

1.3 Άλλες δομές δεδομένων

Κοινό γνώρισμα των παραπάνω δομών που υλοποιήθηκαν με χρήση μονοδιάστατου πίνακα είναι ότι οι διαδοχικοί κόμβοι αποθηκεύονται σε συνεχόμενες θέσεις της κύριας μνήμης. Στην ενότητα αυτή 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.

1.3.1 Λίστες

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.

Πώς όμως θα μπορούσατε να αποθηκεύσετε έναν συγκεκριμένο αριθμό στοιχείων του ίδιου τύπου, για παράδειγμα: τα μέρη που θα θέλατε να επισκεφτείτε; Μία από τις δομές δεδομένων που έχετε δει μέχρι τώρα και που θα μπορούσατε να χρησιμοποιήσετε για να αποθηκεύσετε τα στοιχεία αυτά είναι οι πίνακες. Πολύ απλά και εύκολα μπορείτε να τους δημιουργήσετε και να προσπελάστε άμεσα οποιοδήποτε στοιχείο τους.

Το γεγονός, όμως, ότι το μέγεθος ενός πίνακα παραμένει σταθερό (στατική δομή δεδομένων), δυσχεραίνει την προσθήκη και τη διαγραφή στοιχείων. Θεωρείστε την περίπτωση που θέλετε να προσθέσετε στον πίνακα και άλλα μέρη στα οποία θέλετε να ταξιδέψετε. Τι θα κάνετε αν ο πίνακας είναι γεμάτος; Μια πρώτη απάντηση θα ήταν να δημιουργήσουμε έναν νέο μεγαλύτερο πίνακα και να αντιγράψουμε σε αυτόν τα στοιχεία του προηγούμενου πίνακα. Έτσι, θα είχαμε μεγαλύτερο χώρο για να προσθέσουμε και νέα στοιχεία. Θεωρείτε ότι αυτή είναι η ιδανική λύση, όταν η προσθήκη και η διαγραφή στοιχείων αποτελεί συχνό φαινόμενο;

Ας μελετήσουμε μία άλλη λύση, δηλαδή μία νέα δυναμική δομή δεδομένων, τις **συνδεδεμένες λίστες**, που απαντάει σε αυτές τις προκλήσεις. Αρχικά, ας θεωρήσουμε το πλέγμα των κελιών που παρουσιάζεται στην Εικόνα 1.3.1.α ως τη μνήμη του υπολογιστή. Σε κάθε κελί αντιστοιχίζεται μία διεύθυνση. Για να προσπελάσουμε ένα οποιαδήποτε κελί θα πρέπει να γνωρίζουμε τη διεύθυνσή του. Στην Εικόνα 1.3.1.α παρουσιάζεται ο χώρος που καταλαμβάνει ο πίνακας Α με το γαλάζιο χρώμα. Ο πίνακας αυτός αποτελείται από έξι στοιχεία τα οποία είναι αποθηκευμένα σε συνεχόμενες θέσεις μνήμης.

The diagram illustrates the memory layout for an array of 10 elements, A[1] through A[10]. The left grid shows the first 5 elements (A[1] to A[5]) in blue. The right grid shows the remaining 5 elements (A[6] to A[10]) in various colors: red (A[6]), orange (A[7]), green (A[8]), yellow (A[9]), and light blue (A[10]).

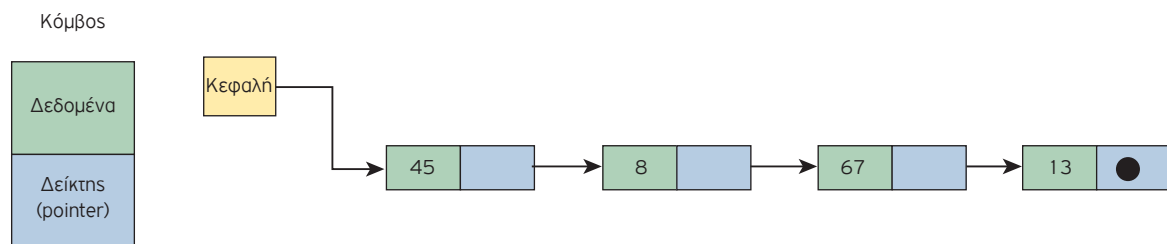
Εικόνα 1.3.1. α: Ο πίνακας A στη μνήμη

β: Με το χρώμα κόκκινο αναπαριστάνουμε τα ελεύθερα κελιά

Συνήθως όμως, η μνήμη του υπολογιστή δεν είναι τόσο καθαρή και τακτοποιημένη. Στην Εικόνα 1.3.1.β φαίνονται τα διάσπαρτα (μη συνεχόμενα) ελεύθερα κελιά με το κόκκινο χρώμα. Στην προκειμένη περίπτωση, δεν μπορείτε να «τοποθετήσετε» έναν πίνακα στη μνήμη, διότι δεν υπάρχουν συνεχόμενες ελεύθερες θέσεις. Μπορείτε, όμως, να χρησιμοποιήσετε τις διάσπαρτες αυτές ελεύθερες θέσεις για να αποθηκεύσετε τα δεδομένα σας, αν θεωρήσετε ότι αποτελούν «στοιχεία» μίας συνδεδεμένης λίστας.

Η συνδεδεμένη λίστα αποτελείται από μία σειρά από κόμβους, που συνήθως βρίσκονται σε απομακρυσμένες θέσεις μνήμης. Κάθε κόμβος αποτελείται από δύο κύρια τμήματα (Εικόνα 1.3.2.α). 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.



Εικόνα 1.3.2. α: Αναπαράσταση κόμβου

β: Αναπαράσταση απλά συνδεδεμένης λίστας με 3 κόμβους

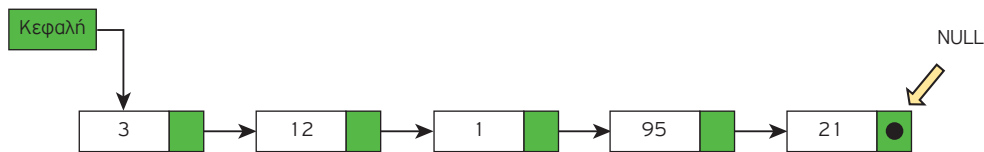
Ένα παράδειγμα απλά συνδεδεμένης λίστας με τρεις κόμβους, που περιέχουν ακραίους, δίνεται στην Εικόνα 1.3.2.β. Ο δείκτης του τελευταίου κόμβου της λίστας έχει ως τιμή το NULL (κενό) και τον αναπαριστούμε συμβολικά με το σύμβολο «●». Οι δείκτες των υπολοίπων κόμβων περιέχουν τη διεύθυνση του επόμενου κόμβου αντίστοιχα και τους απεικονίζουμε συμβολικά με ένα βέλος για να υποδηλώσουμε τη σύνδεση μεταξύ του προηγούμενου και του επόμενου κόμβου. Κάθε λίστα συνοδεύεται από έναν δείκτη με το όνομα «Κεφαλή» (head), που δείχνει στον πρώτο κόμβο της λίστας, δηλαδή περιέχει τη διεύθυνση του πρώτου κόμβου της λίστας.



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)**.



Εικόνα 1.3.3. Μία απλά συνδεδεμένη λίστα με 5 κόμβους.

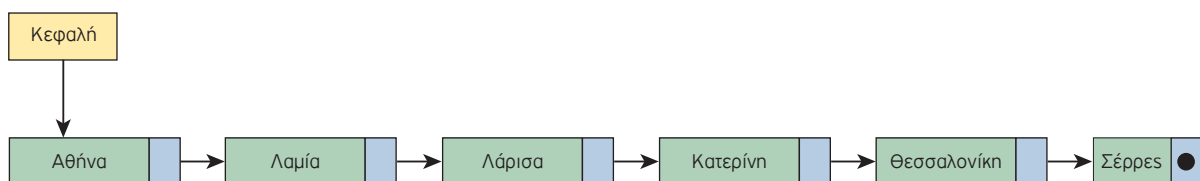
Οι κόμβοι μιας λίστας δεν έχουν ονόματα. Γνωρίζουμε μόνο τις διευθύνσεις τους, που είναι αποθηκευμένες στους προηγούμενους κόμβους και αυτές θα αξιοποιήσουμε για να τους προσπελάσουμε. Επομένως, αν θελήσουμε να έχουμε πρόσβαση στον τέταρτο κόμβο μιας λίστας, για να επεξεργαστούμε τα δεδομένα που περιέχει, 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.



