

Δομές Δεδομένων

Τι ονομάζουμε δομή δεδομένων

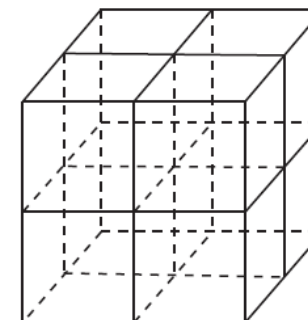
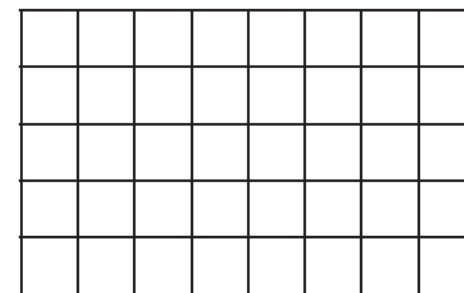
- Στην πληροφορική, η έννοια της **δομής δεδομένων** αναφέρεται στους διαφορετικούς δυνατούς τρόπους οργάνωσης και αποθήκευσης δεδομένων μέσα σε έναν υπολογιστή, ώστε τα δεδομένα αυτά να μπορούν να χρησιμοποιηθούν αποδοτικά.

Δομές Δεδομένων

- Πίνακες (στατική δομή δεδομένων)
- Στοίβα
- Ουρά
- Λίστες
- Δένδρα
- Γράφοι

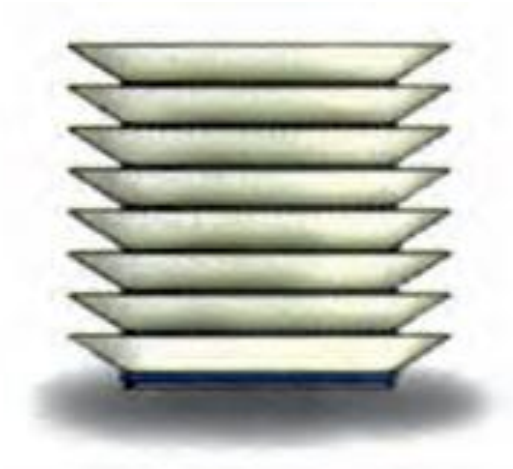
Πίνακες (στατική δομή δεδομένων)

- Ορίσουμε τον πίνακα ως μια δομή που περιέχει στοιχεία του ίδιου τύπου (δηλαδή ακέραιους, πραγματικούς κ.λπ).
- Γενικά η αναφορά στα στοιχεία ενός πίνακα γίνεται με τη χρήση του συμβολικού ονόματος του πίνακα ακολουθούμενου από την τιμή ενός ή περισσότερων δεικτών (indexes) σε παρένθεση ή αγκύλη.
- Ένας πίνακας μπορεί να είναι μονοδιάστατος, αλλά στη γενικότερη περίπτωση μπορεί να είναι δισδιάστατος, τρισδιάστατος και γενικά n -διάστατος πίνακας.



Στοίβα

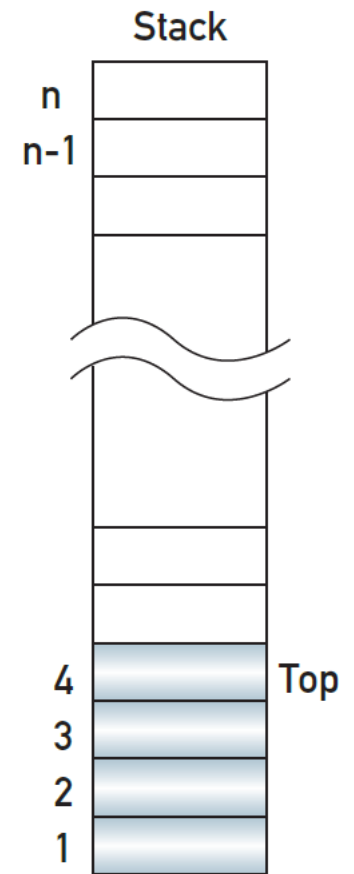
- **Στοίβα (stack)**, ονομάζεται μια δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο, ώστε τα στοιχεία που βρίσκονται στην κορυφή της στοίβας λαμβάνονται πρώτα, ενώ αυτά που βρίσκονται στο βάθος της στοίβας λαμβάνονται τελευταία.
- Η παραπάνω μέθοδος ονομάζεται **Τελευταίο Μέσα, Πρώτο Έξω ή LIFO** (=Last In First Out). Μπορούμε να φανταστούμε την τοποθέτηση των στοιχείων μιας στοίβας σε **κατακόρυφη** σειρά. Χαρακτηριστικό παράδειγμα είναι μια στοίβα από πιάτα. Παίρνουμε προς χρήση το πιάτο που τοποθετήσαμε τελευταίο.



Στοίβα – Κύριες Λειτουργίες

Οι **κύριες λειτουργίες** σε μια στοίβα είναι δύο:

1. Η **ώθηση** (push) στοιχείου στην κορυφή της στοίβας.
Στη διαδικασία της ώθησης ελέγχουμε αν η στοίβα είναι γεμάτη. Στην περίπτωση που προσπαθήσουμε να «προσθέσουμε» ένα στοιχείο σε μια ήδη γεμάτη στοίβα, έχουμε **υπερχείλιση** (overflow) της στοίβας.
2. Η **απώθηση** (pop) στοιχείου από τη στοίβα.
Στη διαδικασία της απώθησης ελέγχουμε αν υπάρχει ένα τουλάχιστον στοιχείο στη στοίβα. Στην περίπτωση που προσπαθήσουμε να «αφαιρέσουμε» ένα στοιχείο από μία κενή στοίβα, έχουμε **υποχείλιση** (underflow) της στοίβας.



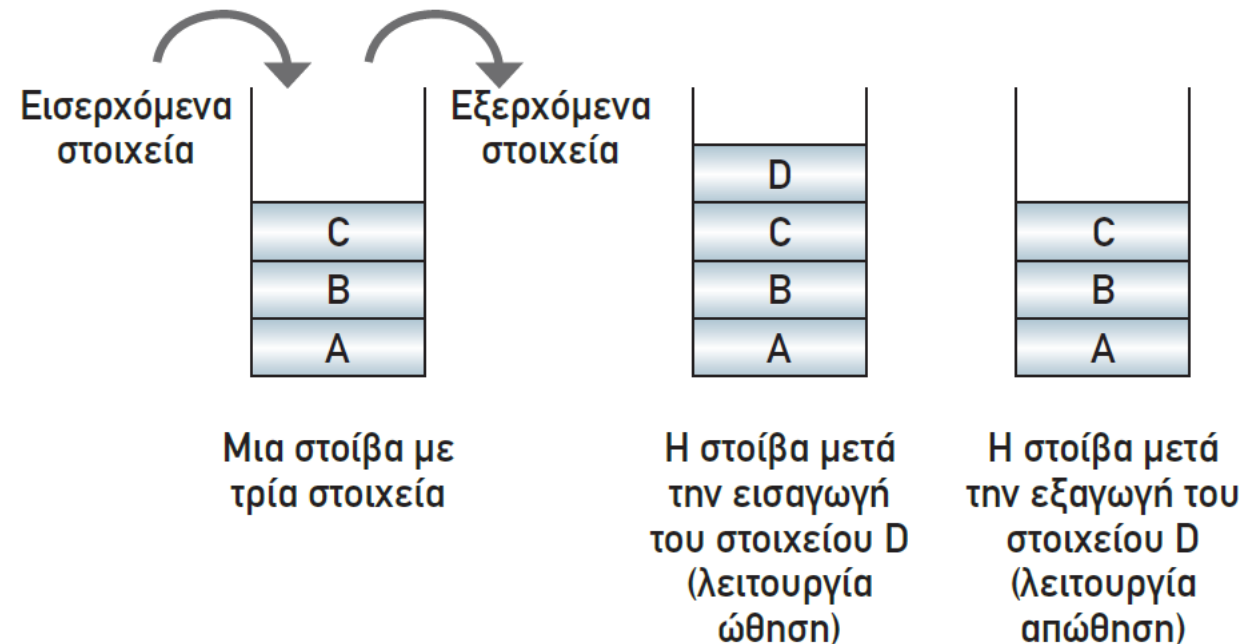
Σχ. 3.3 Υλοποίηση στοίβας με χρήση πίνακα

Στοίβα – υλοποίηση

Μια στοίβα μπορεί να υλοποιηθεί πολύ εύκολα με τη βοήθεια ενός μονοδιάστατου πίνακα.

