
Γ Λυκείου

Ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον

ΆΛΛΕΣ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

Περιεχόμενα

Κεφάλαιο 1 (Βιβλίο II): Δομές Δεδομένων και Αλγόριθμοι

- 1.3 Άλλες Δομές Δεδομένων
 - 1.3.1 Λίστες
 - 1.3.2 Δένδρα
 - 1.3.3 Γράφοι

Κατηφόρης Παναγιώτης
2019-2020

1.3 Άλλες Δομές Δεδομένων

Κοινό γνώρισμα των δομών της **στοίβας** και της **ουράς**, που υλοποιήθηκαν με χρήση μονοδιάστατου πίνακα, είναι ότι οι διαδοχικοί κόμβοι αποθηκεύονται σε **συνεχόμενες θέσεις της κύριας μνήμης**.

Στην συνέχεια παρουσιάζεται η περίπτωση τριών σημαντικών δομών δεδομένων, στις οποίες **οι κόμβοι δεν είναι απαραίτητο να κατέχουν συνεχόμενες θέσεις μνήμης**.

Οι δομές αυτές ανήκουν στην κατηγορία των **δυναμικών δομών** δεδομένων και πρόκειται για τις **λίστες**, τα **δένδρα** και τους **γράφους**.

Οι δομές δεδομένων που χρησιμοποιούν δείκτες αποκαλούνται δυναμικές (dynamic), γιατί η υλοποίησή τους γίνεται έτσι, ώστε να μην απαιτείται εκ των προτέρων καθορισμός του μέγιστου αριθμού κόμβων. Είναι φανερό ότι οι δομές αυτές είναι πιο ευέλικτες από τη στατική δομή του πίνακα, επειδή επεκτείνονται και συρρικνώνονται κατά τη διάρκεια εκτέλεσης του προγράμματος.

1.3.1 Λίστες

Περιγραφή της λίστας

Η έννοια της λίστας συναντάται αρκετά συχνά στην καθημερινότητά μας. Πόσες φορές δεν έχουμε καταρτίσει λίστες με τα αγαπημένα μας τραγούδια, τις επαφές μας, τα ψώνια που θέλουμε να κάνουμε ή ακόμα και λίστες με επιθυμίες και όνειρα; 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.

(απλά) Συνδεδεμένη λίστα

Ορισμός

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 κόμβος μιας λίστας έχει την μορφή:

Δεδομένα	Δείκτης
----------	---------

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.

Οι κόμβοι μιας (απλά) συνδεδεμένης λίστας είναι διατεταγμένοι σε μια συγκεκριμένη σειρά, χωρίς αυτό να σημαίνει ότι αποθηκεύονται σε συνεχόμενες θέσεις στη μνήμη.

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)**.

Τα δεδομένα κάθε κόμβου μιας λίστας :

- Είναι η είναι η αποθηκευμένη πληροφορία
- Πρέπει να είναι ίδιου τύπου

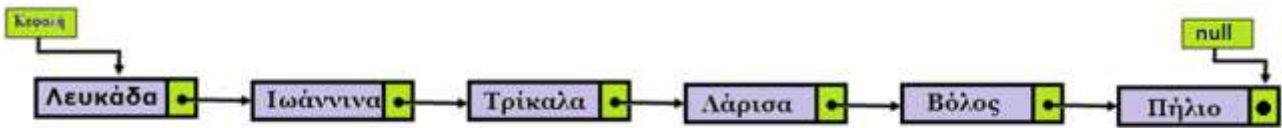
Ο δείκτης κάθε κόμβου μιας λίστας :

- Είναι ένας ιδιαίτερος τύπος που προσφέρεται από τις περισσότερες σύγχρονες γλώσσες προγραμματισμού
- Παίρνει τιμές διευθύνσεις της κύριας μνήμης (RAM)
- Χρησιμεύει για την σύνδεση των κόμβων που είναι αποθηκευμένοι (συνήθως) σε μη συνεχόμενες θέσεις μνήμης. Δηλαδή ο δείκτης ενός κόμβου έχει τιμή την διεύθυνση της RAM που βρίσκεται ο επόμενος κόμβος.
- Είναι πεδίο μέσα στον κόμβο της δομής δεδομένων.

Παράδειγμα μιας λίστα με 6 κόμβους

Η λίστα έχει τιμές τις πόλεις που θα επισκεφτούμε, σχεδιάζοντας ένα ταξίδι, από την Λευκάδα στο Πήλιο.

Ο δείκτης έχει σχήμα βέλους και δείχνει το κόμβο στον οποίο παραπέμπει



Η **κεφαλή (head)** είναι μια μεταβλητή που περιέχει την διεύθυνση μνήμης του 1^{ου} κόμβου της λίστας μας.

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**

Πρόσβαση (προσπέλαση) στους κόμβους μιας λίστας

Οι κόμβοι μιας λίστας δεν έχουν ονόματα. Γνωρίζουμε μόνο τις διευθύνσεις τους, που είναι αποθηκευμένες στους προηγούμενους κόμβους και αυτές θα αξιοποιήσουμε για να τους προσπελάσουμε.