Πρόσβαση στους κόμβους μιας συνδεδεμένης λίστας

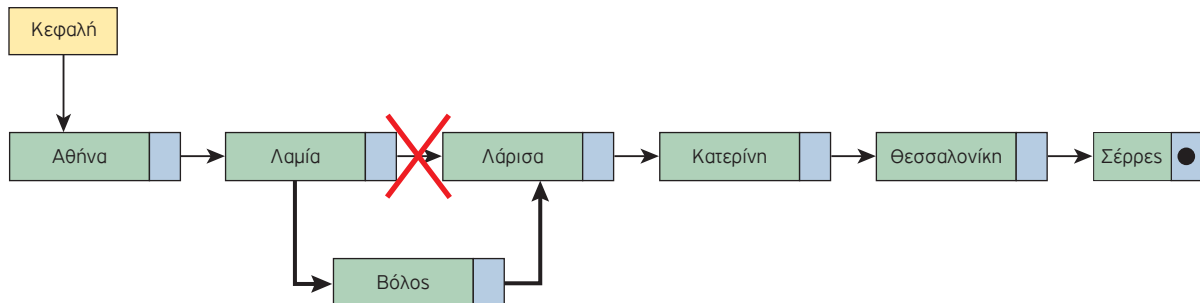
Οι κόμβοι μιας (απλά) συνδεδεμένης λίστας είναι διατεταγμένοι σε μια συγκεκριμένη σειρά, χωρίς αυτό να σημαίνει ότι αποθηκεύονται σε συνεχόμενες θέσεις στη μνήμη. Αντίθετα, είναι διασκορπισμένοι σε όλη τη μνήμη και η σύνδεση μεταξύ τους γίνεται μέσω των δεικτών. Έχουμε άμεση πρόσβαση μόνο στον πρώτο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

Έστω ότι σχεδιάζουμε στη συνέχεια ένα ταξίδι στην Κεντρική Μακεδονία. Η αφετηρία μας είναι η Αθήνα και το τέρμα του ταξιδιού είναι οι Σέρρες. Έχουμε αποθηκεύσει σε μια συνδεδεμένη λίστα τις πόλεις που είναι πιθανό να επισκεφθούμε (Εικόνα 1.3.4).



Εικόνα 1.3.4. Μια απλά συνδεδεμένη λίστα

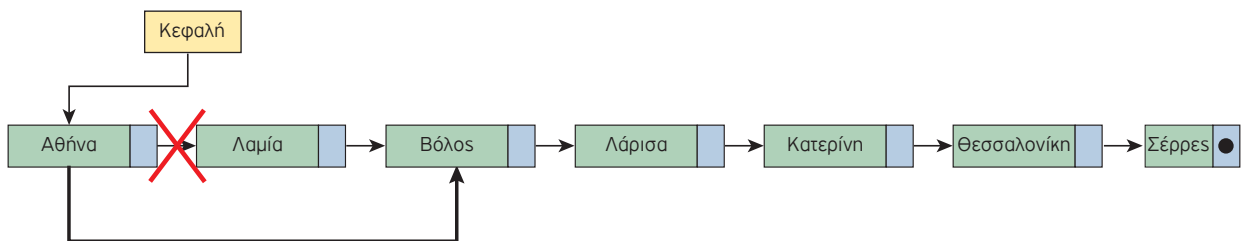
Λίγο πριν το ταξίδι, αποφασίζουμε να επισκεφθούμε τον Βόλο, αμέσως μετά τη Λαμία και πριν φτάσουμε στη Λάρισα. Επομένως, χρειάζεται να εισαγάγουμε ένα νέο κόμβο στη λίστα μας, κάνοντας την κατάλληλη διευθέτηση των δεικτών (Εικόνα 1.3.5).



Εικόνα 1.3.5. Εισαγωγή νέου κόμβου στη λίστα

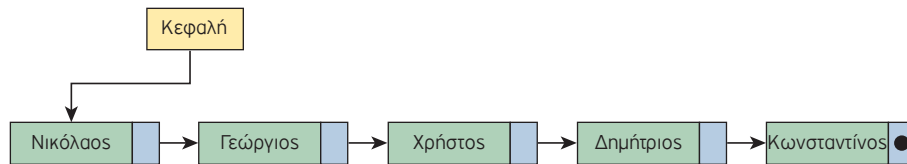
Όπως φαίνεται και στην Εικόνα 1.3.5, οι απαιτούμενες ενέργειες για την εισαγωγή (παρεμβολή) του νέου κόμβου είναι ο δείκτης του δεύτερου κόμβου να δείχνει τον νέο κόμβο και ο δείκτης του νέου κόμβου να δείχνει τον τέταρτο κόμβο (δηλαδή, να πάρει την τιμή που είχε πριν την εισαγωγή ο δείκτης του τρίτου κόμβου). Με αυτόν τον τρόπο, οι κόμβοι της λίστας διατηρούν τη λογική τους σειρά, αλλά οι φυσικές θέσεις στη μνήμη μπορεί να είναι τελείως διαφορετικές.

Αν, επιπλέον, αποφασίσουμε να μην επισκεφθούμε τη Λαμία, θα πρέπει να διαγράψουμε τον παραπάνω κόμβο από τη λίστα κάνοντας μια εκ νέου διευθέτηση των αντίστοιχων δεικτών (Εικόνα 1.3.6). Έτσι, για τη διαγραφή ενός κόμβου αρκεί ν' αλλάξει τιμή ο δείκτης του προηγούμενου κόμβου και να δείχνει πλέον στον επόμενο αυτού που διαγράφεται, όπως φαίνεται στην Εικόνα 1.3.6. Ο κόμβος που διαγράφηκε (ο δεύτερος) αποτελεί «άχρηστο δεδομένο» και ο χώρος μνήμης που καταλάμβανε, παραχωρείται για άλλη χρήση.



Εικόνα 1.3.6. Διαγραφή κόμβου από τη λίστα

Ας δούμε ένα δεύτερο παράδειγμα με μία συνδεδεμένη λίστα. Ας φανταστούμε έναν αγώνα σκυταλοδρομίας (Εικόνα 1.3.7). Οι αθλητές που λαμβάνουν μέρος είναι τοποθετημένοι με μια συγκεκριμένη σειρά και ο κάθε αθλητής για να ξεκινήσει να τρέχει πρέπει να περιμένει τον προηγούμενο του. Σε περιπτώσεις όπως αυτή, που κάθε στοιχείο συνδέεται με το αμέσως επόμενο, χρησιμοποιούμε τη δομή δεδομένων της συνδεδεμένης λίστας για να αποθηκεύσουμε τις τιμές.

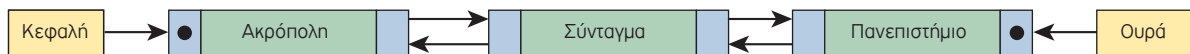


Εικόνα 1.3.7. Το παράδειγμα των αθλητών της σκυταλοδρομίας - Απλά συνδεδεμένη λίστα



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

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 ή 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.



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

Όπως φαίνεται και στην Εικόνα 1.3.8., 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.



Σχεδιάστε οδικές μεταβάσεις

Παρατηρείστε τα σχήματα των απλών και διπλών συνδεδεμένων λιστών. Μπορείτε να εντοπίσετε τις διαφορές τους; Σχεδιάστε στη συνέχεια μία διπλά συνδεδεμένη λίστα που θα αποτυπώνει την οδική μετάβαση μεταξύ Αθηνών και Ιωαννίνων μέσω της νέας Ιονίας οδού.

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).

Διαφορές Λίστας σε σχέση με τον Πίνακα – Πλεονεκτήματα – Μειονεκτήματα

Συνοψίζοντας, θα μπορούσε κάποιος πρόχειρα να συμπεράνει ότι η λίστα δε διαφέρει από τον πίνακα. Παρατηρώντας όμως πιο προσεκτικά, είναι εύκολο να εντοπίσει σημαντικές **διαφορές** μεταξύ τους, οι οποίες παρατίθενται παρακάτω:

- 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 εξ αρχής 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.

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 κατανεμημένοι (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.



Διαφορές στην προσπέλαση μεταξύ δομών δεδομένων

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

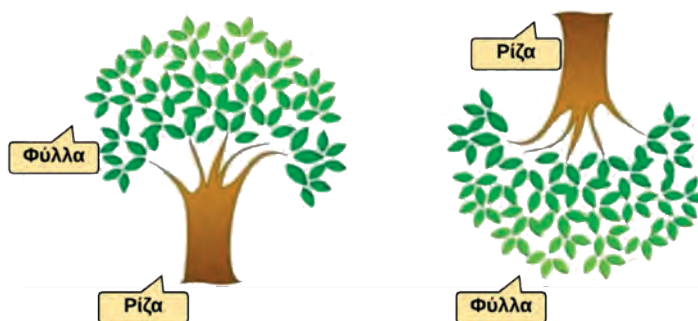
Βασικές πράξεις των συνδεδεμένων λιστών

Οι βασικές πράξεις των συνδεδεμένων λιστών είναι οι παρακάτω:

- 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).

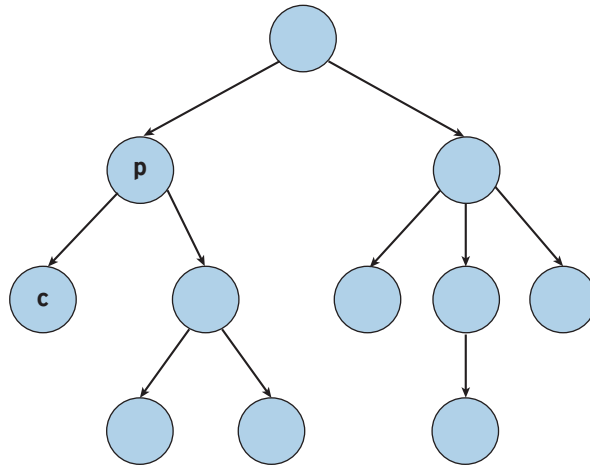
1.3.2 Δένδρα

Μερικές φορές τα πράγματα δεν είναι γραμμικά, όπως έχουμε δει μέχρι τώρα. Σε μία γραμμική δομή, μετά από κάθε στοιχείο ακολουθεί ένα άλλο στοιχείο εκτός και αν είναι το τελευταίο. Τι συμβαίνει όμως στις περιπτώσεις όπου μετά από ένα στοιχείο ακολουθεί όχι ένα, αλλά δύο, τρία ή και περισσότερα στοιχεία; Μπορείτε να φέρετε στον νου σας εικόνες, όπου εμφανίζεται αυτή η συγκεκριμένη συνδεσμολογία των στοιχείων; Μήπως μία από αυτές τις εικόνες είναι το δένδρο; Παρατηρείστε ένα δένδρο στη φύση. Έχει ρίζα και φύλλα, όπως φαίνεται στο αριστερό μέρος της Εικόνας 1.3.9. Η ρίζα είναι στο κάτω μέρος και τα φύλλα στο πάνω μέρος. Στην επιστήμη της Πληροφορικής, όμως, βλέπουμε τα δένδρα ανάποδα. Τα φύλλα είναι στο κάτω μέρος και η ρίζα στο επάνω μέρος, όπως φαίνεται στο δεξιό μέρος της Εικόνας 1.3.9.



