο από τον οποίο ξεκινάει η ακμή και «**παιδί**» τον κόμβο στον οποίο καταλήγει η ακμή. Στην Εικόνα 1 ο κόμβος p είναι γονέας του κόμβου c και ο κόμβος c είναι παιδί του κόμβου p .

Ένας κόμβος μπορεί να έχει κανένα, ένα ή περισσότερα παιδιά.

Όλοι οι κόμβοι, εκτός από έναν, έχουν ακριβώς έναν γονέα. Ο κόμβος χωρίς γονέα ονομάζεται «**ρίζα**» (root) και βρίσκεται στην κορυφή του δένδρου. Κόμβοι με τον ίδιο γονέα ονομάζονται «**αδέλφια**». Οι κόμβοι χωρίς παιδιά ονομάζονται «**φύλλα**».

Μπορούμε να έχουμε ένα **απλό δένδρο**, το οποίο να απαρτίζεται από έναν μόνο κόμβο. Αυτός ο κόμβος είναι και ρίζα του απλού αυτού δένδρου, διότι δεν έχει γονέα και φύλλο, και διότι δεν έχει παιδιά.



1.3.2 ΔΕΝΔΡΑ

Ένα **δένδρο** (tree) είναι μία δομή που αποτελείται από ένα σύνολο κόμβων και ένα σύνολο ακμών μεταξύ των κόμβων με βάση τους εξής κανόνες:

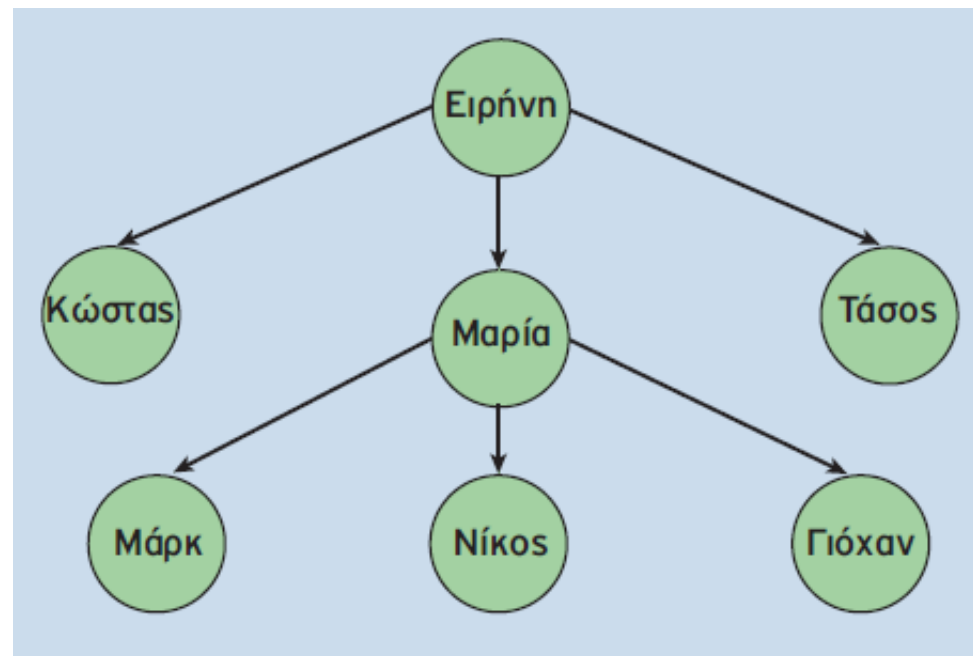
- Υπάρχει ένας ξεχωριστός κόμβος που ονομάζεται **ρίζα**. Αυτός είναι ένας κόμβος χωρίς γονέα.
- Για κάθε κόμβο c , εκτός από τη ρίζα, υπάρχει μόνο μια ακμή που καταλήγει στον κόμβο αυτόν ξεκινώντας από κάποιον άλλον κόμβο p . Ο κόμβος p ονομάζεται γονέας του c και ο κόμβος c παιδί του p .
- Για κάθε κόμβο υπάρχει μία μοναδική διαδρομή, δηλαδή, μια ακολουθία διαδοχικών ακμών, που ξεκινάει από τη ρίζα και τερματίζει σε αυτόν τον κόμβο.

Δένδρο θεωρούμε και το **κενό δένδρο**, δηλαδή το δένδρο που δεν έχει ούτε κόμβους, ούτε ακμές. Το κενό δένδρο είναι το μόνο δένδρο χωρίς ρίζα.

ΑΣΚΗΣΗ

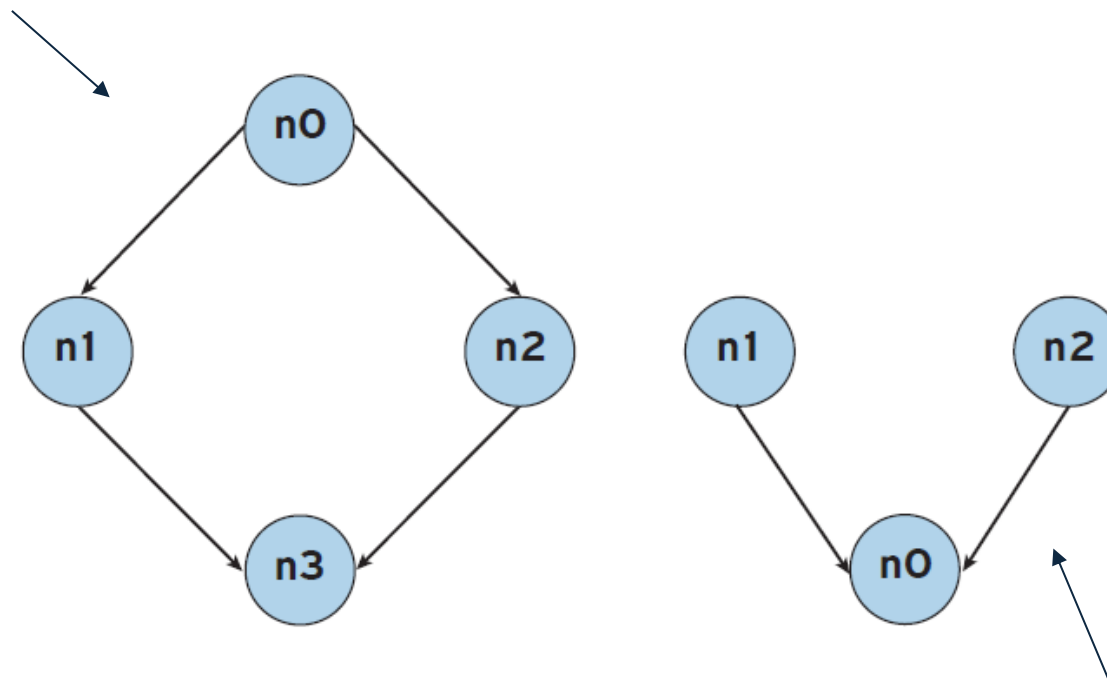
Στο δένδρο της εικόνας βρείτε ποιος/ ποιοι κόμβος/κόμβοι είναι:

- ο γονέας του Τάσου
- τα φύλλα του δένδρου
- η ρίζα του δένδρου
- τα παιδιά της Μαρίας
- ποια είναι αδέρφια



Γιατί δεν είναι δένδρα;

Δεν είναι δένδρο επειδή ο κόμβος n_3 έχει δύο γονείς, τους n_1 και n_2 και, όπως ξέρουμε, σε ένα δένδρο ένας κόμβος πρέπει να έχει ακριβώς έναν γονέα, με εξαίρεση τη ρίζα, που δεν έχει κανέναν. Επίσης, υπάρχουν δύο διαδρομές από την ρίζα n_0 προς τον κόμβο n_3 , πράγμα που και αυτό απαγορεύεται σε ένα δένδρο.

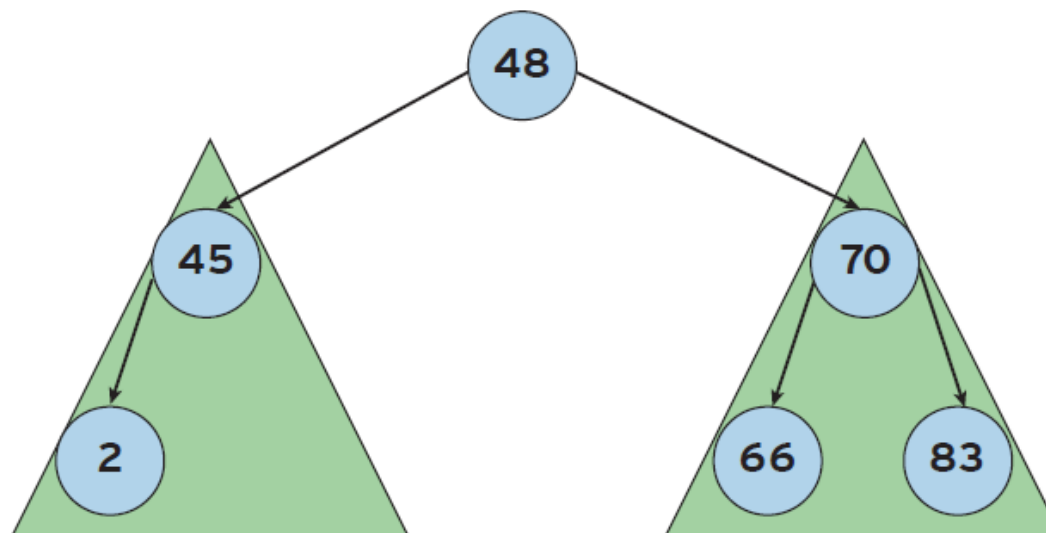


Δεν είναι δένδρο διότι υπάρχουν δύο κόμβοι χωρίς γονέα, οι n_1 και n_2 . Είναι σαν να λέμε ότι έχουμε δύο ρίζες στο δένδρο αυτό. Γνωρίζουμε όμως ότι η ρίζα είναι μοναδική σε ένα δένδρο.

Υποδένδρα

Μέσα σε ένα δένδρο μπορούμε να εντοπίσουμε και άλλα μικρότερα δένδρα, που ονομάζονται υποδένδρα. Πιο συγκεκριμένα, κάθε κόμβος ενός δένδρου μπορεί να θεωρηθεί ως ρίζα ενός υποδένδρου, δηλαδή ενός άλλου μικρότερου δένδρου, που ξεκινάει από τον κόμβο αυτόν.

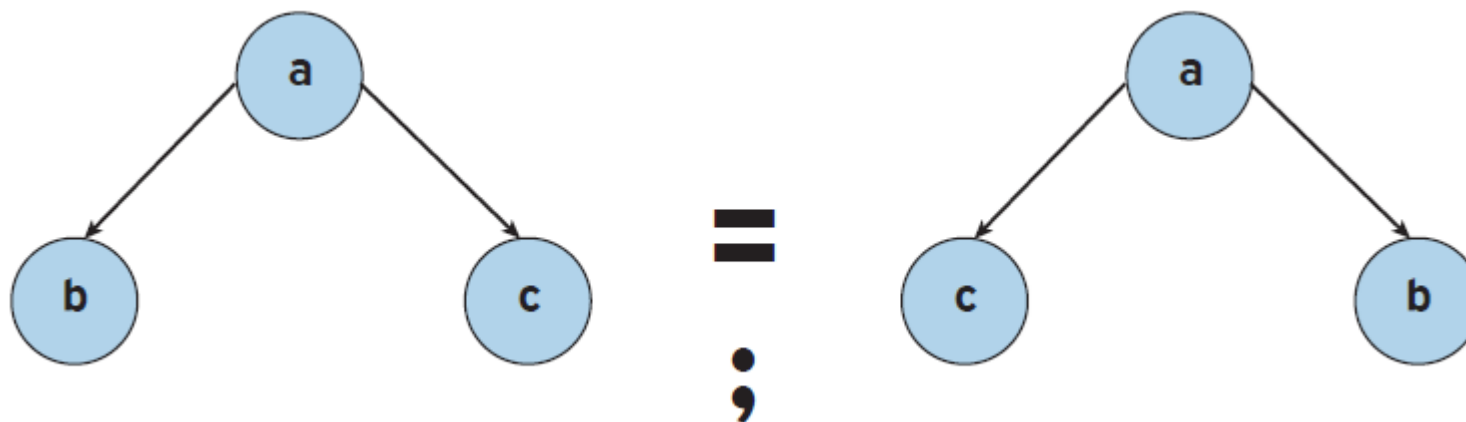
Στο δένδρο της εικόνας ο κόμβος 48 είναι ρίζα και έχει δύο υποδένδρα που ξεκινούν από τους κόμβους 45 και 70 αντίστοιχα. Ο κόμβος 45 έχει ένα υποδένδρο που αποτελείται από τον κόμβο 2. Ο κόμβος 70 έχει δύο υποδένδρα που αποτελούνται από τους κόμβους 66 και 83 αντίστοιχα. Τα υποδένδρα των κόμβων 2, 66 και 83 είναι κενά.