Επομένως, αν θελήσουμε να έχουμε πρόσβαση στον τέταρτο κόμβο μιας λίστας, για να επεξεργαστούμε τα δεδομένα που περιέχει, θα πρέπει να ξεκινήσουμε από τον πρώτο κόμβο της λίστας, η διεύθυνση του οποίου περιέχεται στον δείκτη **Κεφαλή**.

Ξεκινώντας από τον πρώτο κόμβο της λίστας μπορούμε να έχουμε πρόσβαση στη διεύθυνση μνήμης του δεύτερου κόμβου. Από τον δεύτερο κόμβο μπορούμε να έχουμε πρόσβαση στη διεύθυνση μνήμης του τρίτου κόμβου. Και συνεχίζουμε έτσι μέχρι να φτάσουμε στον τέταρτο κόμβο. Αυτός είναι ο μόνος τρόπος για να διασχίσουμε μία συνδεδεμένη λίστα.

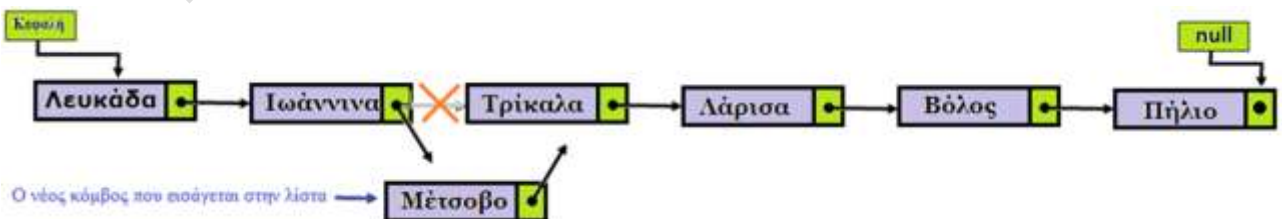
Εισαγωγή δεδομένων (νέου κόμβου) στη λίστα

Η εισαγωγή δεδομένων διευκολύνεται με την χρήση δεικτών.

Έστω ότι θέλουμε, μετά τα Ιωάννινα και πριν τα Τρίκαλα, να επισκεφτούμε το Μέτσοβο. Θα πρέπει λοιπόν, να εισάγουμε ένα νέο κόμβο ανάμεσα από τον 2^ο (Ιωάννινα) και τον 3^ο (Τρίκαλα).

Τι κάνουμε; Απλά:

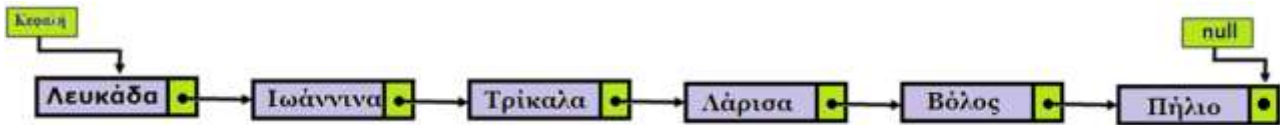
- Βάζουμε τον δείκτη του 2^{ου} κόμβου να δείχνει τον νεοεισαχθέντα κόμβο (και όχι τον 3^ο)
- Ο δείκτης του νεοεισαχθέντα κόμβου θα δείχνει τον παλιό 3^ο και τώρα 4^ο κόμβο



Διαγραφή κόμβου στη λίστα

Η διαγραφή δεδομένων διευκολύνεται με την χρήση δεικτών.

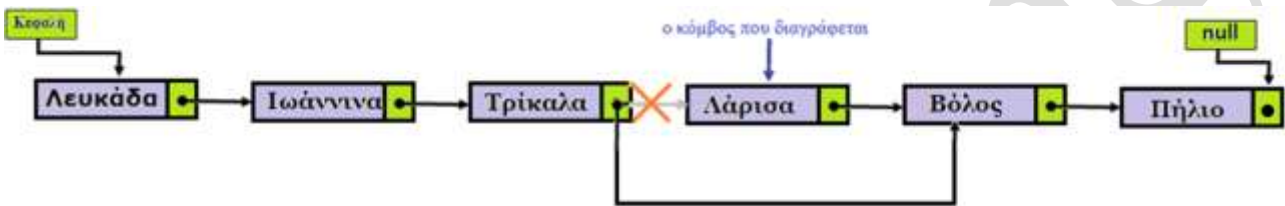
Το ταξίδι μας απεικονίζεται στη παρακάτω λίστα:



Ας πούμε ότι δεν θέλουμε να σταματήσουμε στην Λάρισα. Θα πρέπει λοιπόν να διαγράψουμε από την λίστα τον 4^ο κόμβο.