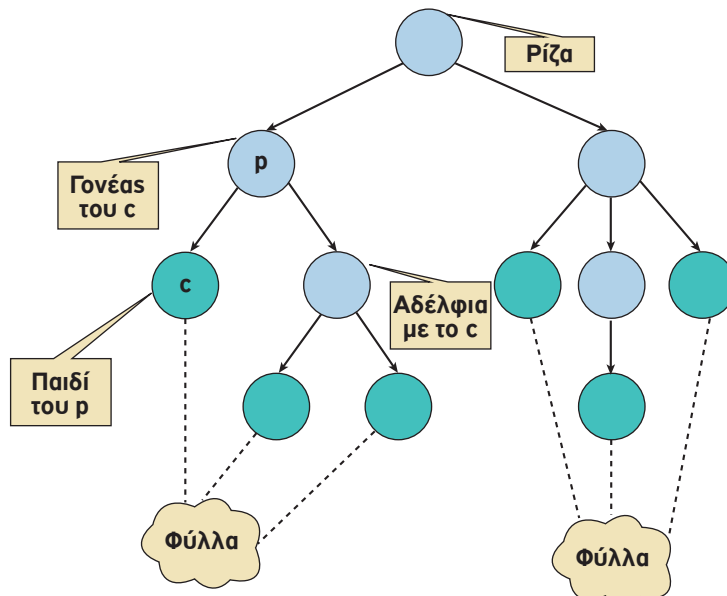
Εικόνα 1.3.9. Τα δένδρα στη φύση και στην Πληροφορική

Ας ξανασχεδιάσουμε το δένδρο που βρίσκεται στο δεξιό μέρος της Εικόνας 1.3.9 με έναν πιο αφαιρετικό τρόπο και στη συνέχεια ας προσπαθήσουμε να ορίσουμε αυτό το οποίο βλέπουμε σε ένα δένδρο, προσδιορίζοντας τη δομή του. Έτσι λοιπόν, θα λέγαμε ότι ένα **δένδρο** αποτελείται από κόμβους, οι οποίοι συνδέονται μεταξύ τους με ακμές. Στην Εικόνα 1.3.10 βλέπουμε τους κόμβους με γαλάζιο χρώμα και τις ακμές ως βέλη μεταξύ των κόμβων.



Εικόνα 1.3.10. Δομή ενός δένδρου

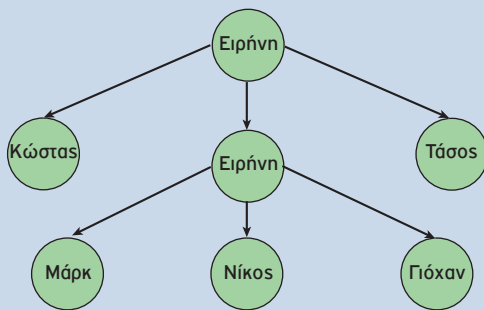
Όταν δύο κόμβοι συνδέονται μεταξύ τους με μία ακμή, τότε ονομάζουμε **«γονέα»** τον κόμβο από τον οποίο ξεκινάει η ακμή και **«παιδί»** τον κόμβο στον οποίο καταλήγει η ακμή. Στην Εικόνα 1.3.11 ο κόμβος p είναι γονέας του κόμβου c και ο κόμβος c είναι παιδί του κόμβου p . Ένας κόμβος μπορεί να έχει κανένα, 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



Οικογενειακές Σχέσεις



Εικόνα 1.3.12. Οικογενειακό δένδρο

Στο δένδρο της Εικόνας 1.3.12 βρείτε ποιος/ποιοι κόμβος/κόμβοι είναι:

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

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.

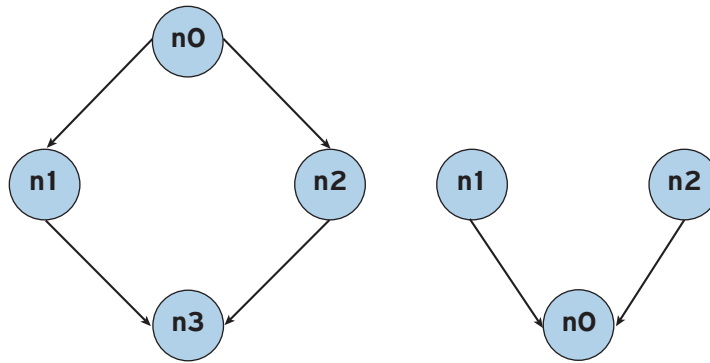


Ένα **δένδρο (tree)** είναι μία δομή που αποτελείται από ένα σύνολο κόμβων και ένα σύνολο ακμών μεταξύ των κόμβων με βάση τους εξής κανόνες:

- 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.

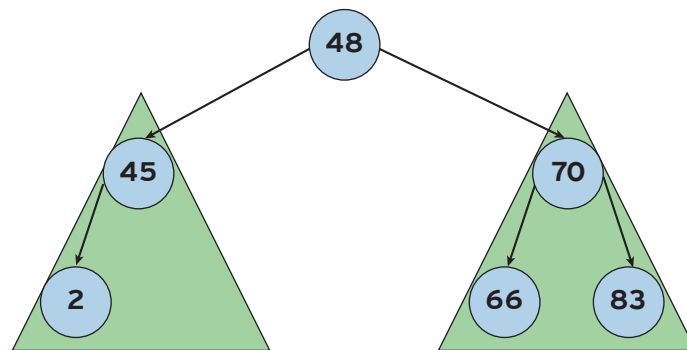
Δένδρο θεωρούμε και το κενό δένδρο, δηλαδή το δένδρο που δεν έχει ούτε κόμβους, ούτε ακμές. Το κενό δένδρο είναι το μόνο δένδρο χωρίς ρίζα.

Στην Εικόνα 1.3.13 παρουσιάζουμε παραδείγματα δομών που δεν είναι δένδρα. Η δομή στα αριστερά δεν είναι δένδρο επειδή ο κόμβος $n3$ έχει δύο γονείς, τους $n1$ και $n2$ 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 $n0$ προς τον κόμβο $n3$, πράγμα που και αυτό απαγορεύεται σε ένα δένδρο. Η δομή στα δεξιά δεν είναι δένδρο διότι υπάρχουν δύο κόμβοι χωρίς γονέα, οι $n1$ και $n2$. Είναι σαν να λέμε ότι έχουμε δύο ρίζες στο δένδρο αυτό. Γνωρίζουμε όμως ότι η ρίζα είναι μοναδική σε ένα δένδρο.



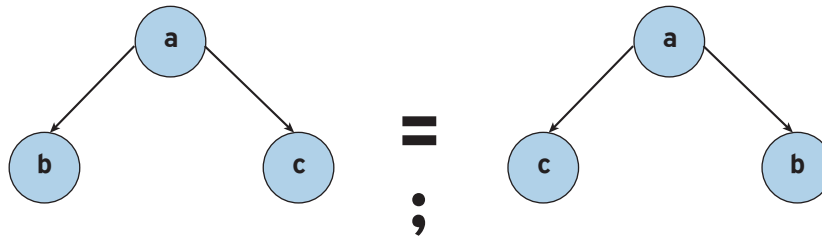
Εικόνα 1.3.13. Δομές που δεν είναι δένδρα

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 1.3.14, όπως φαίνεται και από το σχήμα, 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.



Εικόνα 1.3.14. Αριστερό και δεξιό υποδένδρο του κόμβου 48

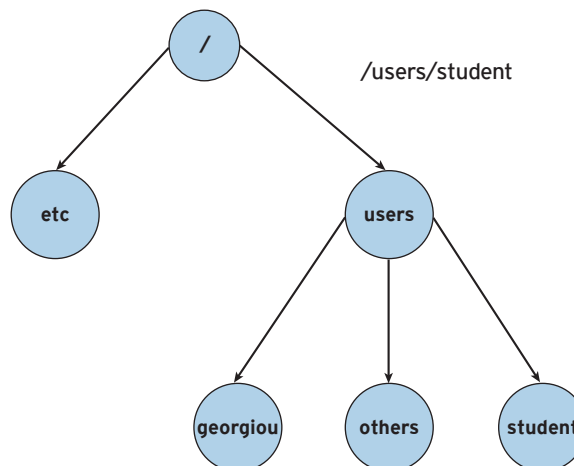
Τα δύο δένδρα της Εικόνας 1.3.15 είναι ίδια ή όχι; Μιλήσαμε για τη σχέση γονέα-παιδιού, αλλά τι γίνεται με τη σχέση μεταξύ των αδελφών; Έχει σημασία η σειρά των αδελφών b και c; Όχι πάντοτε. Για παράδειγμα, αν θέλουμε να μοντελοποιήσουμε την ιεραρχική σχέση των μελών μας οικογένειας και μας ενδιαφέρει 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.



