ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ, ΕΡΕΥΝΑΣ
ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ

ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ
ΠΟΛΙΤΙΚΗΣ

Εισαγωγή στις Αρχές της Επιστήμης των Η/Υ

Β΄ Λυκείου

ΤΟΜΟΣ 1ος

Αθήνα 2014

ΙΝΣΤΙΤΟΥΤΟ ΤΕΧΝΟΛΟΓΙΑΣ
ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΕΚΔΟΣΕΩΝ
«ΔΙΟΦΑΝΤΟΣ»

ΙΝΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ
Πρόεδρος: **Σωτήριος Γκλαβάς**

ΓΡΑΦΕΙΟ ΕΡΕΥΝΑΣ ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΕΦΑΡΜΟΓΩΝ Β΄
Προϊστάμενος:
Παύλος Φ. Μάραντος

ΣΥΓΓΡΑΦΕΙΣ:
Δρ. **Σπυρίδων Δουκάκης**, Πληροφορικός, Μαθηματικός, PIERCE-Αμερικανικό Κολλέγιο Ελλάδος
Χρήστος Δουληγέρης, Καθηγητής Τμήματος Πληροφορικής Πανεπιστημίου Πειραιώς
Δρ. **Θεόδωρος Καρβουνίδης**, Εκπαιδευτικός ΠΕ19
Χρήστος Κοίλιας, Καθηγητής Τμήματος Μηχανικών Πληροφορικής Τ.Ε. ΤΕΙ Αθήνας

Δρ. Αθανάσιος Πέρδος, Πληροφορικός, Φυσικός, Ελληνογαλλική Σχολή Καλαμαρί

ΣΥΝΤΟΝΙΣΤΗΣ: Χρήστος Κοίλιας

ΣΥΛΛΟΓΗ - ΕΠΕΞΕΡΓΑΣΙΑ ΥΛΙΚΟΥ:
Δημήτριος Κοτσιφάκος, Εκπαιδευτικός, ΠΕ 1708

ΚΡΙΤΕΣ-ΑΞΙΟΛΟΓΗΤΕΣ:
Παναγιώτης Βαρζάκας, Μέλος ΔΕΠ (συντονιστής)
Σοφία Τζελέπη, Σχολική Σύμβουλος, ΠΕ19
Πέτρος Ματζάκος, Εκπαιδευτικός, ΠΕ19

ΦΙΛΟΛΟΓΙΚΗ ΕΠΙΜΕΛΕΙΑ:
Μαρία Κοίλια

ΕΞΩΦΥΛΛΟ: Γιώργος Σκούφος

ΣΕΛΙΔΟΠΟΙΗΣΗ: Γιώργος Σκούφος

ΑΝΑΔΟΧΟΣ: ΕΚΔΟΣΕΙΣ ΝΕΩΝ ΤΕ-ΧΝΟΛΟΓΙΩΝ

NewTech Pub.

Στουρνάρη 49Α, 106 82,
Αθήνα, Τηλ. 210-38.45.594
Fax: 210-38.08.009
E-mail:
contact@newtech-publications.gr
URL:
www.newtech-pub.com

«ΔΗΜΙΟΥΡΓΙΑ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΥΛΙΚΟΥ ΓΙΑ ΤΑ ΝΕΑ ΜΑΘΗΜΑΤΑ ΤΟΥ ΓΕΝΙΚΟΥ ΛΥΚΕΙΟΥ» της Πράξης «ΝΕΟ ΣΧΟΛΕΙΟ (ΣΧΟΛΕΙΟ 21ου αιώνα) - ΝΕΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ» ΜΕ ΚΩΔ. ΟΠΣ 295450, των Αξόνων Προτεραιότητας 1, 2 και 3 - ΟΡΙΖΟΝΤΙΑ ΠΡΑΞΗ του ΕΠΙΧΕΙΡΗΣΙΑΚΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ «ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ», που συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση - Ευρωπαϊκό Κοινωνικό Ταμείο και από Εθνικούς Πόρους (ΕΣΠΑ 2007 - 2013).