Πως θα γίνει αυτό; Απλά:

Αλλάζουμε την τιμή τον δείκτη του 3^{ου} κόμβου ώστε να δείχνει τον 5^ο κόμβο (και όχι τον 4^ο)



Ο παλιός 4^{ος} κόμβος (αυτός που διαγράφηκε) αποτελεί «άχρηστο δεδομένο» και ο χώρος μνήμης που καταλάμβανε, παραχωρείται για άλλη χρήση.

Διπλά συνδεδεμένη λίστα (doubly linked list)

Μια λίστα μπορεί να είναι απλά συνδεδεμένη, όπως παραπάνω, δηλαδή να μπορούμε να κινηθούμε **προς μία μόνο κατεύθυνση**, ξεκινώντας από τον πρώτο κόμβο και μετακινούμενοι προς τον τελευταίο, όπως δείχνουν τα βέλη του σχήματος παραπάνω ή να είναι **διπλά συνδεδεμένη (doubly linked list)**, δηλαδή, να μπορούμε να **τη διατρέξουμε και προς τις δύο κατευθύνσεις**. Η χρήση του δεύτερου δείκτη (**ουρά**) προσφέρει τη δυνατότητα ξεκινώντας από οποιοδήποτε κόμβο της λίστας να μπορούμε να διαβάσουμε τη λίστα και προς τις δυο κατευθύνσεις.

Παράδειγμα διπλά συνδεδεμένης λίστας με 4 κόμβους.

Η λίστα έχει τις στάσεις του λεωφορείου που κάνει την διαδρομή Λευκάδα - Βασιλική και αντίστροφα.



Παρατηρούμε 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

1. **Εισαγωγή** κόμβου στη λίστα (εισαγωγή κόμβου στην αρχή, στο τέλος της λίστας ή ενδιάμεσα).
2. **Διαγραφή** κόμβου από τη λίστα (διαγραφή από την αρχή, το τέλος της λίστας ή ενδιάμεσα).
3. **Έλεγχος** για το αν η λίστα είναι κενή.
4. **Αναζήτηση** κόμβου για την εύρεση συγκεκριμένου στοιχείου.
5. **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

1. 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.
2. 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.
3. Οι κόμβοι της **λίστας** αποθηκεύονται σε **μη συνεχόμενες θέσεις μνήμης** σε αντίθεση με τους **πίνακες**, όπου τα στοιχεία αποθηκεύονται σε **συνεχόμενες θέσεις μνήμης**.

Πλεονεκτήματα των λιστών έναντι των πινάκων

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

Μειονεκτήματα των λιστών έναντι των πινάκων

1. **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.
2. 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.

Περιγραφή του δένδρου

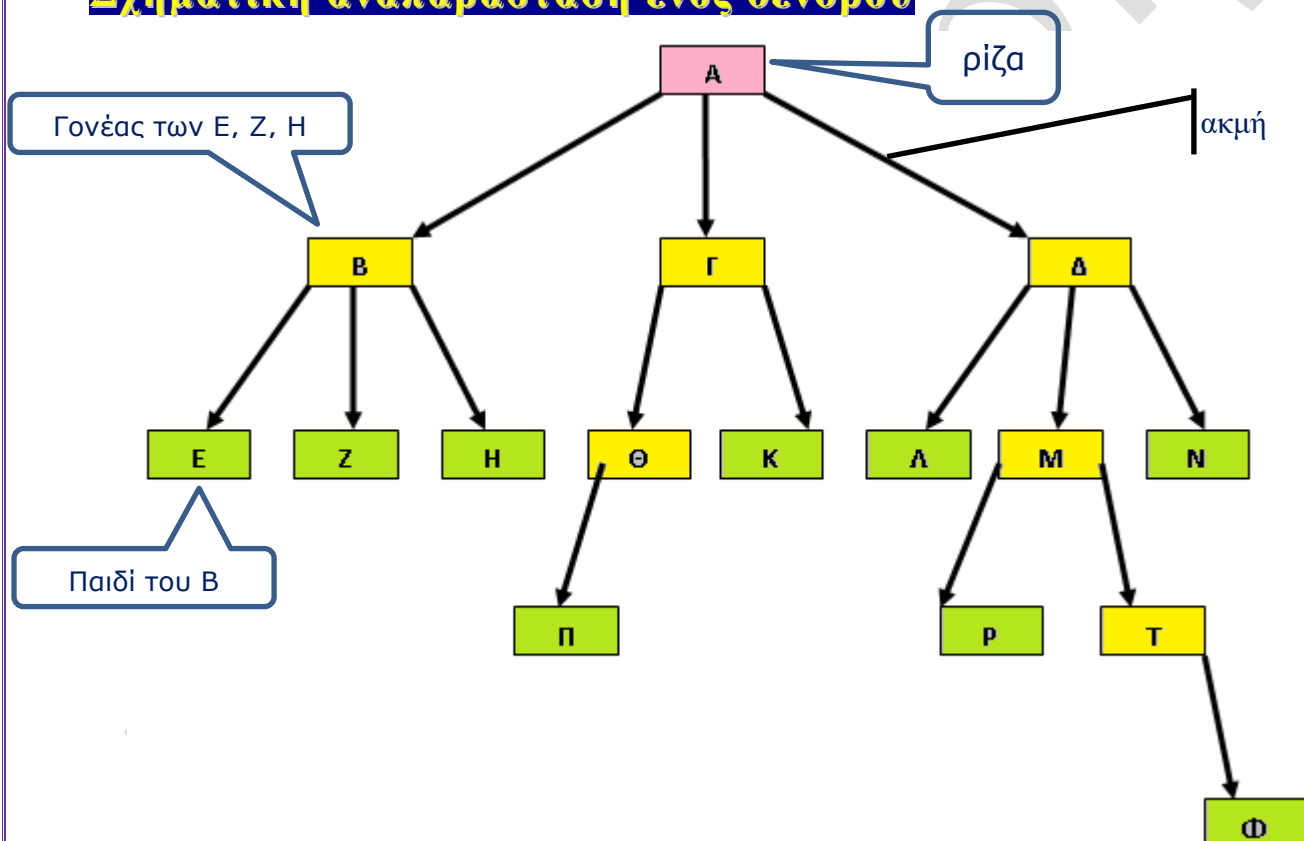
Η έννοια του δέντρου είναι στενά συνδεδεμένη με την έννοια της **ιεραρχίας** και αποτελεί μια **μη γραμμική δομή**