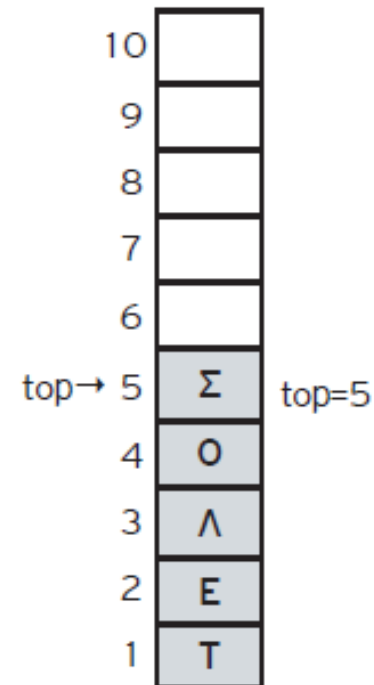
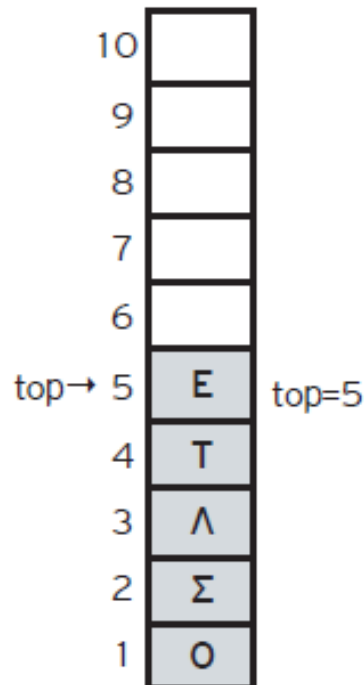
Μια βοηθητική μεταβλητή (με όνομα συνήθως **top**) χρησιμοποιείται για να δείχνει το στοιχείο που τοποθετήθηκε τελευταίο στην κορυφή της στοίβας. Για την εισαγωγή ενός νέου στοιχείου στη στοίβα (ώθηση) αρκεί να αυξηθεί η μεταβλητή **top** κατά ένα και στη θέση αυτή να εισέλθει το στοιχείο.

Αντίθετα για την εξαγωγή ενός στοιχείου από τη στοίβα (απώθηση) εξέρχεται πρώτα το στοιχείο που δείχνει η μεταβλητή **top** και στη συνέχεια η **top** μειώνεται κατά ένα για να δείχνει τη νέα κορυφή.



Στοίβα – παράδειγμα

Σε μια άδεια στοίβα 10 θέσεων ωθούνται τα στοιχεία Ο, Σ, Λ, Τ, Ε. Με ποιον τρόπο πρέπει να γίνει η ώθηση και η απώθηση των στοιχείων, ώστε να έχουμε ως έξοδο τα στοιχεία Τ, Ε, Λ, Ο, Σ, με το στοιχείο Σ να βρίσκεται στην κορυφή της στοίβας



Στοίβα - Ερωτήσεις

α/α	Προτάσεις	Σ	Λ
1	Για την υλοποίηση μιας στοίβας μπορεί να χρησιμοποιηθεί ένας πίνακας.	☐	☐
2	Στη στοίβα το στοιχείο που μπαίνει πρώτο βγαίνει πρώτο.	☐	☐
3	Στην υλοποίηση της στοίβας χρειάζονται δύο μεταβλητές-δείκτες για την υλοποίηση των δύο βασικών λειτουργιών που εκτελούνται σε αυτή.	☐	☐
4	Η λειτουργία της ώθησης μπορεί να εκτελεστεί και σε μια άδεια στοίβα.	☐	☐
5	Η λειτουργία της ώθησης μπορεί να εκτελεστεί και σε μια γεμάτη στοίβα.	☐	☐
6	Η ώθηση στοιχείου γίνεται στην κορυφή της στοίβας.	☐	☐
7	Στη δομή της στοίβας απαιτούνται δύο δείκτες, ο εμπρός και ο πίσω.	☐	☐
8	Υπερχείλιση έχουμε όταν εισάγουμε ένα στοιχείο σε μια ήδη γεμάτη στοίβα.	☐	☐
9	Η μέθοδος LIFO περιγράφει τη διαδικασία εκείνη κατά την οποία το στοιχείο που τοποθετείται τελευταίο εξάγεται πρώτο	☐	☐
10	Κάθε στοιχείο που εισάγεται πρώτο σε μια στοίβα είναι αυτό που εξάγεται πρώτο.	☐	☐

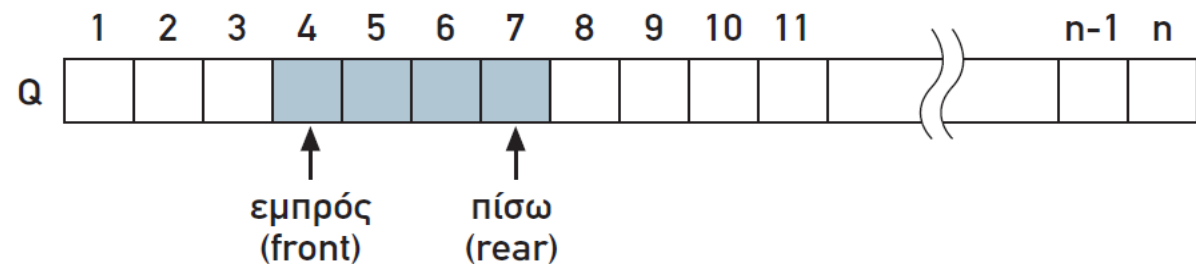
Ουρά

- **Ουρά (Queue)**, ονομάζεται μια δομή δεδομένων το σύνολο των στοιχείων της οποίας είναι διατεταγμένο με τέτοιο τρόπο, ώστε τα στοιχεία που τοποθετήθηκαν πρώτα στην ουρά να λαμβάνονται επίσης πρώτα.
- Η παραπάνω μέθοδος ονομάζεται **Πρώτο Μέσα, Πρώτο Έξω ή FIFO** (=First In First Out).
- Μπορούμε να φανταστούμε την τοποθέτηση των στοιχείων μιας ουράς σε **οριζόντια** σειρά.
Ο πελάτης μιας τράπεζας που μπαίνει πρώτος σε μια ουρά για να εξυπηρετηθεί, είναι αυτός που εξυπηρετείται και πρώτος, με την έξοδό του από την ουρά αναμονής.



Ουρά – Κύριες Λειτουργίες

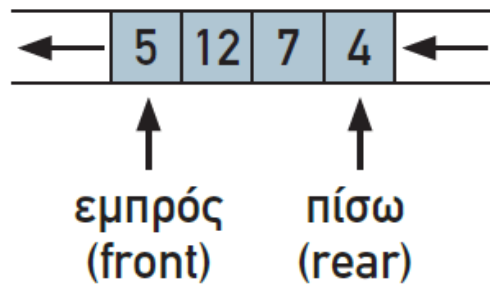
1. Η **εισαγωγή** (enqueue) στοιχείου στο πίσω άκρο της ουράς.
 2. Η **εξαγωγή** (dequeue) στοιχείου από το εμπρός άκρο της ουράς.
- Μια ουρά μπορεί να **υλοποιηθεί** με τη βοήθεια ενός μονοδιάστατου πίνακα.
 - Για την **εισαγωγή** ενός νέου στοιχείου στην ουρά αυξάνεται ο δείκτης **rear** κατά ένα και στη θέση αυτή αποθηκεύεται το στοιχείο.
 - Αντίστοιχα για τη λειτουργία της **εξαγωγής**, εξέρχεται το στοιχείο που δείχνει ο δείκτης **front**, ο οποίος στη συνέχεια αυξάνεται κατά ένα, για να δείχνει το επόμενο στοιχείο που πρόκειται να εξαχθεί.



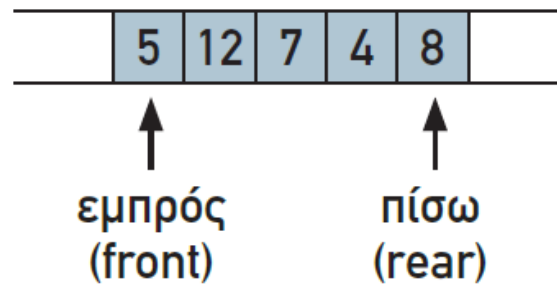
Σχ. 3.5 Υλοποίηση ουράς με χρήση πίνακα

Ουρά – Υλοποίηση

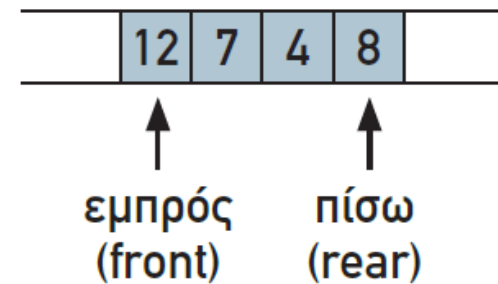
Σε αντίθεση με τη δομή της στοίβας, στην περίπτωση της ουράς απαιτούνται δύο δείκτες: ο **εμπρός (front)** και ο **πίσω (rear)** δείκτης, που μας δίνουν τη θέση του στοιχείου που σε πρώτη ευκαιρία θα εξαχθεί και τη θέση του στοιχείου που μόλις εισήλθε.