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Η αξιολόγηση, η κρίση των προσαρμογών και η επιστημονική επιμέλεια του προσαρμοσμένου βιβλίου πραγματοποιείται από τη Μονάδα Ειδικής Αγωγής του Ινστιτούτου Εκπαιδευτικής Πολιτικής.

Η προσαρμογή του βιβλίου για μαθητές με μειωμένη όραση από το ΙΤΥΕ – ΔΙΟΦΑΝΤΟΣ πραγματοποιείται με βάση τις προδιαγραφές που έχουν αναπτυχθεί από ειδικούς εμπειρογνώμονες για το ΙΕΠ.

ΠΡΟΣΑΡΜΟΓΗ ΤΟΥ ΒΙΒΛΙΟΥ ΓΙΑ ΜΑΘΗΤΕΣ ΜΕ ΜΕΙΩΜΕΝΗ ΟΡΑΣΗ

ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ

Πρόλογος

Το παρόν σύγγραμμα έρχεται να υπηρετήσει τη διδασκαλία του μαθήματος «Εισαγωγή στις Αρχές της Επιστήμης των Η/Υ», προσεγγίζοντας θέματα τόσο της Θεωρητικής όσο και της Εφαρμοσμένης Επιστήμης των Η/Υ. Το πρώτο μέρος καλύπτει θέματα της Θεωρητικής Επιστήμης των Υπολογιστών –από το Πρόβλημα στον Αλγόριθμο και από εκεί στον Προγραμματισμό και τις Εφαρμογές του– μέσω του οποίου επιδιώκεται η ανάπτυξη της αναλυτικής και συνθετικής σκέψης των μαθητών και των μαθητριών. Στο δεύτερο μέρος, το βιβλίο πραγματεύεται ζητήματα των βασικών τομέων της Εφαρμοσμένης Επιστήμης των Υπολογιστών, ώστε να βοηθήσει τους μαθητές και τις μαθήτριες να είναι σε θέση να κατονομάζουν και να εξηγούν επιστημονικές περιοχές της Πληροφορικής.

Αναλυτικότερα, στην πρώτη ενότητα, στο **Κεφάλαιο 1.1** επιχειρείται η διάκριση της Επιστήμης των Η/Υ σε Θεωρητική και Εφαρμοσμένη. Η παραπάνω διάκριση διαχωρίζει το σύγγραμμα σε δύο ακόμα ενότητες. Στη δεύτερη ενότητα και στο **Κεφάλαιο 2.1** επαναπροσεγγίζεται η έννοια του προβλήματος και οι φάσεις επίλυσής του και αναδεικνύονται τα υπολογιστικά προβλήματα. Στη συνέχεια, στο **Κεφάλαιο 2.2** περιγράφεται η έννοια του αλγορίθμου και των χαρακτηριστικών του,

προσδιορίζονται οι διάφορες μορφές αναπαράστασής του και επεξηγούνται οι βασικές έννοιες στην Ανάλυση Αλγορίθμων. Ακολούθως, αναφέρονται οι βασικοί τύποι και οι δομές δεδομένων καθώς και οι βασικές εντολές και δομές που χρησιμοποιούνται σε έναν αλγόριθμο. Τέλος, στο κεφάλαιο αυτό, προσδιορίζεται ο τρόπος εντοπισμού και διόρθωσης λογικών λαθών σε έναν αλγόριθμο. Όλα τα παραπάνω βήματα, αποτελούν τα στάδια προετοιμασίας των μαθητών και μαθητριών, ώστε στο Κεφάλαιο 2.3 να προχωρήσουν στη δημιουργία ενός γνωσιακού και νοητικού σχήματος που να περιλαμβάνει τα είδη και τις τεχνικές προγραμματισμού και συγχρόνως να είναι σε θέση να συνδυάζουν τις

αλγοριθμικές δομές, τα δεδομένα και τις δομές δεδομένων για να δημιουργήσουν προγράμματα. Τέλος, στην ενότητα αυτή αναδεικνύεται ότι οι σημερινές εφαρμογές είναι αρκετά πολύπλοκες και η δημιουργία τους ακολουθεί συγκεκριμένα μοντέλα ανάπτυξης εφαρμογών λογισμικού που εξελίσσονται σε φάσεις.