Πότε δύο δένδρα είναι ίδια;

Δεν έχει πάντα σημασία η σειρά των κόμβων.

Για παράδειγμα, αν θέλουμε να μοντελοποιήσουμε την ιεραρχική σχέση των μελών μιας οικογένειας και μας ενδιαφέρει να οργανώσουμε τα αδέλφια σύμφωνα με την ηλικία τους, τότε τα αδέλφια που θα έχουν γεννηθεί νωρίτερα θα τοποθετηθούν στην δενδρική δομή πιο αριστερά σε σχέση με αυτά που θα έχουν γεννηθεί αργότερα. Σε αυτή την περίπτωση, που για κάθε κόμβο υπάρχει μία γραμμική σχέση μεταξύ των παιδιών του κόμβου αυτού, αναφερόμαστε σε ένα **διατεταγμένο δένδρο**.

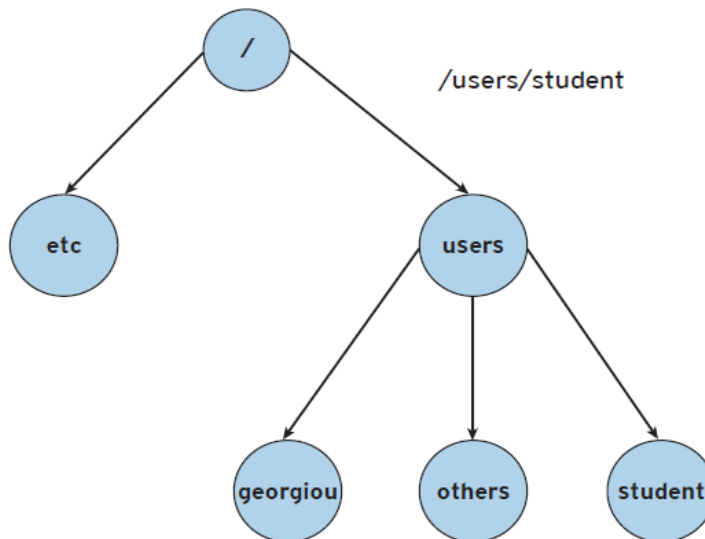


Τα δένδρα ως μη-γραμμική ευέλικτη δομή

Υπάρχουν δύο λόγοι για τους οποίους τα δένδρα είναι τόσο ισχυρά.

Ο πρώτος λόγος αναφέρεται στη **δυναμικότητα** των δένδρων. Είναι πολύ εύκολο να προσθέσετε, να αφαιρέσετε ή να αναζητήσετε ένα στοιχείο σε ένα δένδρο.

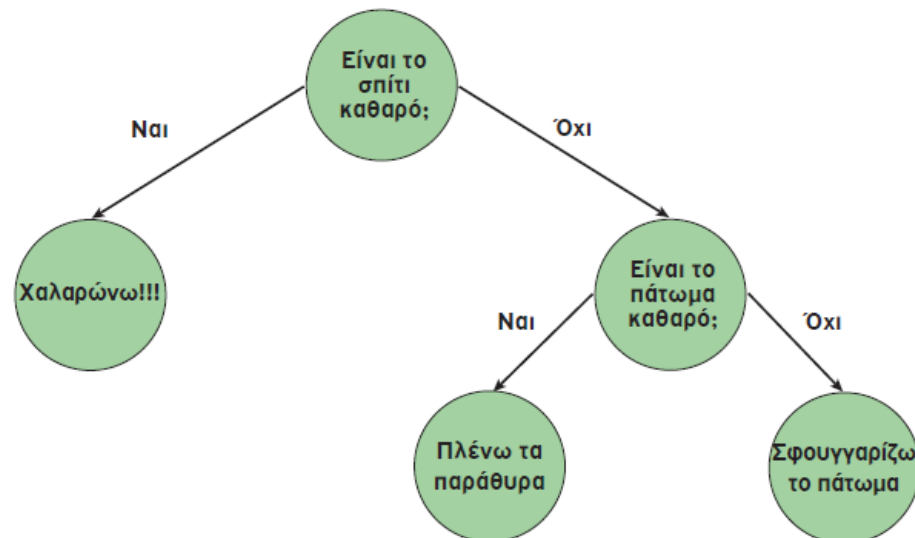
Ο δεύτερος βασικός λόγος είναι ότι η δομή των δένδρων **μεταφέρει πληροφορίες**. Για παράδειγμα, ας θεωρήσουμε το σύστημα αρχείων του υπολογιστή, όπως φαίνεται στην εικόνα. Είναι πολύ εύκολο να προσθέσετε ένα νέο κατάλογο για τον καθηγητή “georgiou”. Επίσης, λόγω του ότι ο κατάλογος “users” είναι παιδί της ρίζας “/” και ότι ο κατάλογος “student” είναι παιδί του “users” μπορούμε να συμπεράνουμε ότι υπάρχει η διαδρομή “/users/student”.