Είναι δομές δεδομένων που υλοποιούνται με την βοήθεια δεικτών (δυναμικές δομές όπως και οι λίστες) αλλά μπορούν να υλοποιηθούν και με στατικές δομές (όπως οι πίνακες).

Το Δέντρο είναι ένα πεπερασμένο σύνολο κόμβων (ίδιου τύπου) και ακμών που συνδέουν τους κόμβους, με βάση κάποια σχέση που δημιουργεί την ιεραρχική δομή των κόμβων.

Το κύριο χαρακτηριστικό του δένδρου είναι, ότι από ένα κόμβο δεν υπάρχει ένας μόνο επόμενος, αλλά περισσότεροι. Υπάρχει ένας μόνο κόμβος που λέγεται **ρίζα** από τον οποίο ξεκινούν όλοι οι άλλοι κόμβοι.

Σχηματική αναπαράσταση ενός δένδρου



Ρίζα: (Κόμβος χωρίς πρόγονο) Ο κόμβος A λέγεται ρίζα του δένδρου.

Φύλλα: (Κόμβος χωρίς παιδιά) Όλοι οι κόμβοι από τους οποίους δεν αρχίζει κάποιο υποδέντρο (δεν έχουν παιδιά),

Οι κόμβοι E, Z, H, Π, Κ, Λ, Ρ, Φ, Ν είναι τα φύλλα του παραπάνω δένδρου.

Γονέας ενός κόμβου λέγεται ο κόμβος στον οποίο ανήκει
π.χ. Ο κόμβος A είναι γονέας των B, Γ, Δ.

Παιδιά ενός κόμβου λέγονται οι κόμβοι που περιέχει
π.χ. Οι κόμβοι B, Γ, Δ λέγονται **παιδιά** της ρίζας. Ομοίως

Οι κόμβοι E, Z, H λέγονται παιδιά του κόμβου B

Αδέλφια: . Κόμβοι με τον ίδιο γονέα

π.χ.Οι κόμβοι E, Z, H είναι αδέλφια γιατί έχουν τον ίδιο πατέρα τον κόμβο B.

Υποδένδρο: τμήμα του δένδρου με ρίζα ένα οποιονδήποτε κόμβο

Ορισμός του δένδρου

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 E, εκτός από τη ρίζα, υπάρχει μόνο μια ακμή που καταλήγει στον κόμβο αυτόν ξεκινώντας από κάποιον άλλον κόμβο B. Ο κόμβος B ονομάζεται γονέας του E και ο κόμβος E παιδί του B.

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.

Χρήση των δένδρων στην πληροφορική

Υπάρχουν δύο λόγοι για τους οποίους τα δένδρα είναι μια τόσο «ισχυρή» δομή.

- 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, εύκολα αναγνώσιμες (από τον άνθρωπο) και εύκολα επεξεργάσιμες από τον υπολογιστή.

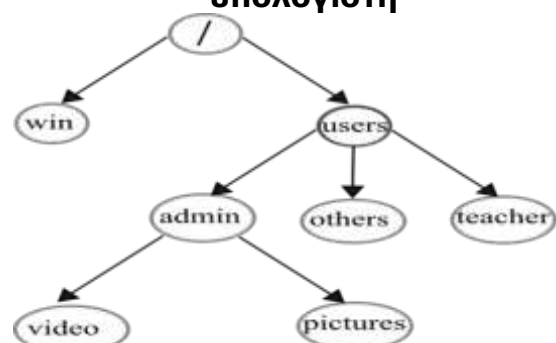
Συνήθως, όταν αναφερόμαστε στα δένδρα, μας έρχονται στο νου διάφορες αναπαραστάσεις δεδομένων του πραγματικού κόσμου αλλά και της Πληροφορικής που διέπονται από ένα είδος **φυσικής ιεραρχίας**.