Εικόνα 1.3.15. Πότε δύο δένδρα είναι ίδια

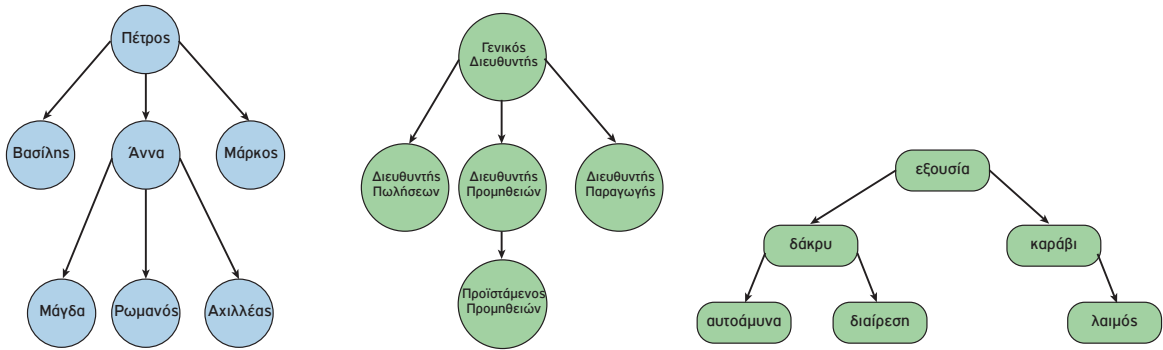
Τα δένδρα είναι μία μη-γραμμική ευέλικτη δομή δεδομένων που χρησιμοποιούνται σε πολλούς τομείς της επιστήμης των υπολογιστών, συμπεριλαμβανομένων των λειτουργικών συστημάτων, των γραφικών, των συστημάτων βάσεων δεδομένων, των παιχνιδιών, της τεχνητής νοημοσύνης και της δικτύωσης υπολογιστών.

Έχετε σκεφτεί γιατί υπάρχουν όμως τόσα 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, όπως θα δούμε στη συνέχεια. 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 1.3.16. 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".



Εικόνα 1.3.16. Μη-γραμμική ευέλικτη δομή δεδομένων

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, όπως είναι το οικογενειακό δένδρο (Εικόνα 1.3.17.α), η δομή ενός οργανισμού ή μιας εταιρείας (Εικόνα 1.3.17.β), ο πίνακας περιεχομένων ενός βιβλίου, τα αρχεία και οι φάκελοι ενός υπολογιστή, ένα λεξικό (Εικόνα 1.3.17.γ) (περισσότερα στην ενότητα των δυαδικών δένδρων αναζήτησης), τα μέρη που απαρτίζουν την μηχανή ενός αυτοκινήτου, τα συστατικά μιας πρότασης (Εικόνα 1.3.18) κ.ά.



Εικόνα 1.3.17.

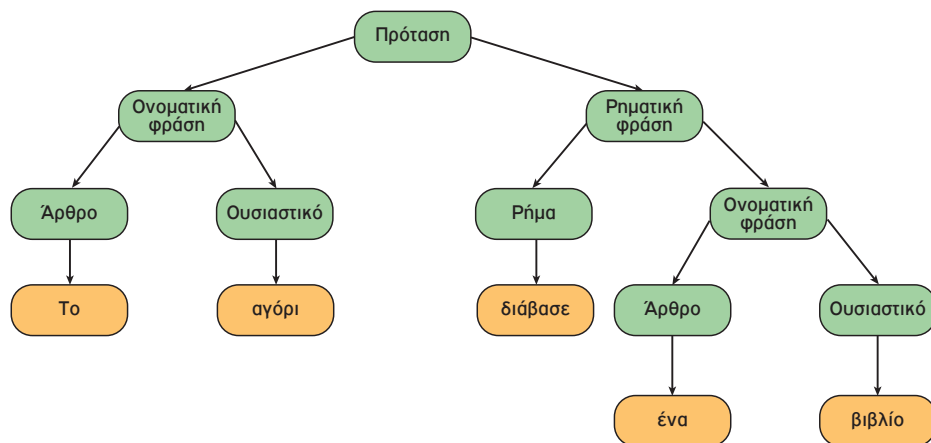
α. Οικογενειακό δένδρο

β. Οργανόγραμμα μιας εταιρείας

γ. Οργάνωση ενός λεξικού

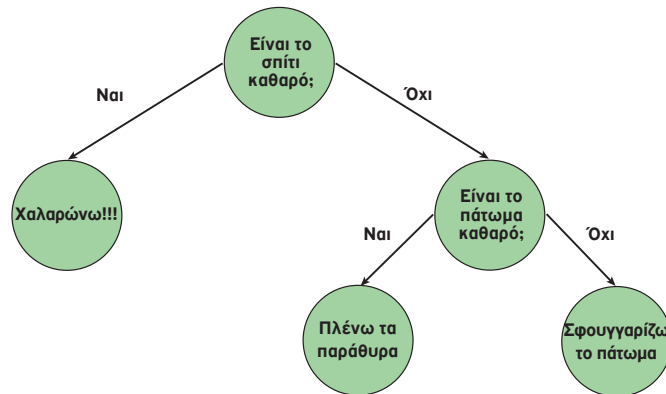
Έχετε σκεφτεί, όμως, ότι τα δένδρα, πέρα από μια αποτελεσματική οργάνωση και διαχείριση δεδομένων, αποτελούν τη βάση αρκετών αλγορίθμων επίλυσης προβλήματος, όπως είναι για παράδειγμα η συμπίεση εικόνων, η ταξινόμηση, η αυτόματη συμπλήρωση λέξεων σε συσκευές κινητών τηλεφώνων, η μεταγλώττιση ενός προγράμματος και η λήψη αποφάσεων;

Για παράδειγμα, ένα απλό πρόβλημα λήψης απόφασης μπορεί να είναι αυτό της καθαριότητας του φοιτητικού σπιτιού σας. Κάποια στιγμή θα κληθείτε να πάρετε αποφάσεις λαμβάνοντας υπόψη κάποιους παράγοντες. Μήπως θα σας βοηθούσε ένα δένδρο απόφασης παρόμοιο με αυτό που εμφανίζεται στην Εικόνα 1.3.19; Ειδικότερα, η μεθοδολογία των δένδρων αποφάσεων έχει γίνει ιδιαίτερα δημοφιλής σε περιπτώσεις ιατρικών διαγνώσεων.



Εικόνα 1.3.18. Η δομή μιας πρότασης

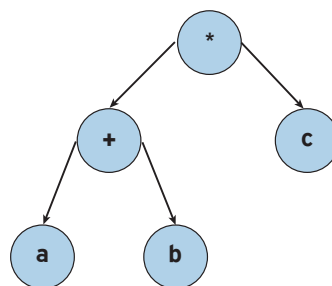
Τα δένδρα απόφασης, όπως πολύ εύκολα μπορείτε να συμπεράνετε από την Εικόνα 1.3.19, είναι δένδρα στα οποία κάθε κόμβος αντιπροσωπεύει ένα χαρακτηριστικό (ιδιότητα), κάθε ακμή αντιπροσωπεύει μια απόφαση (κανόνα) και κάθε φύλλο αντιπροσωπεύει ένα αποτέλεσμα. Στους αλγορίθμους μηχανικής μάθησης (machine learning) τα δένδρα απόφασης έχουν πρωτεύοντα ρόλο.



Εικόνα 1.3.19. Δένδρο Απόφασης

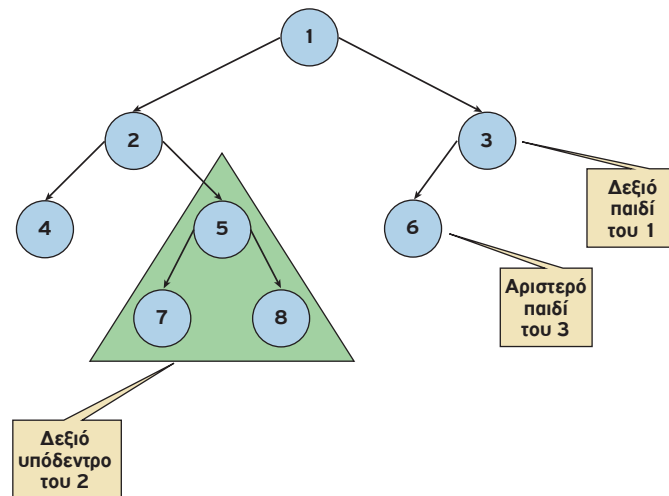
Από την άλλη πλευρά, γνωρίζετε ότι όταν παίζετε παιχνίδια στον υπολογιστή, όπως είναι το σκάκι, η τρίλιζα, το τάβλι και πολλά άλλα, ο υπολογιστής χρησιμοποιεί ένα ειδικό δένδρο, που ονομάζεται δένδρο του παιχνιδιού (game tree), το οποίο μοντελοποιεί όλες τις πιθανές κινήσεις των παικτών για να σας νικήσει; 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, όπως αυτό της Εικόνας 1.3.20.