Δένδρα Απόφασης

Τα δένδρα αποτελούν τη βάση αρκετών αλγορίθμων επίλυσης προβλήματος, όπως είναι για παράδειγμα η συμπίεση εικόνων, η ταξινόμηση, η αυτόματη συμπλήρωση λέξεων σε συσκευές κινητών τηλεφώνων, η μεταγλώττιση ενός προγράμματος και η λήψη αποφάσεων.

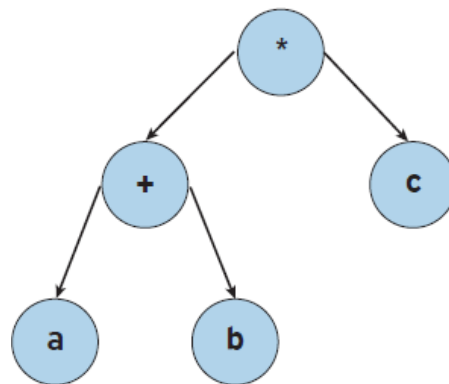
Τα δένδρα απόφασης είναι δένδρα στα οποία κάθε κόμβος αντιπροσωπεύει ένα χαρακτηριστικό (ιδιότητα), κάθε ακμή αντιπροσωπεύει μια απόφαση (κανόνα) και κάθε φύλλο αντιπροσωπεύει ένα αποτέλεσμα. Στους αλγορίθμους μηχανικής μάθησης (machine learning) τα δένδρα απόφασης έχουν πρωτεύοντα ρόλο.



Άλλα Δένδρα

Για τα παιχνίδια στον υπολογιστή, όπως είναι το σκάκι, η τρίλιζα, το τάβλι και πολλά άλλα, ο υπολογιστής χρησιμοποιεί ένα ειδικό δένδρο, που ονομάζεται **δένδρο του παιχνιδιού** (game tree), το οποίο μοντελοποιεί όλες τις πιθανές κινήσεις των παικτών για να σας νικήσει. Θα μπορούσε κάθε κόμβος στο δένδρο, που αντιπροσωπεύει μία συγκεκριμένη κατάσταση παιχνιδιού, να περιέχει πληροφορίες σχετικά με το ποιος παίκτης έχει τη μεγαλύτερη πιθανότητα να κερδίσει από οποιαδήποτε πιθανή κίνηση.

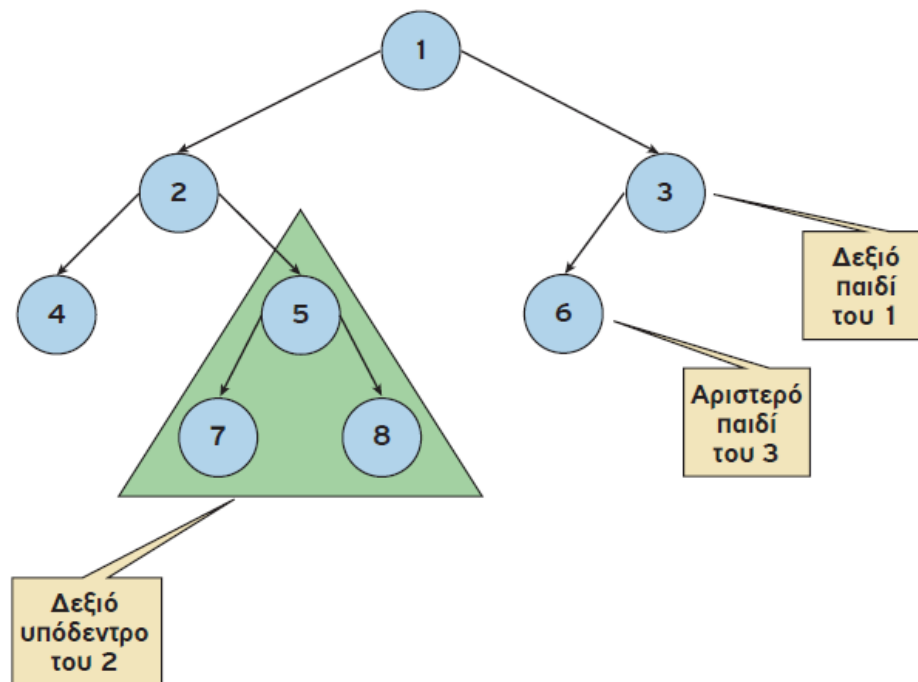
Διαδεδομένα είναι επίσης τα δένδρα για την αναπαράσταση και κατ' επέκταση τον υπολογισμό **αριθμητικών εκφράσεων**.



$(a+b)*c$

Δυαδικά Δένδρα

Ένα **δυαδικό δένδρο** (binary tree) είναι ένα διατεταγμένο δένδρο, στο οποίο κάθε κόμβος έχει το πολύ δύο παιδιά, το αριστερό και το δεξί παιδί. Μπορούμε, συνεπώς, να μιλάμε για αριστερό και δεξιό υποδένδρο ενός κόμβου. Στο δυαδικό δένδρο της εικόνας, ο κόμβος 3 έχει ως αριστερό υποδένδρο, το δένδρο με μοναδικό κόμβο το 6 και ως δεξιό υποδένδρο, το κενό δένδρο. Προφανώς, αν ανταλλάξουμε το αριστερό με το δεξιό υποδένδρο ενός κόμβου παίρνουμε ένα διαφορετικό δένδρο.