Παραδείγματα δένδρων

Ένα γενεαλογικό Δένδρο



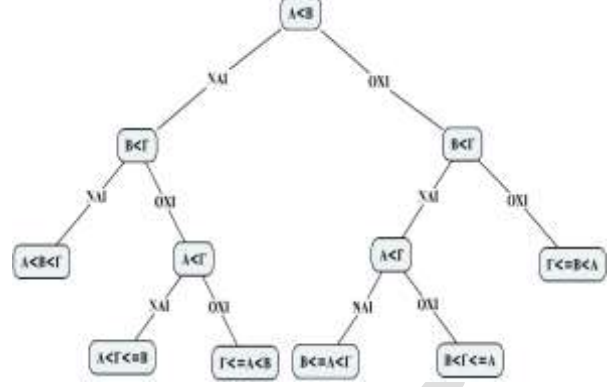
Το σύστημα των αρχείων του υπολογιστή



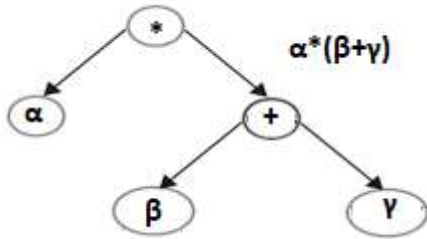
Οργανόγραμμα μιας εταιρείας



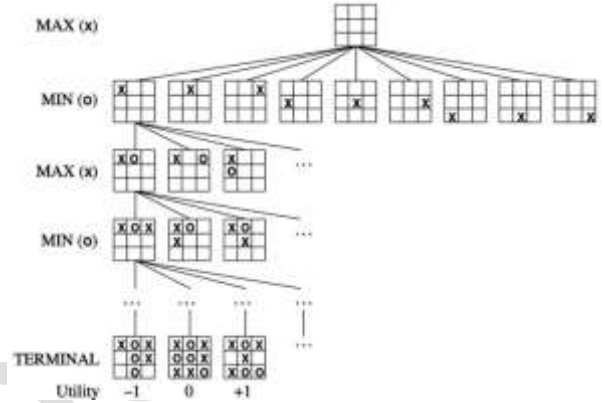
Δένδρο απόφασης



Προτεραιότητα πράξεων



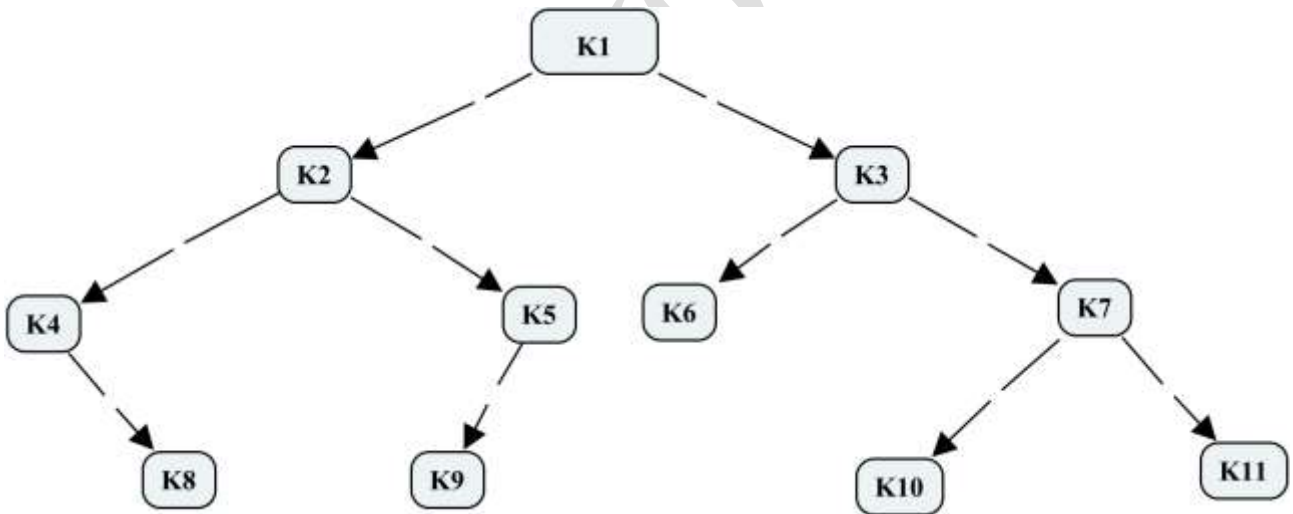
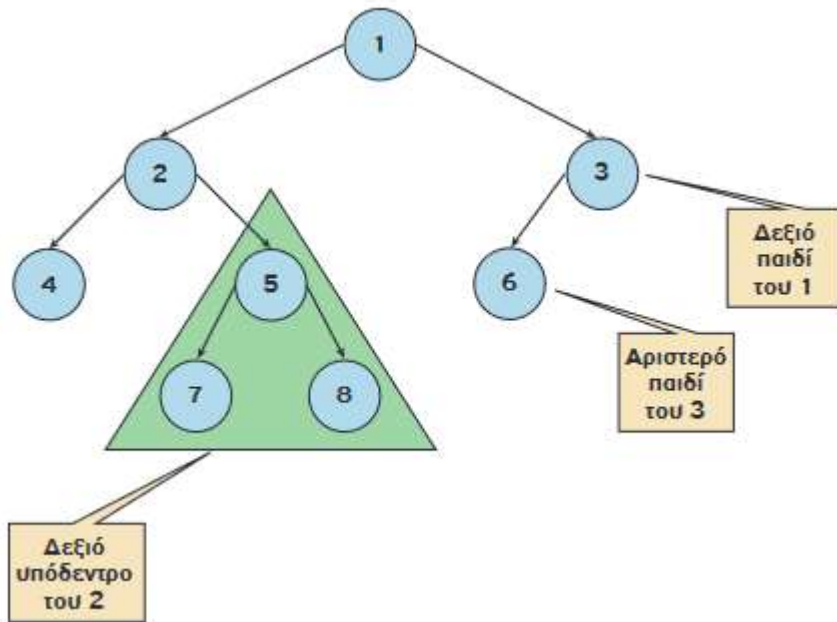
Δένδρο παιχνιδιού (τρίλιζα)