Εικόνα 1.3.20. $(a+b)*c$

Διαδικά Δένδρα

Ένα διαδικό 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 διαδικό δένδρο της Εικόνας 1.3.21, 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.



Εικόνα 1.3.21. Διαδικό δένδρο



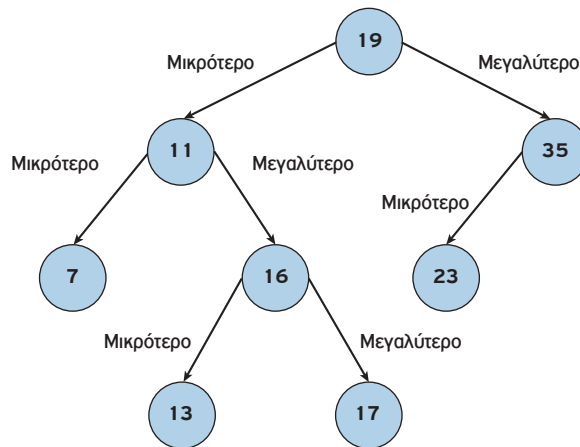
Στην Εικόνα 1.3.21 μπορείτε να βρείτε ποιοι κόμβοι έχουν ένα, δύο ή κανένα παιδί;

Διαδικά Δένδρα Αναζήτησης

Γνωρίζετε ότι 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 διαδικών δένδρων, αυτών των διαδικών δένδρων αναζήτησης;

Θα χρησιμοποιήσουμε τώρα τα διαδικά δένδρα με έναν συγκεκριμένο τρόπο. Θα αποθηκεύσουμε δεδομένα με έναν τρόπο που θα μας επιτρέψει να τα βρίσκουμε πιο εύκολα. Θα διαπιστώσετε και μόνοι σας στη συνέχεια ότι η ιδέα πίσω από ένα διαδικό δένδρο αναζήτησης είναι παρόμοια με αυτήν της διαδικής αναζήτησης σε έναν ταξινομημένο πίνακα.

Ένα διαδικό δένδρο αναζήτησης (binary search tree) είναι ένα διαδικό 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 . Στην Εικόνα 1.3.22 παρουσιάζεται το παράδειγμα ενός διαδικού δένδρου αναζήτησης.



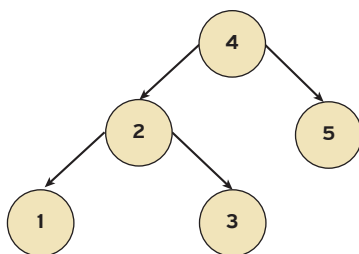
Εικόνα 1.3.22. Δυαδικό δένδρο αναζήτησης

Το δυαδικό δένδρο της Εικόνας 1.3.23.α πληροί τα κριτήρια του δυαδικού δένδρου αναζήτησης, σε αντίθεση με το δυαδικό δένδρο της Εικόνας 1.3.23.β.

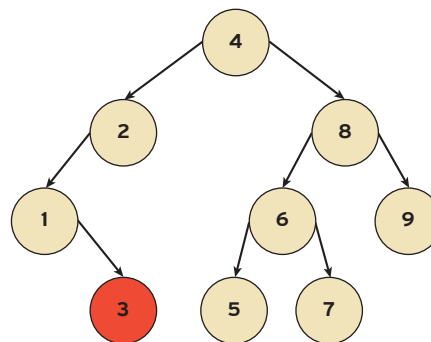
Στο δυαδικό δένδρο, της Εικόνας 1.3.23.β, παρατηρούμε τα εξής, όσον αφορά τον κόμβο 3:

- Ας θεωρήσουμε το υποδένδρο με ρίζα το 1. Ο κόμβος 3 βρίσκεται στο δεξιό υποδένδρο αυτού και είναι μεγαλύτερος από τον κόμβο/ρίζα 1.
- Ας θεωρήσουμε το υποδένδρο με ρίζα το 2. Ο κόμβος 3 βρίσκεται στο αριστερό υποδένδρο αυτού αλλά δεν είναι μικρότερος από τον κόμβο/ρίζα 2.

Εξαιτίας του τελευταίου γεγονότος, δηλαδή της ύπαρξης ενός τουλάχιστον κόμβου (στην περίπτωσή μας του κόμβου 3) που δεν πληροί το κριτήριο του δυαδικού δένδρου αναζήτησης, συμπεραίνουμε ότι δεν μπορούμε να κατατάξουμε το δυαδικό αυτό δένδρο στα δυαδικά δένδρα αναζήτησης.



Εικόνα 1.3.23. α: Δυαδικό δένδρο αναζήτησης



β: Μη δυαδικό δένδρο αναζήτησης

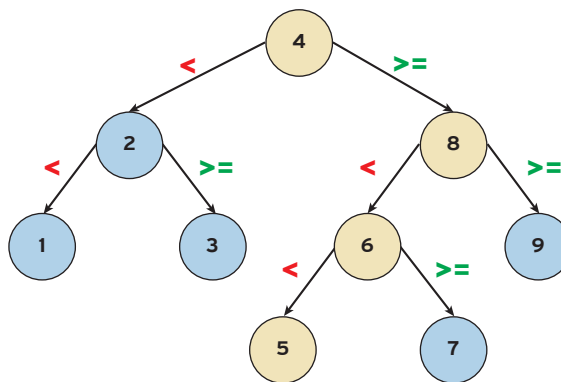


Στην Εικόνα 1.3.23.β, σε ποιο σημείο του δένδρου θα μπορούσε να τοποθετηθεί ο κόμβος 3, έτσι ώστε το δένδρο αυτό να είναι ένα δυαδικό δένδρο αναζήτησης;

Ποιο νομίζετε ότι είναι το πλεονέκτημα των δυαδικών δένδρων αναζήτησης; Το πλεονέκτημα βρίσκεται στο ίδιο το όνομα και συγκεκριμένα στη λέξη «αναζήτηση». Η αναζήτηση για μια συγκεκριμένη τιμή γίνεται ταχύτερα χάρη στον τρόπο αποθήκευσης των τιμών. Για παράδειγμα, ποια είναι η χειρότερη περίπτωση που μπορούμε να συναντήσουμε όταν ψάχνουμε μία τιμή σε μια γραμμική δομή, όπως είναι ο πίνακας; Θεωρείστε ότι ψάχνετε το στοιχείο 5 στον πίνακα της Εικόνας 1.3.24. Στην περίπτωση αυτή, θα χρειαζόταν να διασχίσουμε ολόκληρο τον πίνακα για να βρούμε το στοιχείο 5. Τώρα, θεωρείστε το δένδρο της Εικόνας 1.3.24. Μετά από πόσες συγκρίσεις θα βρείτε το στοιχείο 5;

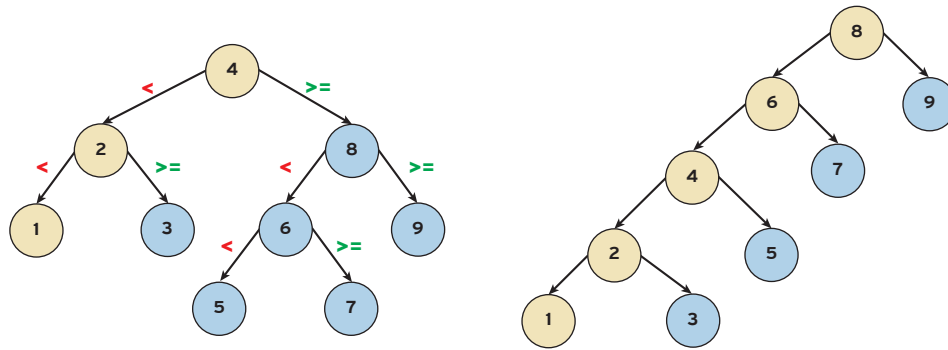
Ας δούμε τον αλγόριθμο της αναζήτησης σε ένα δυαδικό δένδρο αναζήτησης μέσα από μερικά παραδείγματα. Αρχικά, συγκρίνετε τη ρίζα του δένδρου (δηλαδή το 4) με το υπό αναζήτηση στοιχείο (δηλαδή το 5). Ο αριθμός που αναζητάτε είναι μεγαλύτερος από τη ρίζα. Συνεπώς συνεχίζετε με το δεξιό υποδένδρο και αγνοείτε το αριστερό υποδένδρο. Στη συνέχεια, συγκρίνετε το 8 με το 5. Επειδή το 8 είναι μεγαλύτερο από το 5, συνεχίζετε με το αριστερό υποδένδρο. Κατόπιν, συναντάτε το 6 που είναι και αυτό μεγαλύτερο από το 5. Μεταβαίνετε, λοιπόν, στο αριστερό υποδένδρο του 6, όπου βρίσκετε το στοιχείο 5 που αναζητάτε. Στην περίπτωση που δεν υπάρχει το στοιχείο στο δένδρο (για παράδειγμα αν ψάχναμε το στοιχείο 12), η αναζήτηση καταλήγει σε ένα κενό υποδένδρο και εκεί τερματίζει η όλη διαδικασία. Με βάση τον αλγόριθμο αυτό, μειώνουμε δραματικά τον χρόνο για να βρούμε το στοιχείο που ψάχνουμε και αυτό διότι περιορίζουμε αισθητά τους κόμβους τους οποίους επισκεπτόμαστε. Κάθε φορά αφήνουμε στην άκρη ένα υποδένδρο και συνεχίζουμε με το άλλο.