Η τρίτη ενότητα ασχολείται με θέματα της Εφαρμοσμένης Επιστήμης των Υπολογιστών, η οποία εστιάζει τόσο στην επίλυση προβλημάτων, όσο και στη βελτίωση υπαρχουσών λύσεων στον πραγματικό κόσμο. Σε όλους τους τομείς της καθημερινότητας υπάρχει ένας τεράστιος όγκος διαθέσιμης πληροφορίας και γνώσεων, η διαχείριση του οποίου προϋποθέτει τις κατάλληλες υπολογιστικές υποδομές για την αποθήκευση της πληροφορίας καθώς και το σχεδιασμό, την ανάπτυξη και τη συντήρηση λογισμικού μέσω κατάλληλων γλωσσών προγραμματισμού και συστημάτων λογισμικού, τα οποία συνεργάζονται με την υπολογιστική υποδομή. Τέλος, προϋποθέτει την υποδομή μέσω των οποίων οι πληροφορίες διακινούνται με ασφάλεια. Από τους βασικούς τομείς της Εφαρμοσμένης Επιστήμης των Υπολογιστών που έχουν σκοπό τη διερεύνηση και την κάλυψη των προαναφερθεισών αναγκών, έχουν επιλεγεί τέσσερις. Στο **Κεφάλαιο 3.1** περιγράφονται τα Λειτουργικά Συστήματα, δηλαδή το λογισμικό που συντονίζει το υλικό και επικοινωνεί με το χρήστη. Στη συνέχεια, στο **Κεφάλαιο 3.2** επεξηγούνται τα Πληροφοριακά Συστήματα, μέσω των οποίων επιτελείται συλλογή, ανάκτηση, επεξεργασία και αποθήκευση πληροφοριών. Ακολούθως, στο **Κεφάλαιο 3.3** επιχειρείται επισκόπηση των Δικτύων Υπολογιστών για τη λήψη και την προώθηση πληροφοριών και, τέλος, στο **Κεφάλαιο 3.4** περιγράφεται η Τεχνητή Νοημοσύνη, η οποία ερευνά τρόπους ανάπτυξης υπολογιστικών μοντέλων ανθρώπινης γνώσης.

Καταβλήθηκε προσπάθεια, ώστε το υλικό του συγγράμματος να στηρίζεται στις γνώσεις και τις εμπειρίες που έχουν ήδη αποκομίσει οι μαθητές και οι μαθήτριες από την προηγούμενη σχολική τους εκπαίδευση. Με αυτόν τον τρόπο η μελέτη τους θα είναι μία ευχάριστη και δημιουργική διαδικασία οικοδόμησης της γνώσης, που πραγματοποιείται πάνω σε τεχνικές «γνωστικής σκαλωσιάς». Το βιβλίο συνοδεύεται από Παράρτημα, στο οποίο υπάρχει ένα εκτεταμένο γλωσσάρι-λεξικό όρων της Επιστήμης των Υπολογιστών.

Για την υποβοήθηση της αναγνωσιμότητας εκτός από σχήματα, πίνακες και διάφορα πλαίσια, έχουν χρησιμοποιηθεί και αρκετά εικονίδια, τα οποία χαρακτηρίζουν το μέρος του κειμένου που συνοδεύουν.

Αυτά είναι:

Προερωτήσεις

Ορισμός

Ιστορικό σημείωμα

Συμβουλή

Προσοχή

Χρήσιμη πληροφορία

Σημείωση

Ανακεφαλαίωση

Λέξεις- κλειδιά

Θα θέλαμε να ευχαριστήσουμε όλους όσους μας στήριξαν στην προσπάθεια συγγραφής αυτού του βιβλίου και ιδιαιτέρως, την κυρία Αγγελική Γερούση για τη βοήθειά της στην ανάγνωση μέρους του υλικού και τις εύστοχες παρατηρήσεις της.