Καθηφ

Δυαδικό δένδρο

Υπάρχουν πολλά είδη δέντρων. Ένα από τα πιο γνωστά είναι το ονομαζόμενο **δυαδικό δέντρο** (Binary tree), όπου είναι διατεταγμένο δένδρο, στο οποίο κάθε κόμβος έχει το πολύ δύο παιδιά, το **αριστερό** και το **δεξιό** παιδί.



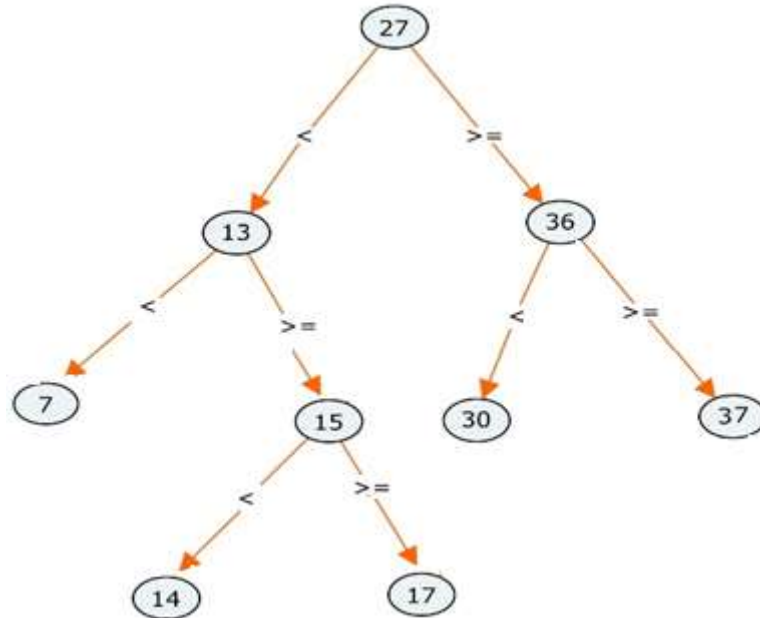
Ένα **ισορροπημένο** δυαδικό δέντρο είναι ένα δυαδικό δέντρο στην οποία το αριστερό και το δεξιό υπόδεντρο κάθε κόμβου διαφέρουν σε ύψος όχι περισσότερο από 1. Μπορεί κανείς επίσης να θεωρήσει ως τέτοια δυαδικά δέντρα στα οποία κανένα φύλλο δεν απέχει πολύ περισσότερο από τη ρίζα, από οποιοδήποτε άλλο φύλλο.

Ισορροπημένο	Μη ισορροπημένο

Δυαδικό δένδρο αναζήτησης

Ένα δυαδικό δένδρο αναζήτησης (binary search tree) είναι ένα δυαδικό δέντρο όπου:

- Οι τιμές στις κορυφές του αριστερού υποδέντρου είναι μικρότερες της ρίζας ΕΝΩ οι τιμές στις κορυφές του δεξιού υποδέντρου είναι μεγαλύτερες ή ίσες της ρίζας (Αριστερό παιδί < Πατέρα ≤ Δεξί παιδί).
- Η ιδιότητα αυτή ισχύει για κάθε υποδέντρο του δυαδικού δέντρου.