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



Εικόνα 1.3.24. Αλγόριθμος αναζήτησης σε ένα δυαδικό δένδρο αναζήτησης

Θεωρείστε ότι αναζητούμε το στοιχείο 1 στα δυαδικά δένδρα αναζήτησης της Εικόνας 1.3.25. Σε ποιο από τα δύο δυαδικά δένδρα αναζήτησης θα βρούμε πιο γρήγορα την τιμή 1; Μήπως στο πρώτο, διότι σε κάθε βήμα απορρίπτουμε όσο περισσότερους κόμβους είναι δυνατόν; Οπτικά, θα μπορούσαμε να θεωρήσουμε ότι το πρώτο δένδρο είναι πιο «ισορροπημένο» σε σχέση με το δεύτερο; Παρατηρούμε, λοιπόν, χωρίς να αναφερθούμε διεξοδικά στα ισορροπημένα δένδρα, ότι αν θέλουμε να έχουμε γρήγορους αλγόριθμους αναζήτησης πρέπει να αποθηκεύουμε τις τιμές στα δυαδικά δένδρα αναζήτησης με έναν συγκεκριμένο τρόπο.

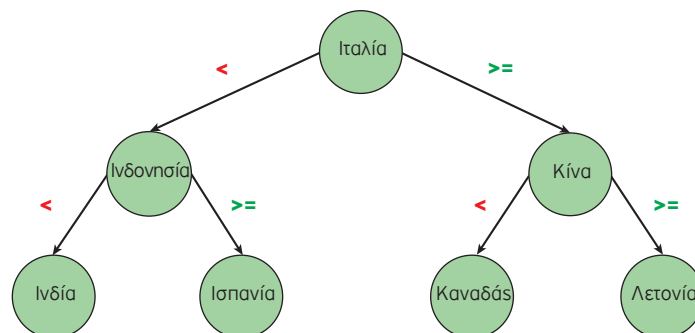


Εικόνα 1.3.25. Η σημασία της δομής ενός δυαδικού δένδρου αναζήτησης στην εύρεση στοιχείων

Σας θυμίζει ο αλγόριθμος αυτός τον αλγόριθμο της δυαδικής αναζήτησης που έχετε συναντήσει στους ταξινομημένους πίνακες; Ο αλγόριθμος της δυαδικής αναζήτησης «τρέχει» γρήγορα στους ταξινομημένους πίνακες σε αντίθεση με τους αλγορίθμους της εισαγωγής και της διαγραφής στοιχείου που είναι πιο χρονοβόροι και αυτό διότι πρέπει να διατηρηθεί η υπάρχουσα διάταξη των στοιχείων. Ας μην ξεχνάμε επίσης ότι οι πίνακες είναι στατικές δομές και επομένως, δεν έχουν δυνατότητα αυξομείωσης του μεγέθους τους. Αυτό σημαίνει ότι κάθε επόμενη εισαγωγή στοιχείου θα οδηγήσει ενδεχομένως σε αντιγραφή όλων των στοιχείων του πίνακα σε έναν μεγαλύτερο πίνακα.

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. Φανταστείτε ότι μετατρέπετε έναν ταξινομημένο πίνακα σε ένα δυαδικό δένδρο, όπως παρουσιάζεται στην Εικόνα 1.3.26. Αυτό μας δίνει τη δυνατότητα να αναζητήσουμε ένα στοιχείο το ίδιο γρήγορα όσο και σε έναν ταξινομημένο πίνακα αλλά και να εισαγάγουμε και να διαγράψουμε εύκολα ένα στοιχείο ακριβώς επειδή δουλεύουμε με δένδρα και όχι με πίνακες.

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



Εικόνα 1.3.26. Αναζήτηση σε ταξινομημένο πίνακα και σε δυαδικό δένδρο αναζήτησης

1.3.3 Γράφοι

Τώρα που μελετήσαμε τις δενδρικές δομές δεδομένων, προκύπτει το ερώτημα «από πού προέρχονται τα δένδρα;». Τα δένδρα είναι στην πραγματικότητα ένα υποσύνολο των γράφων. Αλλά προκειμένου να καταλάβετε πραγματικά γιατί χρησιμοποιούμε γράφους και τι είναι αυτοί, θα πρέπει να αναφερθούμε στη θεωρία γράφων.

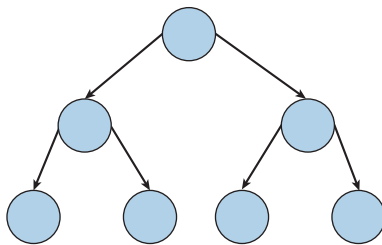
Είδαμε ότι το πιο θεμελιώδες χαρακτηριστικό των μη γραμμικών δομών είναι ότι τα δεδομένα τους δεν ακολουθούν μια σειρά - όπως στους πίνακες ή τις συνδεδεμένες λίστες. Τα δένδρα, όπως είδαμε, ξεκινούν με έναν κόμβο ρίζας και μπορεί να συνδεθούν με άλλους κόμβους, κάτι που σημαίνει ότι θα μπορούσαν να περιέχουν δευτερεύοντα δένδρα στο εσωτερικό τους. 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.

Αλλά τι θα γίνει αν αγνοήσουμε αυτούς τους κανόνες; Τότε, δεν αναφερόμαστε σε δένδρα αλλά σε μία νέα δυναμική δομή δεδομένων, που ονομάζεται γράφος. Τα δένδρα δεν είναι παρά περιορισμένοι τύποι γράφων. Ένα δένδρο θα είναι πάντα ένα γράφος, αλλά δεν είναι όλοι οι γράφοι δένδρα.

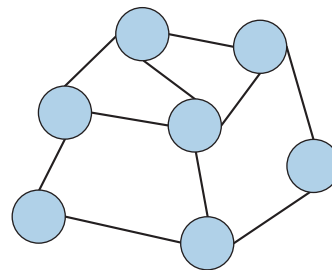
Τι είναι αυτό που κάνει ένα δένδρο διαφορετικό από έναν γράφο;

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 (Εικόνα 1.3.27.α).

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 (Εικόνα 1.3.27.β)! Οι γράφοι, επίσης, δεν έχουν «μονοκατευθυντική» ροή - αντ' αυτού, μπορεί να έχουν κατεύθυνση ή να μην έχουν καμιά κατεύθυνση.



α: Δένδρο



β: Γράφος

Εικόνα 1.3.27.



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

As εξετάσουμε τους δύο τύπους γράφων που είναι αρκετά εύκολο να εντοπιστούν και είναι αρκετά συνηθισμένοι στα προβλήματα θεωρίας γράφων: οι **κατευθυνόμενοι** γράφοι και οι **μη κατευθυνόμενοι** γράφοι.

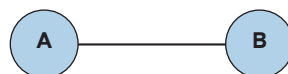
Όπως έχουμε αναφέρει, σε έναν γράφο δεν υπάρχουν πραγματικοί κανόνες για τον τρόπο που ένας κόμβος συνδέεται με έναν άλλο κόμβο. Οι ακμές μπορούν να συνδέσουν τους κόμβους με οποιονδήποτε τρόπο. Οι διαφορετικοί τύποι ακμών είναι πολύ σημαντικοί όταν πρόκειται για την αναγνώριση και τον καθορισμό του τύπου των γράφων. Ως επί το πλείστον, οι γράφοι μπορούν να έχουν δύο τύπους ακμών: ακμές που έχουν κατεύθυνση (κατευθυνόμενες ακμές) και ακμές που δεν έχουν κατεύθυνση (μη κατευθυνόμενες ακμές). Σε μια κατευθυνόμενη ακμή, δύο κόμβοι συνδέονται με πολύ συγκεκριμένο τρόπο.

Στην εικόνα 1.3.28.α, όπου ο κόμβος A συνδέεται με τον κόμβο B, υπάρχει ένας μόνο τρόπος να ταξιδέψουμε μεταξύ αυτών των δύο κόμβων και αυτός είναι από τον κόμβο A στον κόμβο B. Είναι πολύ συνηθισμένο να αναφέρουμε τον κόμβο από τον οποίο ξεκινάμε ως προέλευση και τον κόμβο στον οποίο ταξιδεύουμε ως προορισμό. Σε μια κατευθυνόμενη ακμή, μπορούμε να ταξιδέψουμε μόνο από την προέλευση στον προορισμό, και ποτέ το αντίστροφο (Εικόνα 1.3.28.α). Ωστόσο, σε μια μη κατευθυνόμενη ακμή, η διαδρομή μεταξύ των δύο κόμβων είναι αμφίδρομη, πράγμα που σημαίνει ότι οι κόμβοι προέλευσης και προορισμού δεν είναι σταθεροί.

Αυτή η διαφοροποίηση είναι πραγματικά πολύ σημαντική, επειδή οι ακμές σε έναν γράφο καθορίζουν το πώς αποκαλείται ο γράφος.