α) Μια ουρά
με 4 στοιχεία



β) Η ουρά μετά
την εισαγωγή του
στοιχείου 8



γ) Η ουρά μετά
την εξαγωγή του
στοιχείου 5

Σχ. 3.4. Εισαγωγή και εξαγωγή από ουρά

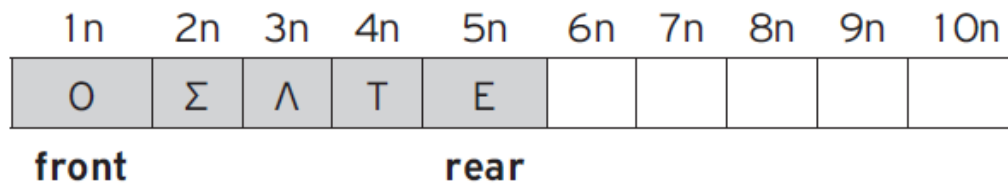
Ουρά – Υλοποίηση

Σε αντίθεση με τη δομή της στοίβας, στην περίπτωση της ουράς απαιτούνται δύο δείκτες: ο **εμπρός (front)** και ο **πίσω (rear)** δείκτης, που μας δίνουν τη θέση του στοιχείου που σε πρώτη ευκαιρία θα εξαχθεί και τη θέση του στοιχείου που μόλις εισήλθε.

ΠΑΡΑΔΕΙΓΜΑ

Σε μια άδεια ουρά 10 θέσεων εισάγονται τα στοιχεία Ο, Σ, Λ, Τ, Ε . Με ποιον τρόπο πρέπει να εισαχθούν και να εξαχθούν τα στοιχεία, ώστε η έξοδος να εμφανίζει τα στοιχεία Τ, Ε, Λ, Ο, Σ;

i) Η αρχική μορφή της ουράς είναι:



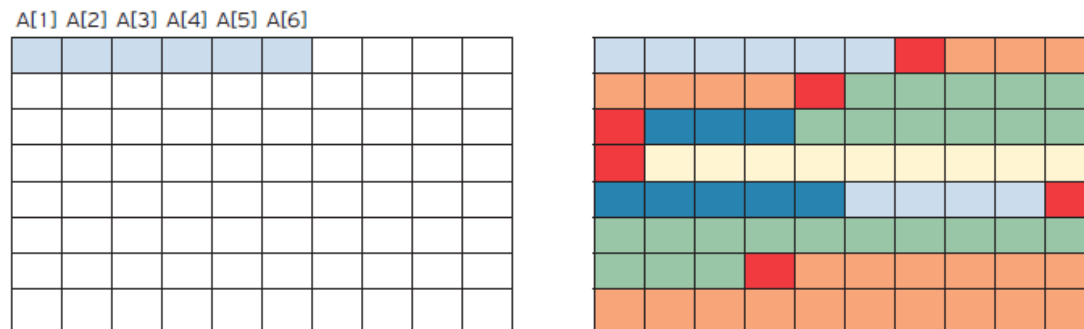
front=1 και rear=5

Ουρά - Ερωτήσεις

α/α	Προτάσεις	Σ	Λ
1	Για την υλοποίηση της ουράς μπορεί να χρησιμοποιηθεί πίνακας.	☐	☐
2	Κατά την εισαγωγή ενός στοιχείου σε ουρά, αυτό τοποθετείται στο μπροστινό άκρο της.	☐	☐
3	Σε μια ουρά κάθε στοιχείο της εξάγεται από το μπροστινό άκρο της.	☐	☐
4	Η απώθηση είναι μια από τις λειτουργίες της ουράς.	☐	☐
5	Η εισαγωγή και η εξαγωγή είναι οι δύο βασικές λειτουργίες της ουράς.	☐	☐
6	Στην ουρά το στοιχείο που μπαίνει πρώτο βγαίνει και πρώτο.	☐	☐
7	Η υλοποίηση της ουράς χρησιμοποιεί μία μεταβλητή-δείκτη για την εκτέλεση των δύο βασικών λειτουργιών της.	☐	☐
8	Η λειτουργία της εξαγωγής μπορεί να εκτελεστεί σε μια γεμάτη ουρά.	☐	☐

Άλλες δομές δεδομένων

- Στις δομές δεδομένων που εξετάσαμε στα προηγούμενα, στοίβες, ουρές το κοινό γνώρισμα είναι ότι υλοποιήθηκαν με χρήση μονοδιάστατου πίνακα.
- Τα στοιχεία ή αλλιώς οι διαδοχικοί κόμβοι αποθηκεύονται σε συνεχόμενες θέσεις της κύριας μνήμης.
- Στην συνέχεια εξετάζουμε τρεις δομές δεδομένων (λίστες, δέντρα, γράφους), στις οποίες οι κόμβοι (ή τα στοιχεία) δεν είναι απαραίτητο να κατέχουν συνεχόμενες θέσεις στην κύρια μνήμη.



Εικόνα 1.3.1. α: Ο πίνακας A στη μνήμη

β: Με το χρώμα κόκκινο αναπαριστάσουμε τα ελεύθερα κελιά

Λίστες

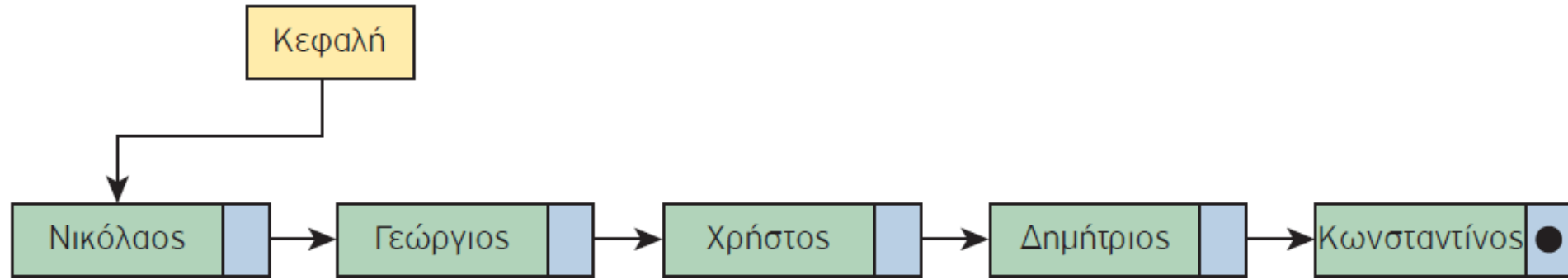
Η λίστα είναι μία συλλογή από αντικείμενα του ίδιου τύπου.

- Η **συνδεδεμένη λίστα** αποτελείται από μία σειρά από κόμβους, που συνήθως βρίσκονται σε απομακρυσμένες θέσεις μνήμης.

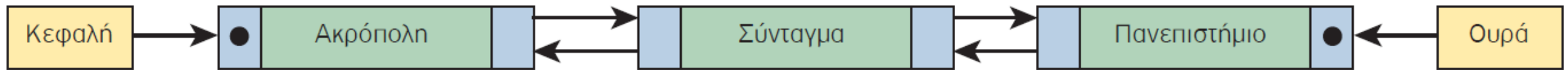
Μια λίστα μπορεί να είναι:

- **απλά συνδεδεμένη**, όπως στο παράδειγμα της σκυταλοδρομίας, 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
- να είναι **διπλά συνδεδεμένη (doubly linked list)**, δηλαδή, να 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. Ένα τέτοιο παράδειγμα είναι οι σταθμοί του μετρό.

Λίστες



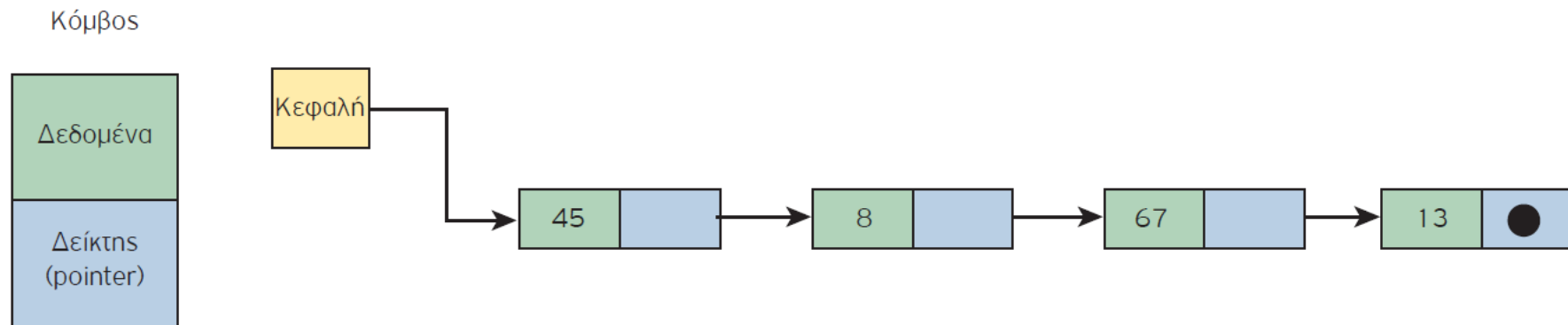
Εικόνα 1.3.7. Το παράδειγμα των αθλητών της σκυταλοδρομίας - Απλά συνδεδεμένη λίστα



Εικόνα 1.3.8. Οι σταθμοί του μετρό - Μια διπλά συνδεδεμένη λίστα

Λίστες

- 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.