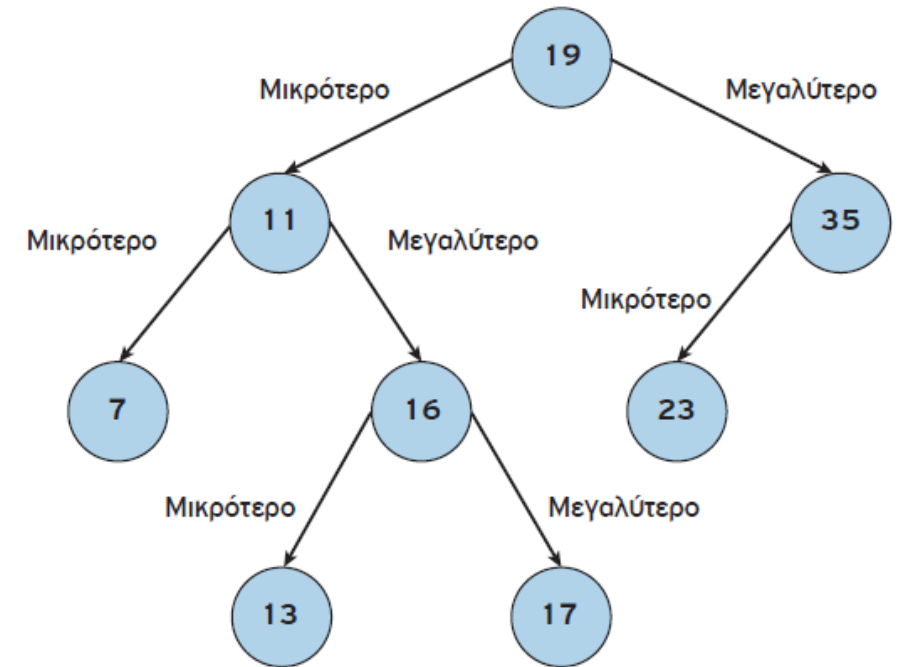
Δυαδικά Δένδρα Αναζήτησης

Οι αλγόριθμοι της αναζήτησης, για παράδειγμα όλων των email που έχετε λάβει ή στείλει σε μία συγκεκριμένη περίοδο ή της εύρεσης όλων των λέξεων, που αρχίζουν από την συμβολοσειρά «πληρ», αξιοποιούν μία ειδική κατηγορία δυαδικών δένδρων, αυτών των δυαδικών δένδρων αναζήτησης.

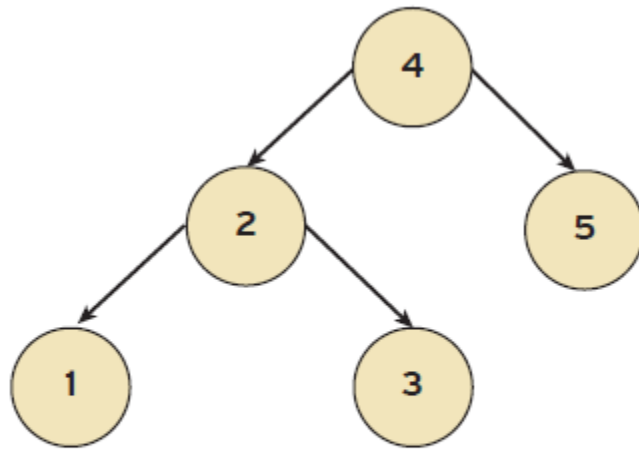
Η ιδέα πίσω από ένα δυαδικό δένδρο αναζήτησης είναι παρόμοια με αυτήν της δυαδικής αναζήτησης σε έναν ταξινομημένο πίνακα.

Ένα δυαδικό δένδρο αναζήτησης (binary search tree) είναι ένα δυαδικό δένδρο, όπου για κάθε κόμβο u , όλοι οι κόμβοι του αριστερού υποδένδρου έχουν τιμές μικρότερες της τιμής του κόμβου u και όλοι οι κόμβοι του δεξιού υποδένδρου έχουν τιμές μεγαλύτερες (ή ίσες) της τιμής του κόμβου u .

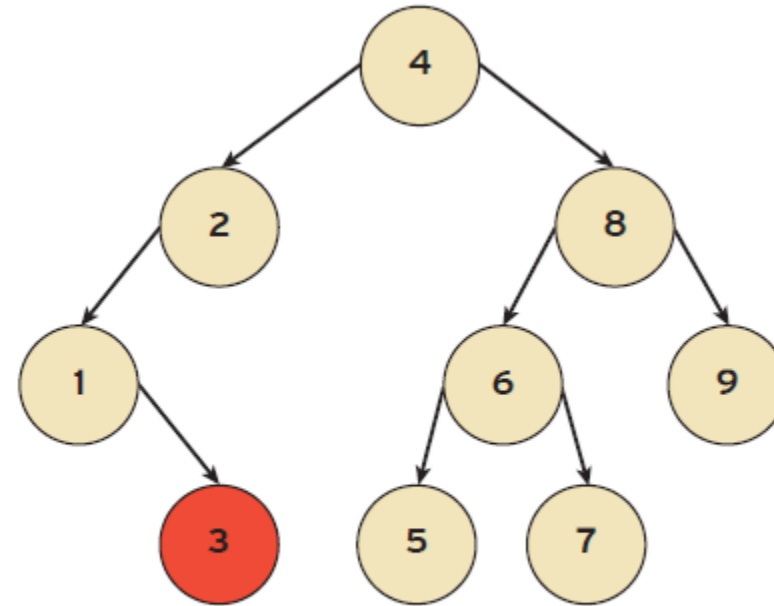
Για λόγους απλούστευσης θεωρούμε ότι δεν υπάρχουν τιμές ίσες με την τιμή του κόμβου u .