α: Κατευθυνόμενη ακμή



β: μη κατευθυνόμενη ακμή

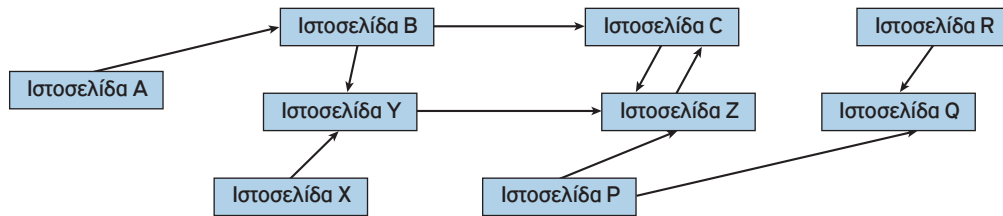
Εικόνα 1.3.28.



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)**.

Οι γράφοι είναι γύρω μας, απλά δεν τους βλέπουμε. Ο Παγκόσμιος Ιστός (WWW) είναι ένας τεράστιος γράφος (Εικόνα 1.3.29)! 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



Κοινωνικά Δίκτυα - Facebook

Υπάρχει, όμως, ένα ακόμη καλύτερο παράδειγμα που απεικονίζει τις καθημερινές μας αλληλεπιδράσεις και η υλοποίησή του στηρίζεται στη δομή δεδομένων «γράφος»: τα κοινωνικά δίκτυα.

Το Facebook, ένα τεράστιο κοινωνικό δίκτυο, είναι ένας τύπος γράφου. Και αν σκεφτούμε περισσότερο το πώς λειτουργεί πραγματικά, αρχίζουμε να κατανοούμε καλύτερα τον τρόπο διασυνδέσεων των κόμβων καθώς και με ποιον τύπο γράφου αναπαριστάνεται. 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. Ποιον τύπο γράφου θα χρησιμοποιήσουμε για την αναπαράσταση του κοινωνικού δικτύου Facebook;



Κοινωνικά Δίκτυα - Twitter

A) Το Twitter, από την άλλη πλευρά, λειτουργεί πολύ διαφορετικά από το Facebook. Μπορώ να σε ακολουθήσω, αλλά δεν απαιτείται να με ακολουθήσεις. Με ποιον τύπο γράφου μπορούμε να αναπαραστήσουμε το Twitter;

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;

B) Τι θα συμβεί εάν αποφασίσω να σε ακολουθήσω και εγώ; Αλλάζω την κατεύθυνση της ακμής που δημιουργήθηκε όταν με ακολουθούσες; Ξαφνικά η ακμή γίνεται αμφίδρομη και επομένως μη κατευθυνόμενη;



Επτά γέφυρες του Königsberg

Υπάρχει μια ενδιαφέρουσα ιστορία πίσω από την προέλευσή του προβλήματος «οι Επτά γέφυρες του Königsberg». Η πρόκληση (ή απλά σπαζοκεφαλιά) με τις γέφυρες του Königsberg, ήταν να μπορεί κάποιος να διασχίσει και τις επτά γέφυρες μόνο μία φορά περπατώντας μέσα στην πόλη. Κοιτάξτε πρώτα την Εικόνα 1.3.30 για να αντιληφθείτε το πρόβλημα:



Εικόνα 1.3.30. Οι επτά γέφυρες του Königsberg

Προσπαθήστε και δείτε αν μπορείτε να περπατήσετε μέσα από την πόλη και να διασχίσετε όλες τις γέφυρες μόνο μία φορά. Θα πρέπει να λάβετε υπόψη δύο πράγματα ενώ προσπαθείτε να λύσετε το παραπάνω πρόβλημα-αίνιγμα:

- πρέπει να διασχίσετε όλες τις γέφυρες
- δεν πρέπει να διασχίσετε περισσότερο από μία φορά κάποια γέφυρα

Μπορείτε να δοκιμάσετε οποιονδήποτε αριθμό συνδυασμών θέλετε, αλλά θα δείτε ότι είναι αδύνατον. Δεν υπάρχει κανένας τρόπος με τον οποίο κάποιος μπορεί να περπατήσει μέσα στην πόλη και να διασχίσει κάθε γέφυρα μόνο μία φορά. Ο Leonhard Euler αναπαράστησε αυτό το πρόβλημα σε γράφο, όπου αναπαράστησε τα τμήματα γης (A,B,C,D) με κορυφές και τις γέφυρες με ακμές.

Αναζητήστε στο Διαδίκτυο πως ο Euler αναπαράστησε το πρόβλημα με γράφο και για ποιο λόγο είναι αδύνατο να λυθεί αυτό το πρόβλημα.

1.3.4 Ερωτήσεις - Ασκήσεις

Ε.1: Δώστε δύο παραδείγματα εφαρμογών από την καθημερινή ζωή

- απλά συνδεδεμένης λίστας
- διπλά συνδεδεμένης λίστας

Ε.2: Βρείτε το σωστό.

Χαρακτηρίστε τις παρακάτω προτάσεις ως Σωστές ή Λάθος. Στην περίπτωση που πιστεύετε ότι είναι λανθασμένες δικαιολογήστε την επιλογή σας.

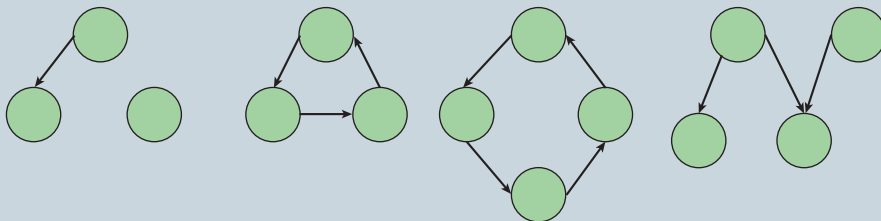
α/α	Προτάσεις	Σ	Λ
1	Μια απλά συνδεδεμένη λίστα μπορούμε να τη διατρέξουμε και προς τις δύο κατευθύνσεις.	☐	☐
2	Σε μία λίστα δε χρειάζεται να οριστεί ένα αρχικό μέγεθος.	☐	☐
3	Δεν είναι δυνατό να υπάρχει «τυχαία» πρόσβαση σε μια απλά συνδεδεμένη λίστα.	☐	☐
4	Σε μια λίστα, τα στοιχεία δεν μπορούν να προστεθούν 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 προσπελαύνονται και από τις δύο κατευθύνσεις.	☐	☐

Ε.3: Σχεδιάστε γεωγραφικές ιεραρχίες

Σχεδιάστε ένα δένδρο που θα αποτυπώνει την ιεραρχία μιας γεωγραφικής περιοχής σε επίπεδο χωρών, νομών και πόλεων.

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

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

Ε.6: Εγκυκλοπαίδεια

Οι πληροφορίες σε μια εγκυκλοπαίδεια μπορούν να οργανωθούν σε ένα σύνολο θεματικών κατηγοριών, όπως Θετικές Επιστήμες, Ιστορία, Τέχνες κ.λπ. Κάθε θεματική κατηγορία μπορεί με την σειρά της να υποδιαιρείται σε μια άλλη σειρά θεματικών υποκατηγοριών σε διάφορα επίπεδα και υποεπίπεδα με τα άρθρα να βρίσκονται στη βάση της δομής. Για παράδειγμα, η κατηγορία Θετικές Επιστήμες μπορεί να περιλαμβάνει τις υποκατηγορίες Μαθηματικά, Φυσική, Χημεία, Πληροφορική κ.λπ.

Επιλέξτε ποια από τις παρακάτω δομές δεδομένων είναι καταλληλότερη για την αναπαράσταση αυτών των πληροφοριών:

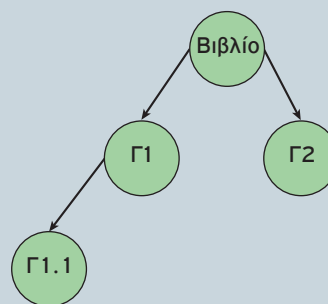
i) δένδρο **ii)** λίστα **iii)** πίνακας

Ε.7: Περιεχόμενα βιβλίου

Στην Εικόνα 1.3.32 παρουσιάζεται αριστερά η δομή των περιεχομένων ενός βιβλίου και δεξιά ένα ημιτελές δένδρο, το οποίο αναπαριστά εν μέρει την δομή του βιβλίου αυτού. Προσπαθήστε να συμπληρώσετε το δένδρο αυτό, ώστε να απεικονίζει την δομή του βιβλίου στο σύνολό της.