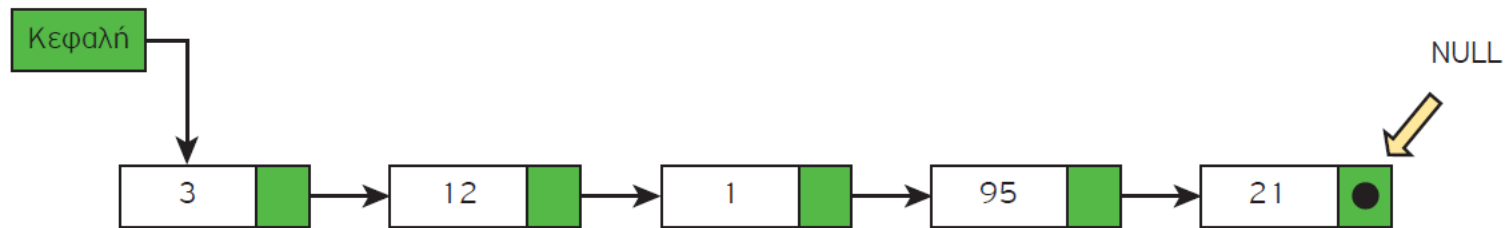


Εικόνα 1.3.2. α: Αναπαράσταση κόμβου

β: Αναπαράσταση απλά συνδεδεμένης λίστας με 4 κόμβους

Λίστες

Η λίστα είναι μία συλλογή από αντικείμενα του ίδιου τύπου



Εικόνα 1.3.3. Μία απλά συνδεδεμένη λίστα με 5 κόμβους.

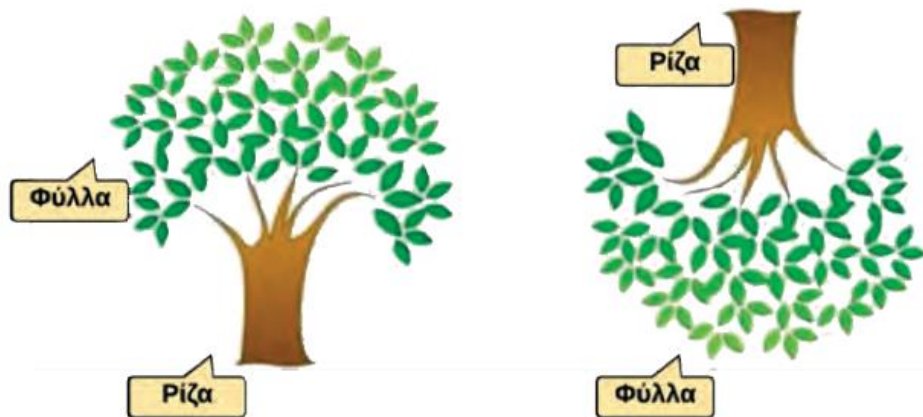
Λίστες - Ερωτήσεις

α/α	Προτάσεις	Σ	Λ
1	Μια απλά συνδεδεμένη λίστα μπορούμε να τη διατρέξουμε και προς τις δύο κατευθύνσεις.	☐	☐
2	Σε μία λίστα δε χρειάζεται να οριστεί ένα αρχικό μέγεθος.	☐	☐
3	Δεν είναι δυνατό να υπάρχει «τυχαία» πρόσβαση σε μια απλά συνδεδεμένη λίστα.	☐	☐
4	Σε μια λίστα, τα στοιχεία δεν μπορούν να προστεθούν ή να αφαιρεθούν από τη μέση της λίστας, παρά μόνο από την αρχή ή το τέλος της.	☐	☐
5	Στη διπλά συνδεδεμένη λίστα τα περιεχόμενα των κόμβων προσπελούνται και από τις δύο κατευθύνσεις.	☐	☐

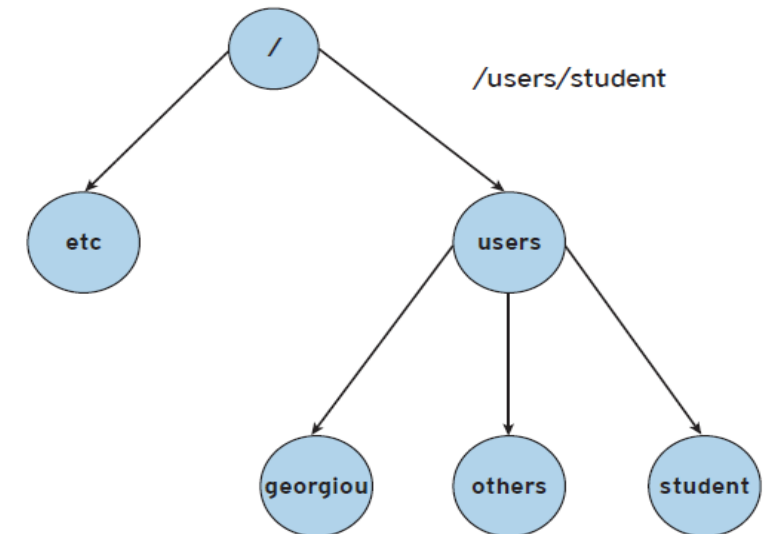
Δένδρα

- Σε μία γραμμική δομή, μετά από κάθε στοιχείο ακολουθεί ένα άλλο στοιχείο εκτός και αν είναι το τελευταίο.
- Σε μια μη γραμμική δομή τα δεδομένα τους δεν ακολουθούν μια σειρά - όπως στους πίνακες ή τις συνδεδεμένες λίστες.

Τα δένδρα είναι μία μη-γραμμική ευέλικτη δομή δεδομένων που χρησιμοποιούνται σε πολλούς τομείς της επιστήμης των υπολογιστών.



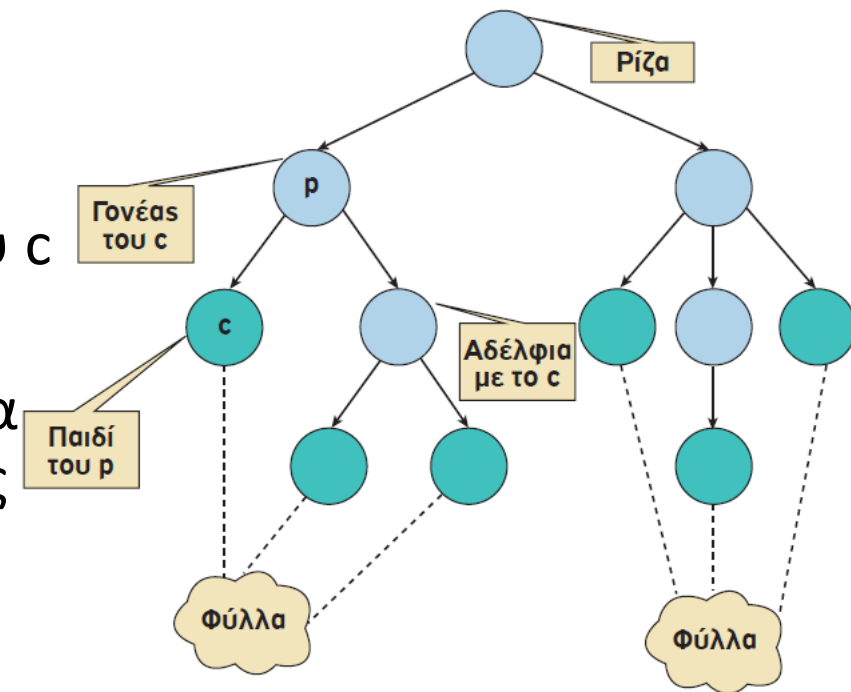
Εικόνα 1.3.9. Τα δένδρα στη φύση και στην Πληροφορική



Εικόνα 1.3.16. Μη-γραμμική ευέλικτη δομή δεδομένων

Δένδρα

- 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 1.3.11 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**».