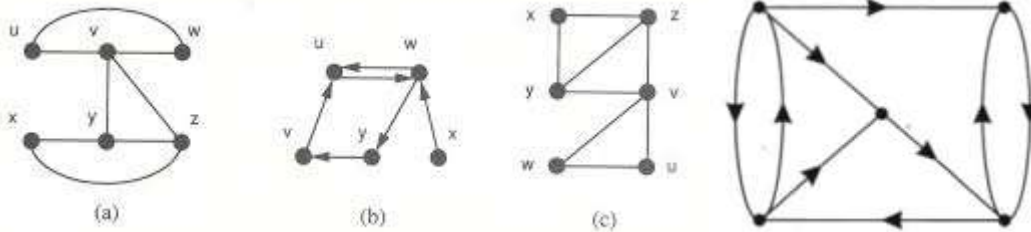
Τα δυαδικά δένδρα αναζήτησης συνδυάζουν τα **πλεονεκτήματα των λιστών**, όσον αφορά τις πράξεις της εισαγωγής και της διαγραφής, αλλά και τα **πλεονεκτήματα των ταξινομημένων πινάκων**, όσον αφορά την πράξη της αναζήτησης.

Το βασικό, λοιπόν, πλεονέκτημα των δυαδικών δέντρων αναζήτησης βρίσκεται στο ίδιο το όνομα και συγκεκριμένα στη λέξη «αναζήτηση». Η αναζήτηση για μια συγκεκριμένη τιμή γίνεται ταχύτερα χάρη στον τρόπο αποθήκευσης των τιμών.

Τι είναι ο γράφος

Ένας γράφος αποτελείται από ένα σύνολο **κόμβων** (που λέγονται **σημεία** ή **κορυφές**) και ένα σύνολο **γραμμών** (που λέγονται **ακμές** ή **τόξα**) που ενώνουν μερικούς ή όλους τους κόμβους. Οι κόμβοι αυτοί ενώνονται τυχαία μεταξύ τους (Δεν υπάρχει κάποια ιεραρχική οργάνωση). Οι κόμβοι δεν είναι απαραίτητο να κατέχουν συνεχόμενες θέσεις μνήμης.

Ο γράφος αποτελεί την πιο γενική δομή δεδομένων, με την έννοια ότι όλες οι προηγούμενες δομές δεδομένων μπορούν να θεωρηθούν περιπτώσεις γράφων.



Τύποι γράφων



Κατευθυνόμενη ακμή



Μη κατευθυνόμενη ακμή

κατευθυνόμενος γράφος

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

μη κατευθυνόμενος γράφος

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

Παραδείγματα

Οι γράφοι προσφέρουν μια χρήσιμη μέθοδο για τη διατύπωση και λύση πολλών προβλημάτων, όπως σε:

- δίκτυα και συστήματα τηλεπικοινωνιών (π.χ. το Internet),
- χάρτες
- επιλογή δρομολογίων,
- προγραμματισμό εργασιών,
- ανάλυση προγραμμάτων.

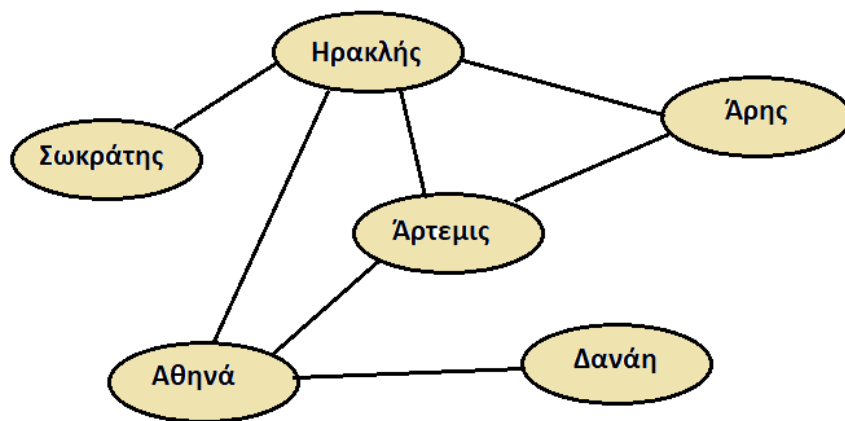
Σχηματικά στον γράφο οι κόμβοι (μαύροι κύκλοι) μπορούν να ενώνονται με πολλούς άλλους κόμβους (με τα τόξα).

Εφαρμογές

1. Μη κατευθυνόμενος γράφος (Facebook)

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, είναι **αμφίδρομη**. Η φιλία, λοιπόν, στο μπορεί να αναπαρασταθεί με έναν **μη-κατευθυνόμενο γράφο**.