Βιβλίο

Γ1
 Γ1.1
 Γ1.2
 Γ2
 Γ2.1
 Γ2.1.1
 Γ2.1.2
 Γ2.2
 Γ2.3
 Γ3

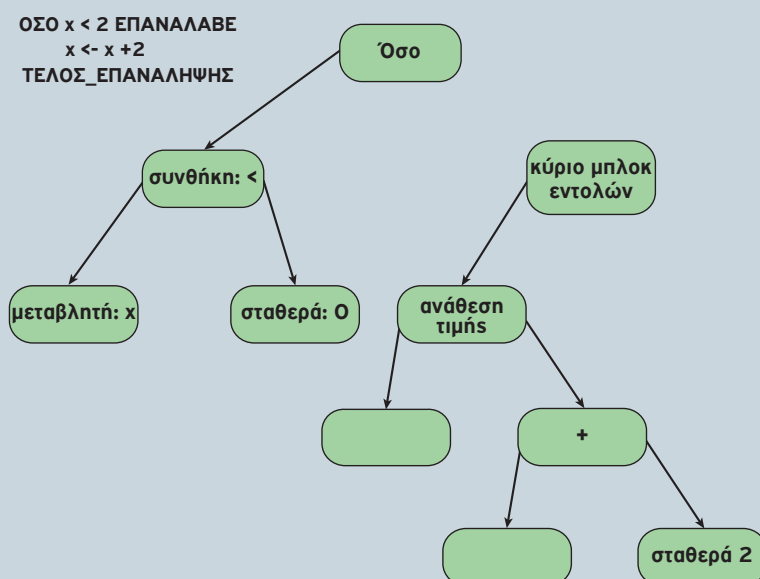


Εικόνα 1.3.32. Περιεχόμενα βιβλίου

Διαγραμματική αναπαράσταση

Ε.8: Δένδρο για την αναπαράσταση κώδικα

Προσπαθήστε να έρθετε στη θέση του μεταγλωττιστή του υπολογιστή σας και να μετατρέψετε το πρόγραμμα της Εικόνας 1.3.33 στο αντίστοιχο δένδρο. Εμείς το κάναμε για εσάς αλλά παραλείψαμε να δώσουμε περιεχόμενο σε κάποιους κόμβους. Μπορείτε να συμπληρώσετε το περιεχόμενο εσείς;



Εικόνα 1.3.33. Δομή τμήματος προγράμματος

Ε.9: Βρείτε το σωστό.

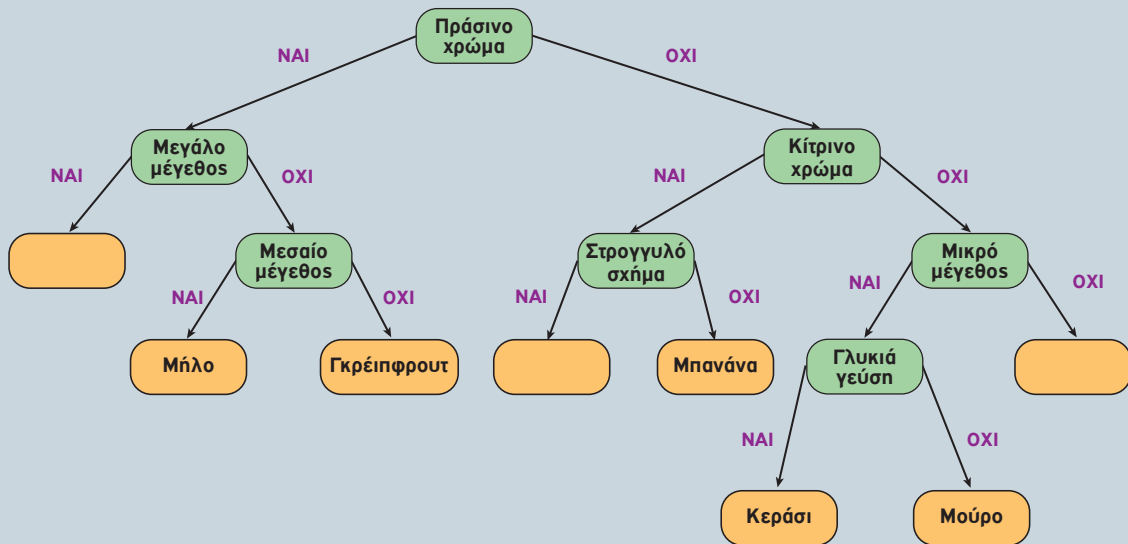
Χαρακτηρίστε τις παρακάτω προτάσεις ως Σωστές ή Λάθος. Στην περίπτωση που πιστεύετε ότι είναι λανθασμένες δικαιολογήστε την επιλογή σας.

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	☐	☐

Ε.10: Δένδρο απόφασης

Το δένδρο απόφασης της Εικόνας 1.3.34 αποτελεί μια προσπάθεια κατηγοριοποίησης μιας σειράς φρούτων, όπως το καρπούζι, το μήλο, το γκρέιπφρουτ, το λεμόνι, η μπανάνα, το κεράσι, το μούρο και το πορτοκάλι, με βάση τα χαρακτηριστικά τους, όπως το χρώμα, το μέγεθος, το σχήμα και η γεύση.

Κάποια φρούτα, όμως, όπως το καρπούζι, το πορτοκάλι και το λεμόνι δεν έχουν κατηγοριοποιηθεί ακόμα. Προσπαθήστε να τα ταξινομήσετε ονοματίζοντας τα κενά φύλλα με τις λέξεις καρπούζι, πορτοκάλι και λεμόνι.

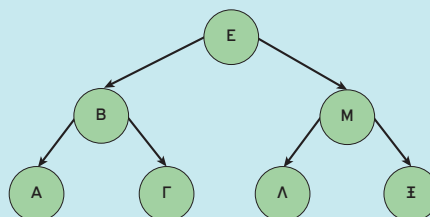


Εικόνα 1.3.34. Κατηγοριοποίηση φρούτων

Ε.12: Δεξιά ή αριστερά;

Στο δυαδικό δένδρο αναζήτησης της Εικόνας 1.3.35, αφού συγκρίνετε το στοιχείο Π (το στοιχείο που προσπαθείτε να βρείτε) με το στοιχείο Ε (τη ρίζα) και βρείτε ότι δεν είναι ίσα, σε ποια κατεύθυνση θα το αναζητήσετε στη συνέχεια;

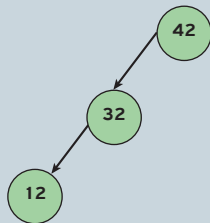
- ☐ Στα αριστερά του Ε
- ☐ Στα δεξιά του Ε
- ☐ Δεν μπορείτε να πείτε και πρέπει να δοκιμάσετε και προς τις δύο κατευθύνσεις.



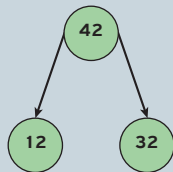
Εικόνα 1.3.35. Δυαδική αναζήτηση

Ε.11: Χαμένοι στο δάσος των δυαδικών δένδρων

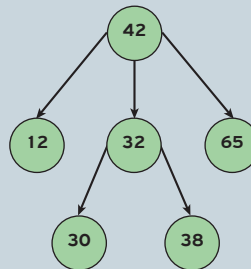
Ποια από τα παρακάτω δένδρα είναι δυαδικά δένδρα αναζήτησης. Εξηγήστε το γιατί:



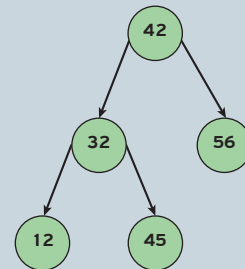
α. _____



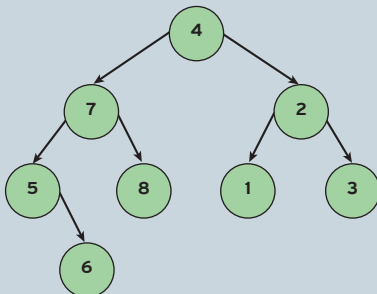
β. _____



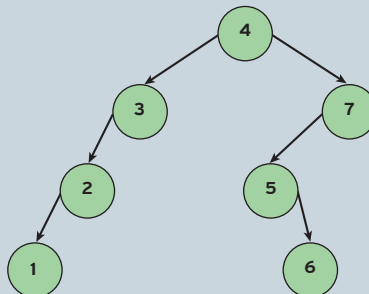
γ. _____



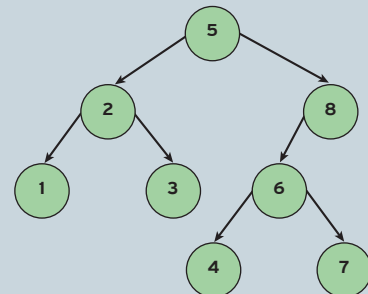
δ. _____



α. _____



β. _____



γ. _____

Ε.13: Τηλεφωνικό Κέντρο Ταξί

Ας υποθέσουμε ότι έχετε τηλεφωνήσει για να έρθει να σας πάρει ένα ταξί. Ένα από τα πιο σημαντικά πράγματα, που είναι κρίσιμα για τη λειτουργία του κέντρου που καλείτε, είναι η ικανότητά του να συνδυάζει τους οδηγούς με τους επιβάτες με αποτελεσματικό τρόπο. Σκεφτείτε ότι υπάρχουν 6 πιθανά ταξί τα οποία μπορούν να έρθουν εγκαίρως να σας παραλάβουν. Πώς θα επιλέξει το κέντρο να σας παραχωρήσει ένα ταξί; Μπορούμε να χρησιμοποιήσουμε γράφο για να δούμε πώς μπορεί να πραγματοποιηθεί η διαδικασία επιλογής του κατάλληλου ταξί. Το πρώτο ταξί (Α) απέχει 12, το δεύτερο (Β) 5, το τρίτο (Γ) 10, το τέταρτο (Δ) 7, το πέμπτο (Ε) 13 και το έκτο (ΣΤ) 20 χιλιόμετρα.

- i) Να σχεδιάσετε τον γράφο που αναπαριστά το πρόβλημα (πάνω στις ακμές γράψτε τα χιλιόμετρα).
- ii) Ποιον τύπο γράφου χρησιμοποιήσατε;
- iii) Ποιο ταξί θα επιλέξει το κέντρο; Θεωρήστε ότι το μόνο κριτήριο για την επιλογή είναι η χιλιομετρική απόσταση.