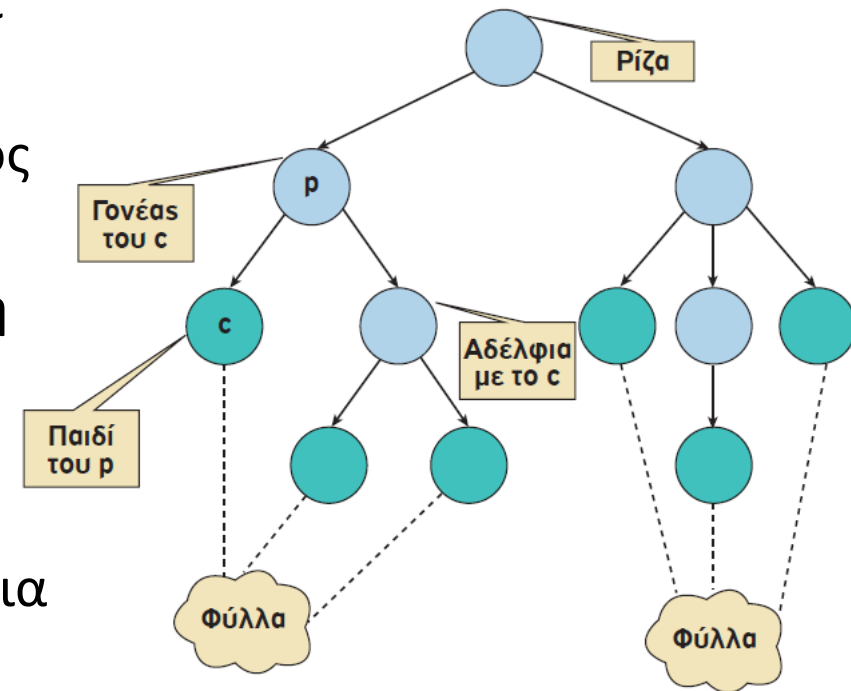


Εικόνα 1.3.11. Σχέσεις μεταξύ των κόμβων ενός δένδρου

Δένδρα

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.

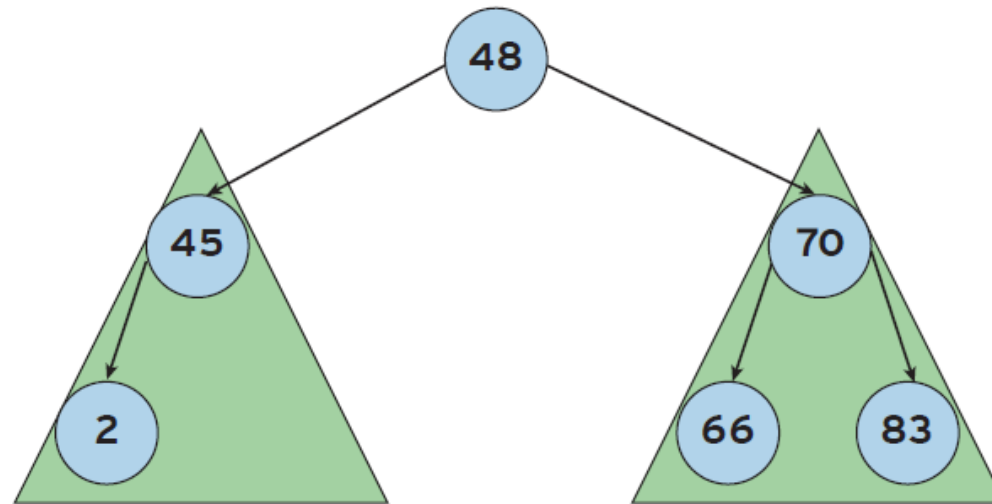


Εικόνα 1.3.1 1. Σχέσεις μεταξύ των κόμβων ενός δένδρου

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.

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.

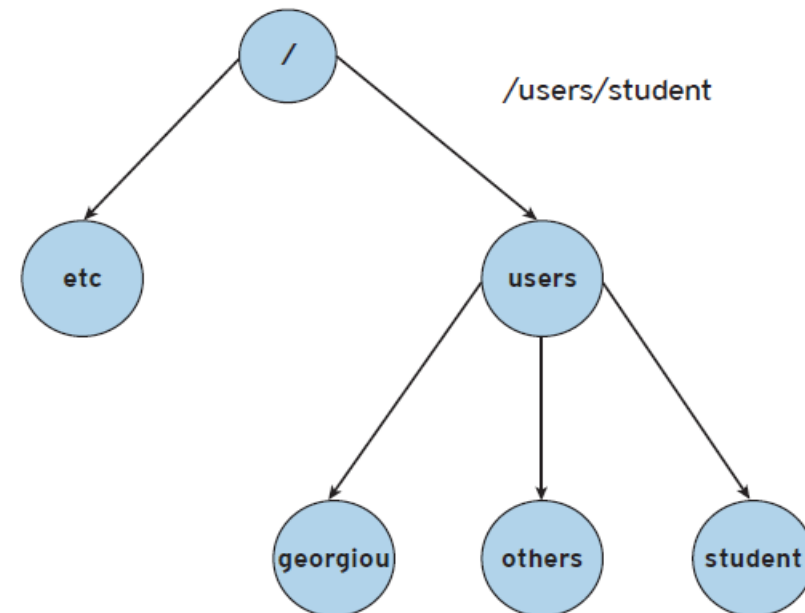


Εικόνα 1.3.14. Αριστερό και δεξιό υποδένδρο του κόμβου 48

Δένδρα

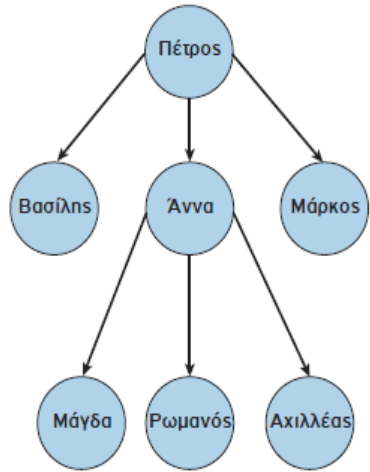
Υπάρχουν δύο λόγοι για τους οποίους τα δένδρα είναι τόσο ισχυρά.

- 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.

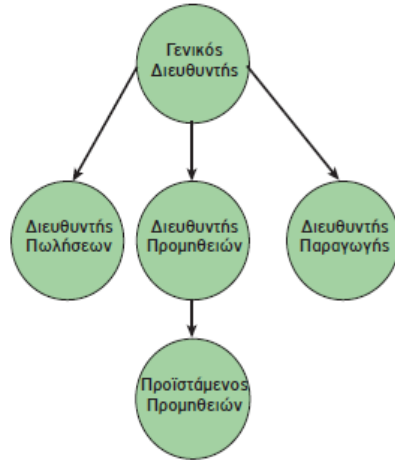


Εικόνα 1.3.16. Μη-γραμμική ευέλικτη δομή δεδομένων

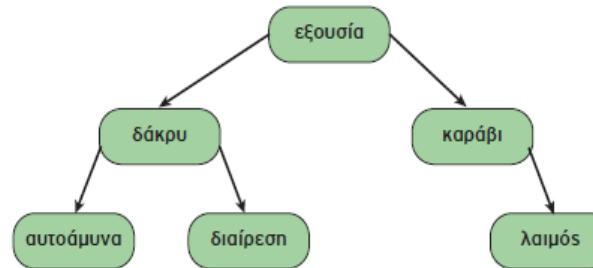
Δένδρα



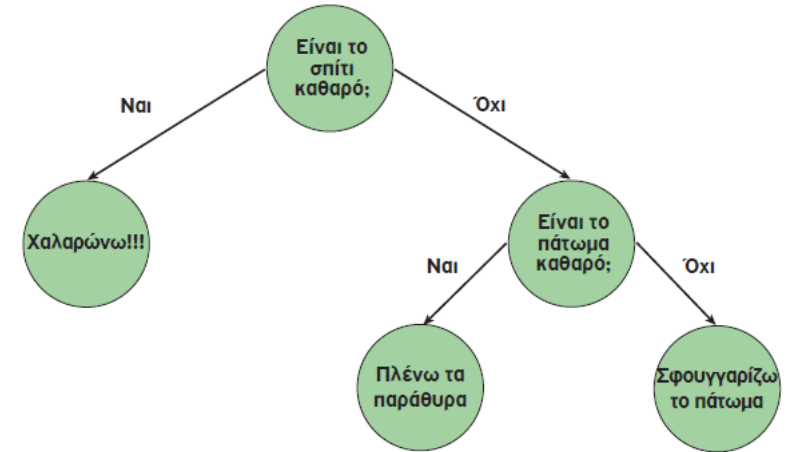
α. Οικογενειακό δένδρο



β. Οργανόγραμμα μιας εταιρείας



γ. Οργάνωση ενός λεξικού



Εικόνα 1.3.19. Δένδρο Απόφασης

Τα δένδρα, πέρα από μια αποτελεσματική οργάνωση και διαχείριση δεδομένων, 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