Δυαδικά Δένδρα Αναζήτησης



Δυαδικό δένδρο αναζήτησης



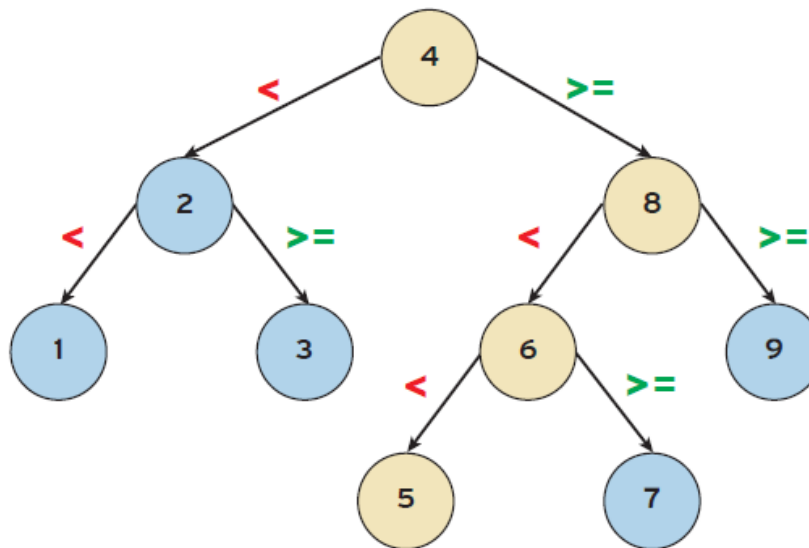
Μη δυαδικό δένδρο αναζήτησης

Αλγόριθμος αναζήτησης σε ένα δυαδικό δένδρο αναζήτησης

Η αναζήτηση για μια συγκεκριμένη τιμή γίνεται ταχύτερα χάρη στον τρόπο αποθήκευσης των τιμών.
πχ. Ψάχνουμε τον αριθμό 5.

1	8	2	9	3	6	7	4	5
---	---	---	---	---	---	---	---	---

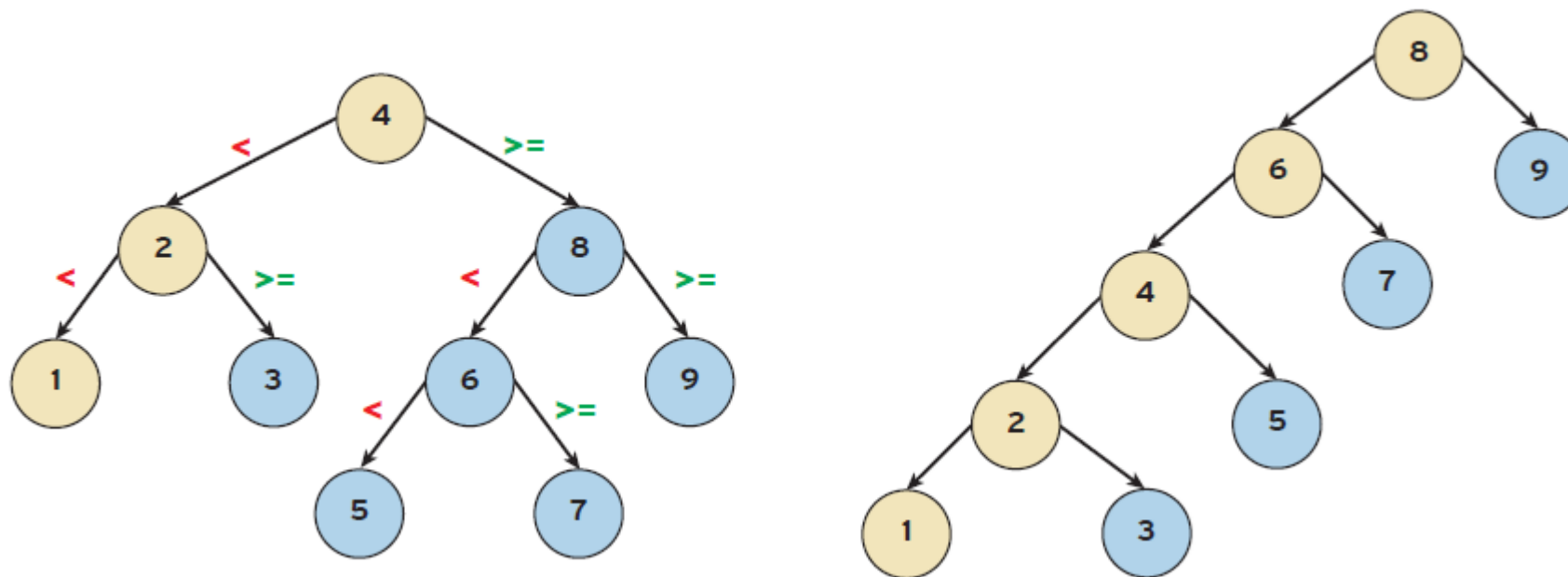
σε πίνακα:
9 επαναλήψεις



σε δυαδικό δένδρο
αναζήτησης:
4 επαναλήψεις

Αλγόριθμος αναζήτησης σε ένα δυαδικό δένδρο αναζήτησης

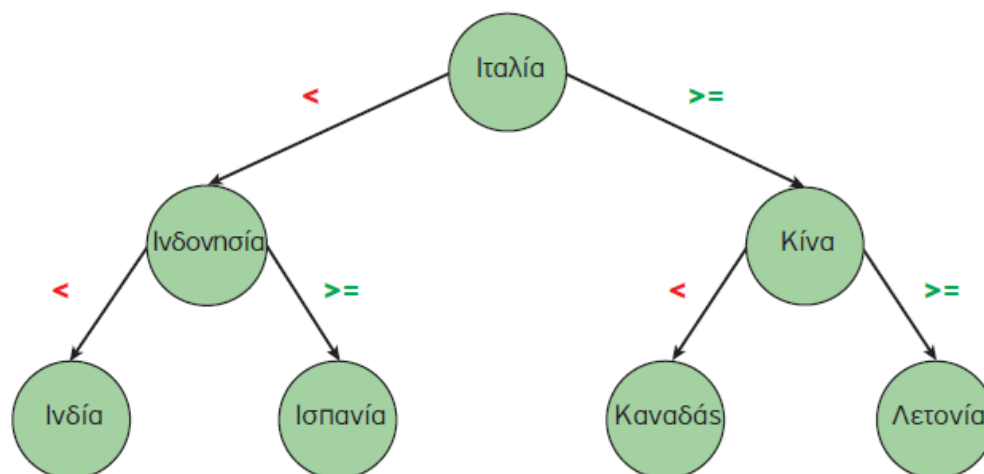
Αν θέλουμε να έχουμε γρήγορους αλγόριθμους αναζήτησης πρέπει να αποθηκεύουμε τις τιμές στα δυαδικά δένδρα αναζήτησης με έναν συγκεκριμένο τρόπο.