Ας θεωρήσουμε έξι χρήστες του Facebook, τον Σωκράτη, την Αθηνά, τον Ηρακλή, την Άρτεμη, τον Άρη και την Δανάη. Οι χρήστες αυτοί θα είναι οι **κορυφές** του γράφου. Η **ύπαρξη ακμής** μεταξύ δύο κορυφών σηματοδοτεί την ύπαρξη φιλίας μεταξύ των αντίστοιχων χρηστών. Ο γράφος μας έχει την μορφή:



Προκειμένου να κωδικοποιήσουμε τον παραπάνω γράφο στην Γλώσσα με πίνακα γειτνίασης χρειαζόμαστε έναν δισδιάστατο τετραγωνικό πίνακα 6x6 ακέραίου τύπου. Οι διαστάσεις του πίνακα είναι και οι δύο ίσες με 6, όσο και το πλήθος των κορυφών του γράφου. Ο πίνακας θα έχει την μορφή:

		Σωκράτης	Ηρακλής	Άρης	Άρτεμις	Αθηνά	Δανάη
		1	2	3	4	5	6
Σωκράτης	1	0	1	0	0	0	0
Ηρακλής	2	1	0	1	1	1	0
Άρης	3	0	1	0	1	0	0
Άρτεμις	4	0	1	1	0	1	0
Αθηνά	5	0	1	0	1	0	1
Δανάη	6	0	0	0	0	1	0

Η τιμή 1 σε κάποια θέση του πίνακα σημαίνει ότι τα ονόματα που αντιστοιχούν στην αντίστοιχη γραμμή και στήλη του πίνακα είναι φίλοι, ενώ μια τιμή 0 σημαίνει ότι δεν είναι φίλοι.

Παρατηρήσεις.

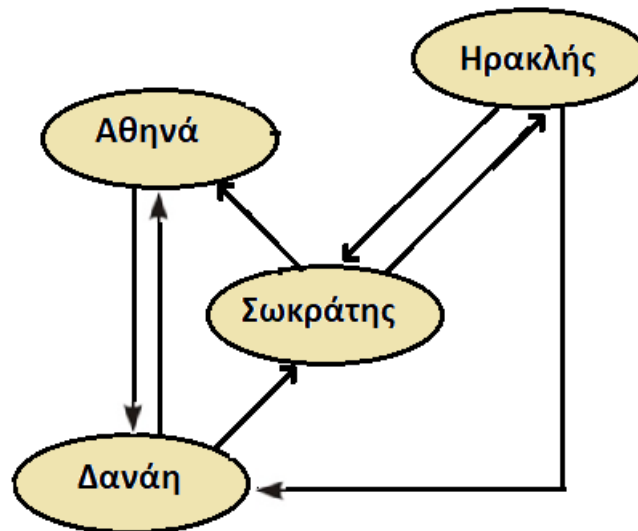
α) Ο πίνακας σχέσεων θα μπορούσε να είναι τύπου χαρακτήρα και το 1 να αντικατασταθεί με την τιμή φιλία, ενώ το 0 με το κενό. Επίσης θα μπορούσε να λογικός πίνακας οπότε το 1 θα ήταν ΑΛΗΘΗΣ και το 0 ΨΕΥΔΗΣ.

β) Παρατηρούμε ότι οι τιμές στις θέσεις $[i,j]$ και $[j,i]$ είναι ίδιες. Θα μπορούσαμε, λοιπόν, να αποτυπώσουμε την σχέση φιλίας σε ένα τριγωνικό πίνακα.

2. Μη κατευθυνόμενος γράφος (Twitter)

Το Twitter, λειτουργεί πολύ διαφορετικά από το Facebook. Μπορώ να σε ακολουθήσω, αλλά δεν απαιτείται να με ακολουθήσεις. Η σχέση, λοιπόν, μεταξύ δύο χρηστών του Twitter, είναι **μονόδρομη**, θα μπορούσε λοιπόν να αναπαρασταθεί με έναν κατευθυνόμενο γράφο.

Ας θεωρήσουμε τέσσερις χρήστες του Twitter, τον Σωκράτη, την Αθηνά, τον Ηρακλή, και την Δανάη. Οι χρήστες αυτοί θα είναι οι κορυφές του γράφου. Η ύπαρξη **ακμής - βέλους** μεταξύ δύο κορυφών, η αριστερή κορυφή ακολουθεί την δεξιά. Ο γράφος μας έχει την μορφή:



Η κωδικοποίηση τον παραπάνω γράφου με πίνακα γειτνίασης γίνεται με έναν δισδιάστατο τετραγωνικό πίνακα 4x4 ακέραιου τύπου. Ο πίνακας θα έχει την μορφή:

		Σωκράτης	Ηρακλής	Αθηνά	Δανάη
		1	2	3	4
Σωκράτης	1	0	1	1	0
Ηρακλής	2	1	0	0	1
Αθηνά	3	0	0	0	1
Δανάη	4	1	0	1	0

Η τιμή 1 σε κάποια θέση $[i,j]$ σημαίνει ότι ο Σωκράτης (1) ακολουθεί την Αθηνά (3). Παρατηρούμε ότι η τιμή του πίνακα στην θέση $[3,1]$ είναι 0, που σημαίνει ότι η Αθηνά (3) δεν ακολουθεί τον Σωκράτη (1).