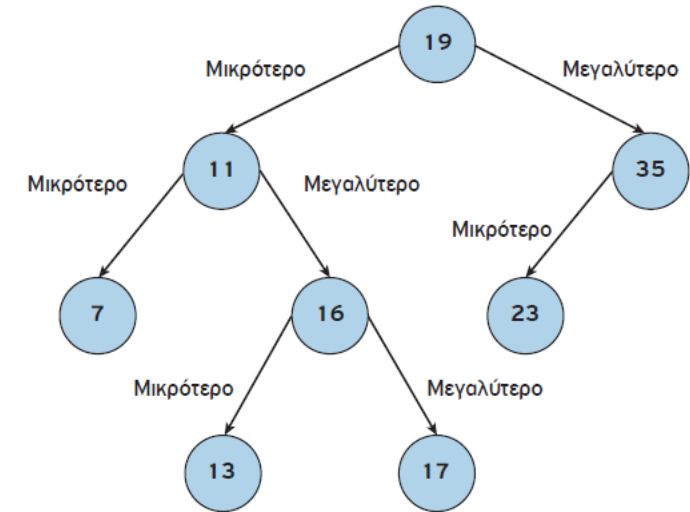
Δυαδικά Δένδρα (binary tree)

Ένα **δυαδικό δένδρο** (binary tree) είναι ένα διατεταγμένο δένδρο (*), στο οποίο κάθε κόμβος έχει το πολύ δύο παιδιά, το αριστερό και το δεξί παιδί.

(*) Διατεταγμένο δένδρο είναι ένα δέντρο στο οποίο η σειρά των παιδιών έχει σημασία (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.

Δυαδικά Δένδρα ΑΝΑΖΗΤΗΣΗΣ

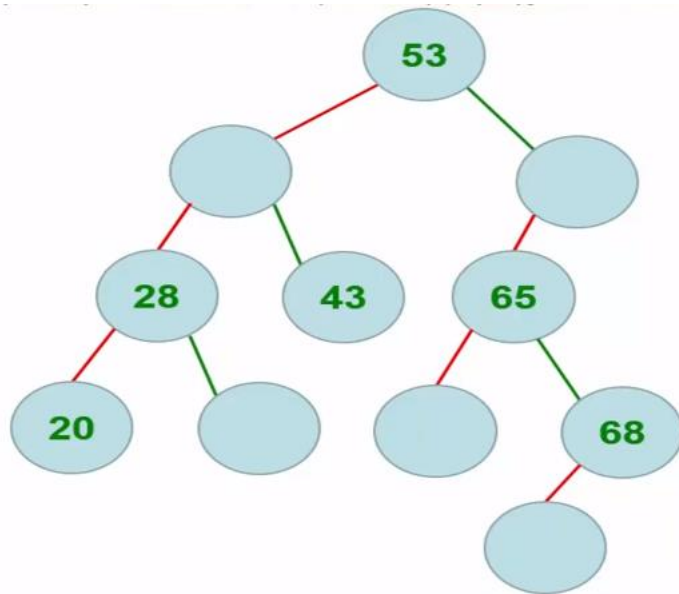
Ένα **δυαδικό δένδρο αναζήτησης** (binary search tree) είναι ένα δυαδικό δένδρο, όπου για κάθε κόμβο u , όλοι οι κόμβοι του αριστερού υποδένδρου έχουν τιμές μικρότερες της τιμής του κόμβου u και όλοι οι κόμβοι του δεξιού υποδένδρου έχουν τιμές μεγαλύτερες (ή ίσες) της τιμής του κόμβου u . Για λόγους απλούστευσης θεωρούμε ότι δεν υπάρχουν τιμές ίσες με την τιμή του κόμβου u .



Εικόνα 1.3.22. Δυαδικό δένδρο αναζήτησης

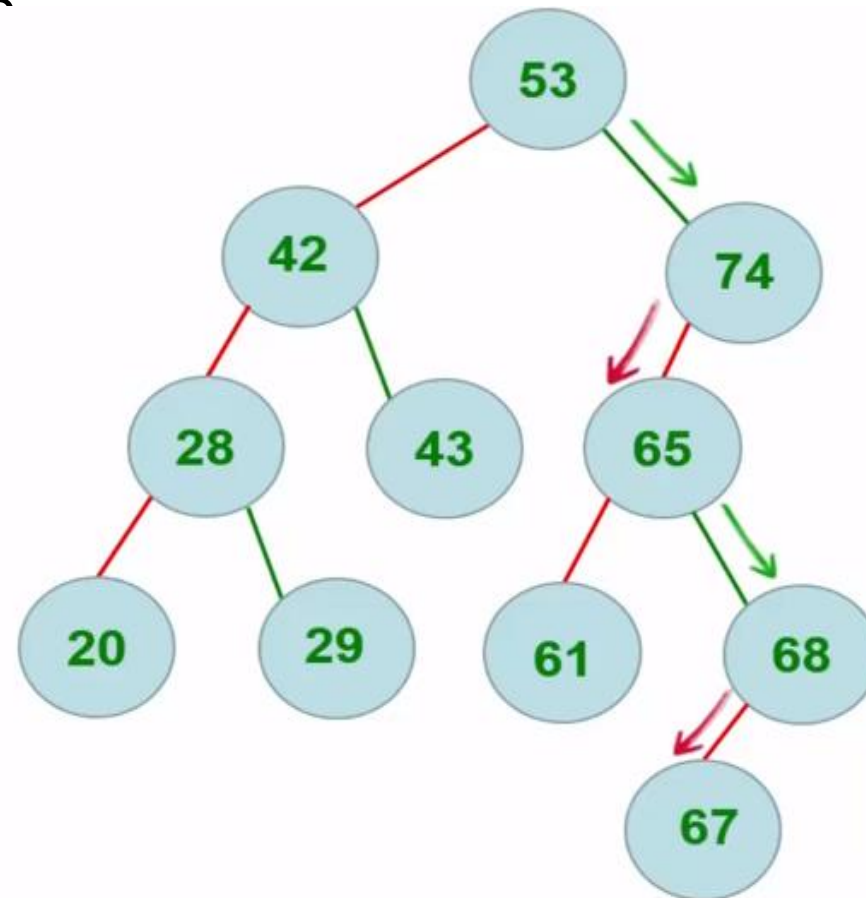
ΑΣΚΗΣΗ 1 Δυαδικά Δένδρα

Δίνεται το παρακάτω δυαδικό δένδρο. Να συμπληρώσετε τους κενούς κόμβους βάζοντας έναν από τους παρακάτω αριθμούς σε κάθε κόμβο ώστε να προκύψει δυαδικό δέντρο αναζήτησης: 42, 29, 74, 61, 67



ΛΥΣΗ ΑΣΚΗΣΗΣ 1 Δυαδικά Δένδρα

Δίνεται το παρακάτω δυαδικό δένδρο. Να συμπληρώσετε τους κενούς κόμβους βάζοντας έναν από τους παρακάτω αριθμούς σε κάθε κόμβο ώστε να προκύψει δυαδικό δέντρο αναζήτησης: 42, 29, 74, 61, 67



ΑΣΚΗΣΗ 2 Δυαδικά Δένδρα

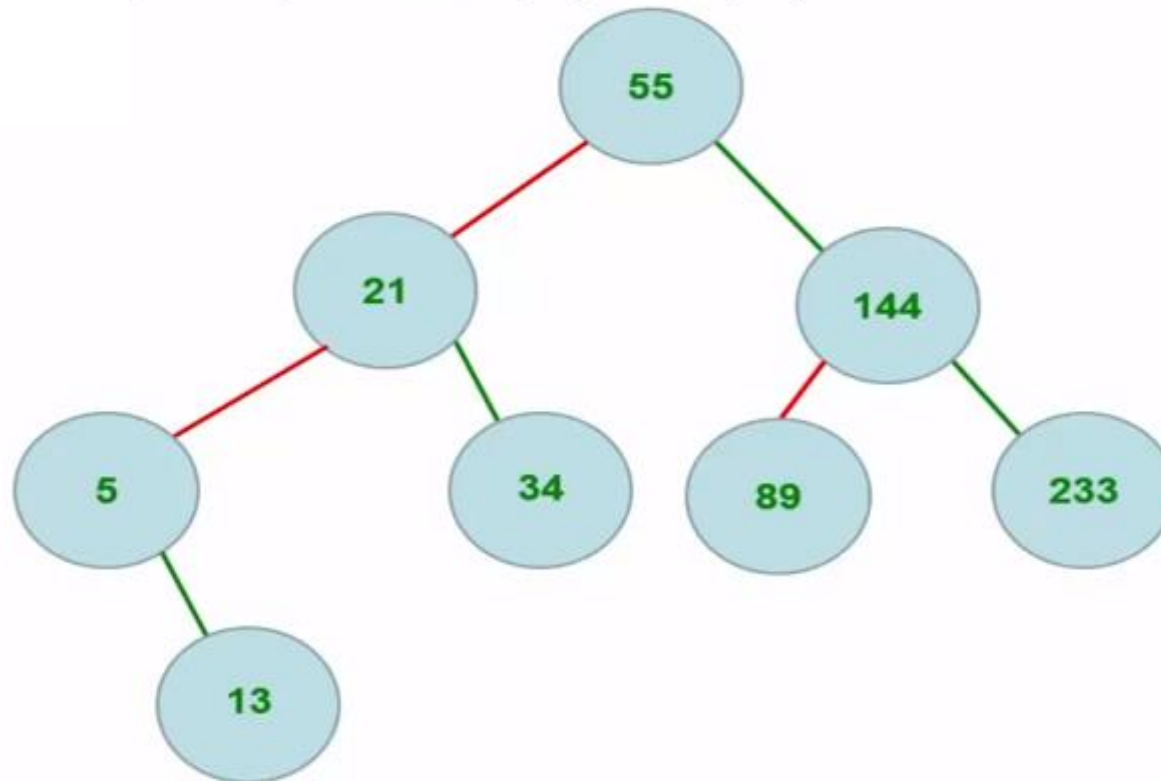
Δίνεται η ακολουθία αριθμών 55, 144, 21, 34, 89, 5, 233, 13 οι οποίοι εισάγονται σε δυαδικό δέντρο αναζήτησης με την σειρά.