Αναζήτηση σε ταξινομημένο πίνακα και σε δυαδικό δένδρο αναζήτησης

Τα δυαδικά δένδρα αναζήτησης συνδυάζουν τα πλεονεκτήματα των λιστών, όσον αφορά τις πράξεις της εισαγωγής και της διαγραφής, αλλά και τα πλεονεκτήματα των ταξινομημένων πινάκων, όσον αφορά την πράξη της αναζήτησης.

Ινδία	Ινδονησία	Ισπανία	Ιταλία	Καναδάς	Κίνα	Λετονία
-------	-----------	---------	--------	---------	------	---------



1.3.3 ΓΡΑΦΟΙ

Τα δένδρα, γενικά, διέπονται από συγκεκριμένους κανόνες ενώ σε ορισμένους τύπους δένδρων ισχύουν ιδιαίτεροι κανόνες, όπως στα δυαδικά δένδρα αναζήτησης, στα οποία οι κόμβοι μπορεί να έχουν μόνο δύο συνδέσεις με δύο κόμβους ανά πάσα στιγμή.

Αν αγνοήσουμε αυτούς τους κανόνες τότε, δεν αναφερόμαστε σε δένδρα αλλά σε μία νέα δυναμική δομή δεδομένων, που ονομάζεται **γράφος**.

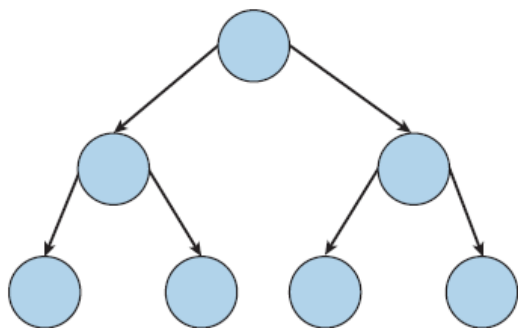
Τα δένδρα είναι στην πραγματικότητα ένα υποσύνολο των γράφων, περιορισμένοι τύποι γράφων. **Ένα δένδρο θα είναι πάντα ένα γράφος, αλλά δεν είναι όλοι οι γράφοι δένδρα.**

Ένας γράφος (graph) είναι μία δομή που αποτελείται από ένα σύνολο κόμβων (ή σημείων ή κορυφών) και ένα σύνολο γραμμών (ή ακμών ή τόξων) που ενώνουν μερικούς ή όλους τους κόμβους. Ο γράφος αποτελεί την πιο γενική δομή δεδομένων, με την έννοια ότι όλες οι προηγούμενες δομές που παρουσιάστηκαν μπορούν να θεωρηθούν περιπτώσεις γράφων.

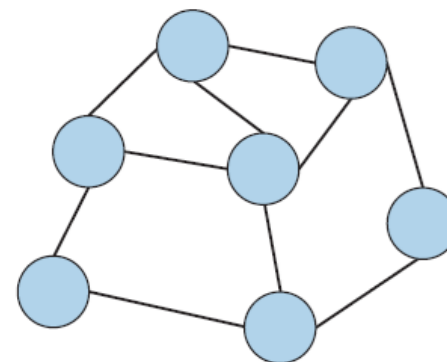
Δέντρο VS Γράφος

Ένα **δένδρο** μπορεί μόνο να ρέει προς **μία κατεύθυνση**, από τον κόμβο ρίζας σε κόμβους φύλλων ή κόμβους παιδιών. Ένα δένδρο μπορεί να έχει μόνο μονόδρομες συνδέσεις - ένας κόμβος παιδιού μπορεί να έχει μόνο έναν γονέα και ένα δένδρο δεν μπορεί να έχει βρόχους ή κυκλικούς δεσμούς.

Με τους γράφους, όλοι αυτοί οι περιορισμοί δεν υπάρχουν. Οι γράφοι δεν έχουν την έννοια ενός κόμβου «ρίζας». Οι κόμβοι μπορούν να συνδεθούν με οποιονδήποτε τρόπο. Για παράδειγμα, ένας κόμβος μπορεί να συνδεθεί με άλλους πέντε. Οι **γράφοι**, επίσης, δεν έχουν «μονοκατευθυντική» ροή - αντ' αυτού, μπορεί να **έχουν κατεύθυνση ή να μην έχουν καμιά κατεύθυνση**.



Δένδρο

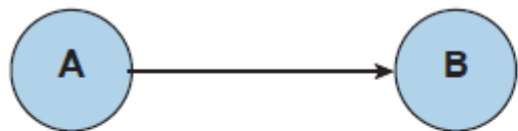


Γράφος

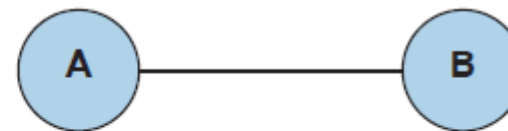
Τύποι Γράφων

Εάν όλες οι ακμές σε έναν γράφο έχουν κατεύθυνση, ο γράφος ονομάζεται **κατευθυνόμενος γράφος** (directed graph).

Εάν όλες οι ακμές σε έναν γράφο δεν έχουν κατεύθυνση, ο γράφος ονομάζεται **μη κατευθυνόμενος γράφος** (undirected graph).