1. Να σχεδιάσε το τελικό δέντρο μετά την τοποθέτηση αριθμών.

ΛΥΣΗ ΑΣΚΗΣΗ 2 Δυαδικά Δένδρα

Δίνεται η ακολουθία αριθμών 55, 144, 21, 34, 89, 5, 233, 13 οι οποίοι εισάγονται σε δυαδικό δέντρο αναζήτησης με την σειρά.

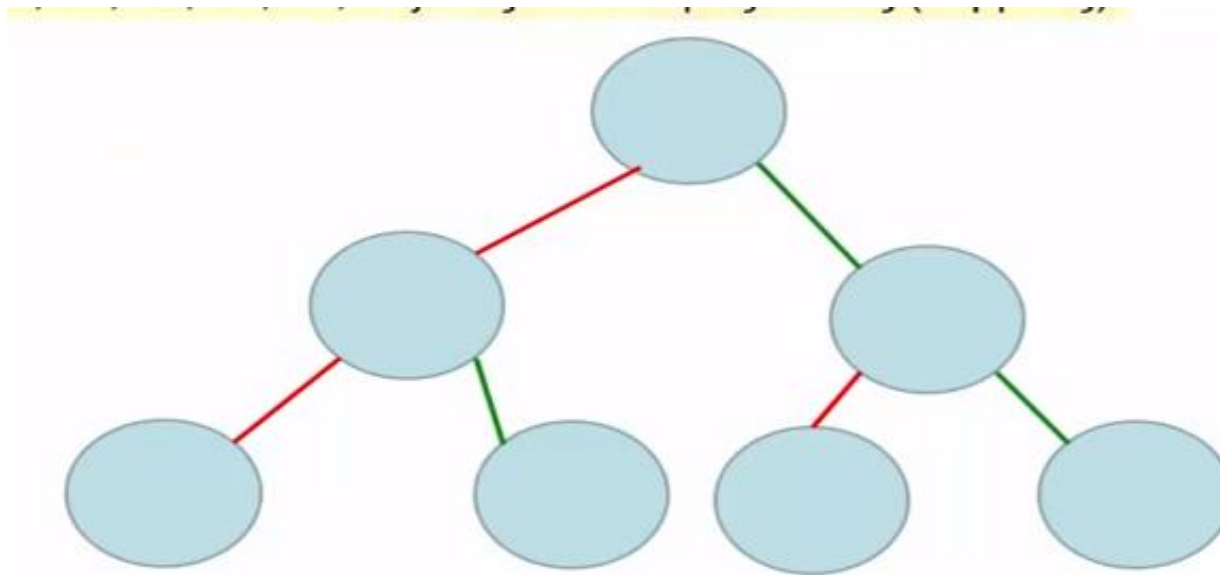
1. Να σχεδιάσε το τελικό δέντρο μετά την τοποθέτηση αριθμών.



ΑΣΚΗΣΗ 3 Δυαδικά Δένδρα

Δίνεται το παρακάτω ελλιπές δυαδικό δέντρο αναζήτησης (όπου λείπουν οι ακμές, 2 κόμβοι και τα περιεχόμενα τα κόμβων).

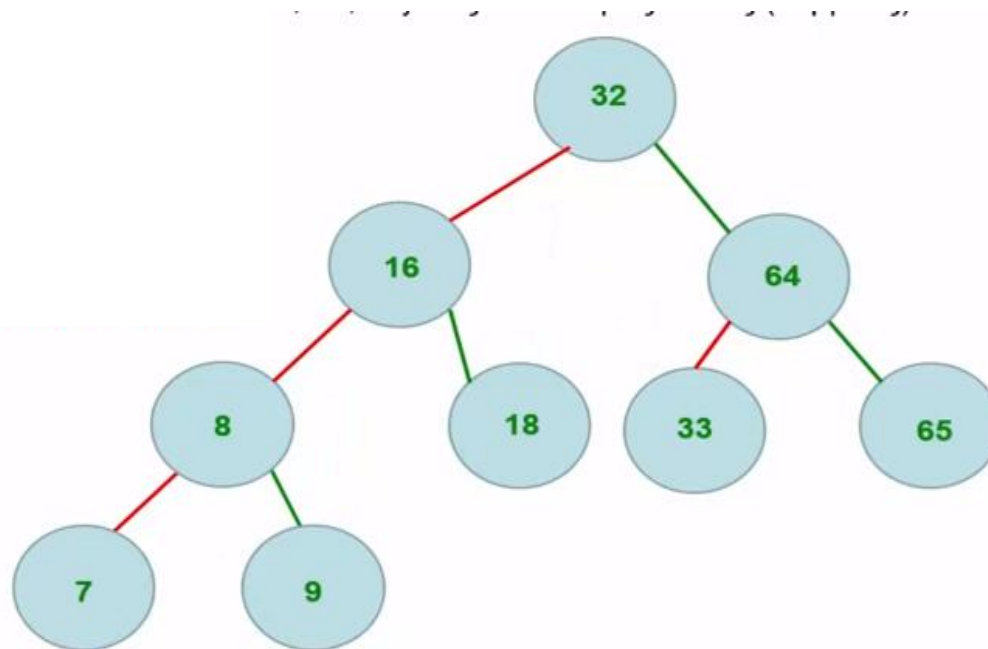
Να προσθέσετε στην κατάλληλη θέση τις ακμές που λείπουν μεταξύ των κόμβων, τους 2 κόμβους που λείπουν και να εισάγετε τις τιμές {7, 8, 9, 16, 18, 32, 33, 64, 65} στις κατάλληλες θέσεις (κόμβους).



ΛΥΣΗ ΑΣΚΗΣΗ 3 Δυαδικά Δένδρα

Δίνεται το παρακάτω ελλιπές δυαδικό δέντρο αναζήτησης (όπου λείπουν οι ακμές, 2 κόμβοι και τα περιεχόμενα τα κόμβων).

Να προσθέσετε στην κατάλληλη θέση τις ακμές που λείπουν μεταξύ των κόμβων, τους 2 κόμβους που λείπουν και να εισάγετε τις τιμές {7, 8, 9, 16, 18, 32, 33, 64, 65} στις κατάλληλες θέσεις (κόμβους).



Δένδρα - Ερωτήσεις

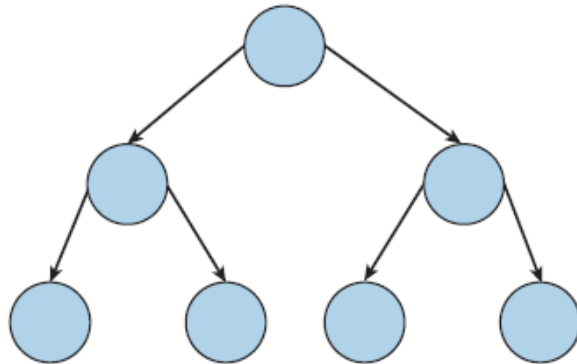
α/α	Προτάσεις	Σ	Λ
1	Η ρίζα ενός δένδρου δεν μπορεί ποτέ να είναι φύλλο.	☐	☐
2	Σε ένα δυαδικό δένδρο, φύλλα συναντάμε μόνο στο αριστερό υποδένδρο.	☐	☐
3	Σε ένα δυαδικό δένδρο, κάθε κόμβος-γονέας μπορεί να έχει το πολύ δύο παιδιά	☐	☐
4	Δεν είναι δυνατό να υπάρχουν δύο διαφορετικές διαδρομές από την ρίζα προς έναν άλλον κόμβο ενός δένδρου.	☐	☐
5	Σε ένα δυαδικό δένδρο, κάθε κόμβος έχει μηδέν, ένα ή δύο υποδένδρα.	☐	☐
6	Η ρίζα ενός δένδρου είναι ο μόνος κόμβος ενός δένδρου που δεν έχει γονέα.	☐	☐
7	Τα φύλλα ενός δένδρου είναι απομονωμένοι κόμβοι που δε συνδέονται με άλλους κόμβους.	☐	☐
8	Σε ένα δένδρο, κάθε κόμβος-γονέας μπορεί να έχει οποιονδήποτε αριθμό παιδιών	☐	☐
9	Μπορούν να υπάρχουν διαφορετικές δομές δυαδικών δένδρων αναζήτησης που αποθηκεύουν τα ίδια στοιχεία.	☐	☐
10	Κάθε δένδρο είναι γράφος	☐	☐

Γράφοι

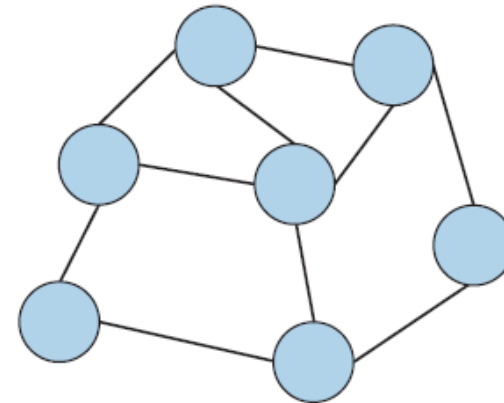
Θεμελιώδες χαρακτηριστικό των μη γραμμικών δομών είναι ότι τα δεδομένα τους δεν ακολουθούν μια σειρά - όπως στους πίνακες ή τις συνδεδεμένες λίστες.

Ένα δένδρο θα είναι πάντα ένα γράφος, αλλά δεν είναι όλοι οι γράφοι δένδρα.

Τι είναι αυτό που κάνει ένα δένδρο διαφορετικό από έναν γράφο;



α: Δένδρο



β: Γράφος

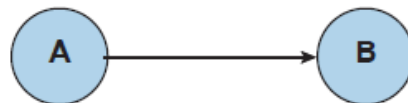
Εικόνα 1.3.27.

Γράφοι

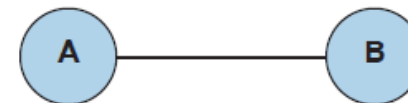
Ένας **γράφος (graph)** είναι μία δομή που αποτελείται από ένα σύνολο κόμβων (ή σημείων ή κορυφών) και ένα σύνολο γραμμών (ή ακμών ή τόξων) που ενώνουν μερικούς ή όλους τους κόμβους. Ο γράφος αποτελεί την πιο γενική δομή δεδομένων, με την έννοια ότι όλες οι προηγούμενες δομές που παρουσιάστηκαν μπορούν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

Τύποι Γράφων

- **Κατευθυνόμενοι** γράφοι, όπου όλες οι ακμές σε έναν γράφο έχουν κατεύθυνση.
- **Μη κατευθυνόμενοι** γράφοι, όπου όλες οι ακμές σε έναν γράφο δεν έχουν κατεύθυνση.



α: Κατευθυνόμενη ακμή

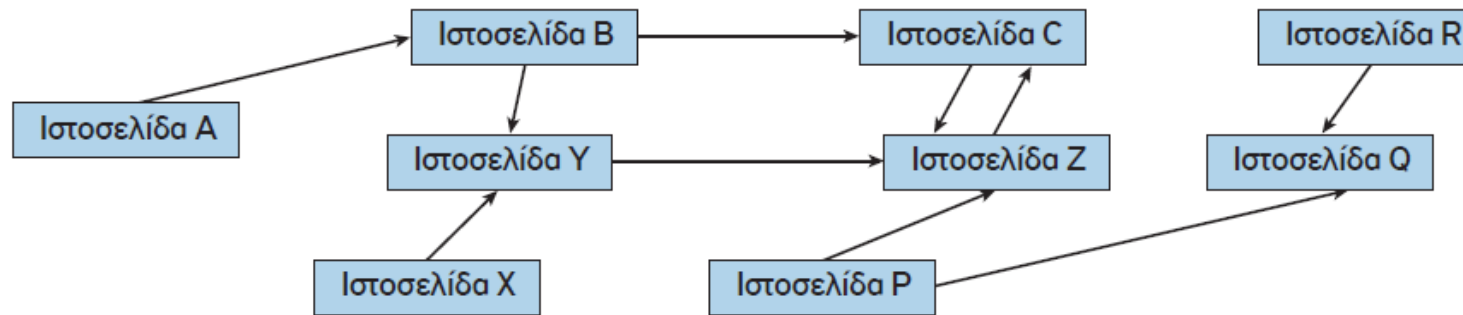


β: μη κατευθυνόμενη ακμή

Γράφοι

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**.

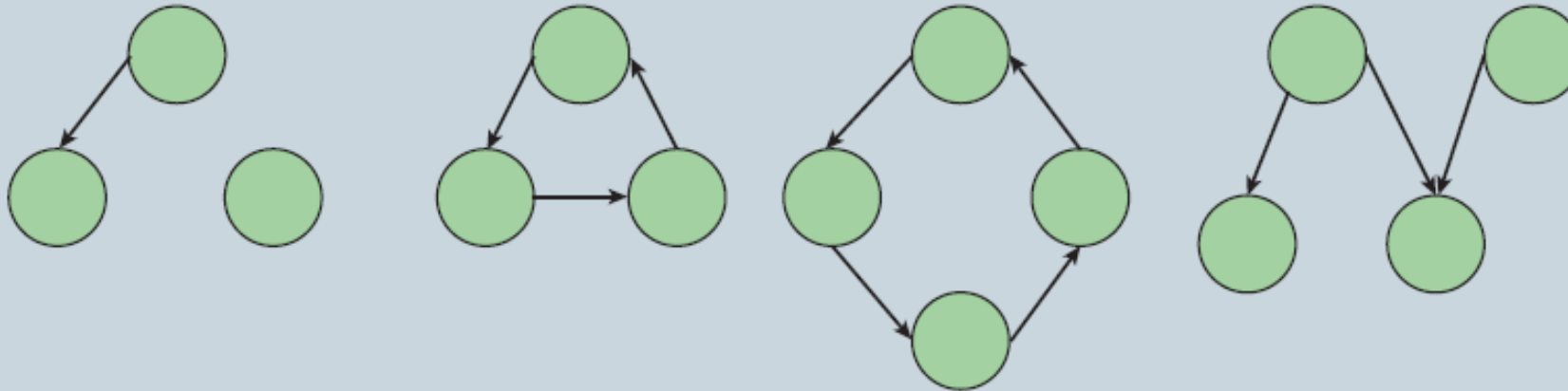


Εικόνα 1.3.29. Παγκόσμιος Ιστός – Η ιστοσελίδα R περιέχει σύνδεσμο προς την ιστοσελίδα Q

Γράφοι - Ερωτήσεις

Ε.4: Δένδρα ή Γράφοι;

Στην Εικόνα 1.3.31 ποιες από τις παρακάτω δομές είναι δένδρα και ποιες είναι γράφοι. Προσπαθήστε να εξηγήσετε το γιατί.



α. _____ β. _____ γ. _____ δ. _____

Εικόνα 1.3.31. Εντοπίστε τα δένδρα και τους γράφους

Γράφοι - Ερωτήσεις

Ε.5: Επιλέξτε τη σωστή απάντηση.

1. Ποια από τις βασικές δομές δεδομένων είναι η πιο κατάλληλη για να αναπαραστήσετε τη δομή των καταλόγων, των υποκαταλόγων και των αρχείων στον σκληρό σας δίσκο:

- ☐ πίνακας
- ☐ λίστα
- ☐ δένδρο
- ☐ ουρά
- ☐ στοίβα

2. Ο κατάλογος των φοιτητών που εγγράφονται σε ένα μάθημα είναι ταξινομημένος αλφαβητικά με βάση το ονοματεπώνυμο και περιλαμβάνει ένα σύνολο πληροφοριών σχετικών με τον φοιτητή, όπως είναι ο κωδικός του φοιτητή, η ημερομηνία γέννησης, το φύλο, η διεύθυνση, ο αριθμός τηλεφώνου κ.λπ. Επιλέξτε ποια από τις παρακάτω δομές δεδομένων είναι καταλληλότερη για την αναπαράσταση αυτών των πληροφοριών:

- ☐ στοίβα
- ☐ δένδρο
- ☐ λίστα
- ☐ ουρά