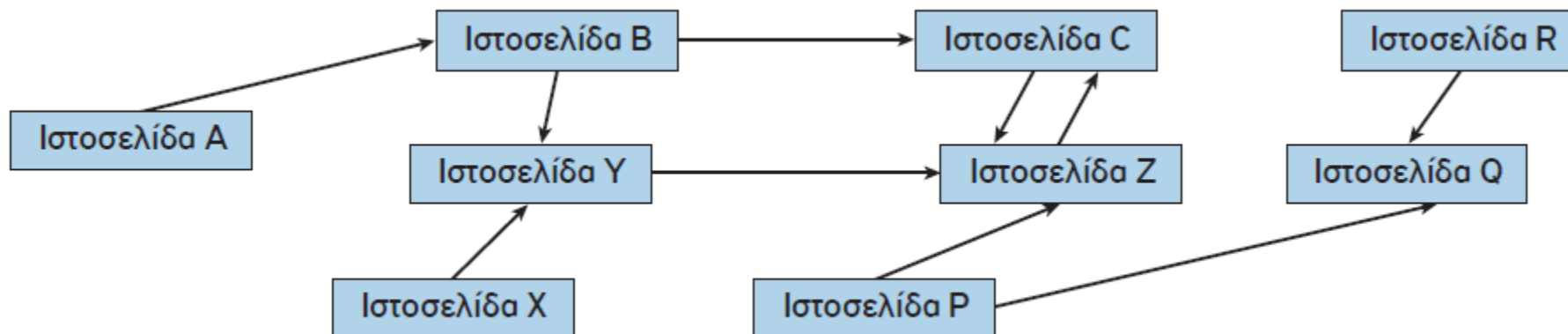
Κατευθυνόμενη ακμή



μη κατευθυνόμενη ακμή

Παράδειγμα- Παγκόσμιος ιστός

Ο Παγκόσμιος Ιστός (WWW) είναι ένας τεράστιος γράφος! Όταν κάνουμε κλικ ανάμεσα σε ιστότοπους και πλοηγούμαστε μεταξύ των διευθύνσεων URL, κάνουμε πραγματικά περιήγηση σε έναν γράφο. Μερικές φορές αυτές οι αναπαραστάσεις γράφων έχουν κόμβους με ακμές που είναι μη κατευθυνόμενες - μπορούμε να προχωρήσουμε από μια ιστοσελίδα σε μια άλλη και το αντίστροφο - και άλλες ακμές που κατευθύνονται - μπορούμε να πάμε μόνο από την ιστοσελίδα A στην ιστοσελίδα B, και ποτέ το αντίστροφο.

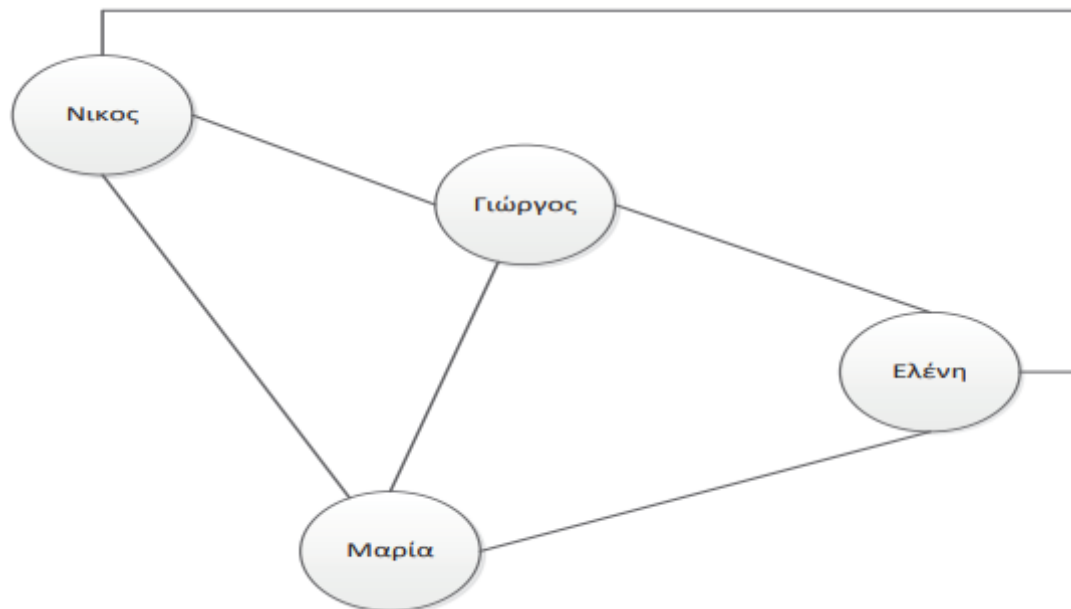


Παράδειγμα- Facebook

Το Facebook, ένα τεράστιο κοινωνικό δίκτυο, είναι ένας τύπος γράφου.

Στο Facebook, εάν σε προσθέσω σαν φίλο, πρέπει να αποδεχτείς το αίτημά μου. Δεν είναι δυνατόν να είμαι φίλος σου στο δίκτυο χωρίς να είσαι και δικός μου. Η σχέση μεταξύ δύο χρηστών (κόμβοι), είναι αμφίδρομη. Δεν υπάρχει η έννοια κόμβου «προέλευσης» και «προορισμού» - αντ' αυτού, είσαι φίλος μου και είμαι δικός σου.

Η σχέση μεταξύ των κόμβων είναι αμφίδρομη, επομένως αναπαρίσταται με **μη κατευθυνόμενες ακμές- μη κατευθυνόμενος γράφος**.



Παράδειγμα- Twitter

Στο Twitter μπορώ να σε ακολουθήσω, αλλά δεν απαιτείται να με ακολουθήσεις. Θα μπορούσαμε να αναπαραστήσουμε το Twitter ως κατευθυνόμενο γράφο. Κάθε ακμή που δημιουργούμε αντιπροσωπεύει μια μονόδρομη σχέση. Όταν με ακολουθείς στο Twitter, δημιουργείται μια ακμή στον γράφο με τον λογαριασμό σου ως κόμβο προέλευσης και τον λογαριασμό μου ως τον κόμβο προορισμού.



Εάν αποφασίσω να σε ακολουθήσω και εγώ στο Twitter, δημιουργώ μια δεύτερη ακμή, με τον λογαριασμό μου σαν κόμβο προέλευσης και τον δικό σου σαν κόμβο προορισμού. Η ακμή ΔΕ γίνεται αμφίδρομη και επομένως μη κατευθυνόμενη διότι θα μπορούσα να σε αγνοήσω σε οποιαδήποτε χρονική στιγμή.