Παραμένουμε στη διάθεση των εκπαιδευτικών και των μαθητών για οποιεσδήποτε παρατηρήσεις ή σχόλια, ώστε να ενισχυθεί περαιτέρω το παρόν σύγγραμμα με απώτερο στόχο να υποστηρίζει τη διδασκαλία του μαθήματος.

Αθήνα, Ιούλιος 2014
Οι συγγραφείς

ΕΝΟΤΗΤΑ 1η
Βασικές Έννοιες

ΚΕΦΑΛΑΙΟ

1.1. Επιστήμη των Υπολογιστών

Επιστήμη των Υπολογιστών

Στόχοι του κεφαλαίου είναι οι μαθητές

✓ να περιγράφουν τους βασικούς τομείς της Επιστήμης των Υπολογιστών και

✓ να μπορούν να αναφερθούν στα πεδία τόσο της Θεωρητικής όσο και σε αυτά της Εφαρμοσμένης Επιστήμης των Υπολογιστών.

Προερωτήσεις

- Τι ερευνά η Επιστήμη των Υπολογιστών; Ποιες επιστημονικές περιοχές προσπαθεί να εξελίξει;

- **Πώς συνδέεται το Θεωρητικό της μέρος με το Εφαρμοσμένο;**
- **Πώς συνδέεται η εφαρμογή της με την επίλυση προβλημάτων του πραγματικού κόσμου;**

1.1 Η Επιστήμη των Υπολογιστών

Η Επιστήμη των Υπολογιστών μελετά τα θεωρητικά θεμέλια και τη φύση των πληροφοριών, των αλγορίθμων και των υπολογισμών, καθώς και τις τεχνολογικές εφαρμογές τους σε αυτοματοποιημένα υπολογιστικά συστήματα, από τις σκοπιές σχεδίασης, ανάπτυξης, υλοποίησης, διερεύνησης και ανάλυσης. Η Επιστήμη των Υπολογιστών διακρίνεται σε

δύο μεγάλες ενότητες: τη Θεωρητική και την Εφαρμοσμένη.

Η Επιστήμη των Υπολογιστών ως διακριτή επιστήμη προέκυψε κατά τη δεκαετία του 1940 χάρη στην εύρεση των μαθηματικών ιδιοτήτων του υπολογισμού και την κατασκευή ηλεκτρονικών υπολογιστικών μηχανών.

1.2 Θεωρητική Επιστήμη των Υπολογιστών

Η Θεωρητική Επιστήμη των Υπολογιστών (Theoretical Computer Science) ερευνά κυρίως το σχεδιασμό των αλγορίθμων και των υπολογιστικών μεθόδων που χρησιμοποιούνται για την άντληση, την επεξεργασία, την ανάλυση και την αποθήκευση πληροφοριών. Βασικές έννοιες της Θεωρητικής Επιστήμης των Υπολογιστών, είναι η **Ανάλυση Αλγορίθμων**, η **Θεωρία Υπολογισιμότητας** και η **Θεωρία Πολυπλοκότητας**. Υπάρχει μία διαρκής αλληλεπίδραση μεταξύ της Θεωρητικής και της Εφαρμοσμένης Επιστήμης των Υπολογιστών. Για παράδειγμα, η **Θεωρία Γλωσσών Προγραμματισμού**, η οποία μελετά προσεγγίσεις για την περιγραφή των υπολογισμών, οδηγεί στην ανάπτυξη γλωσσών προγραμματισμού και το σχεδιασμό λογισμικού και εφαρμογών.

Η Ανάλυση Αλγορίθμων ασχολείται με τον σχεδιασμό και την ανάλυση της πολυπλοκότητας των αλγορίθμων.

Η Θεωρία Υπολογισιμότητας ερευνά αν και πόσο αποδοτικά κάποια προβλήματα μπορούν να επιλυθούν με συγκεκριμένα υπολογιστικά μοντέλα.

Η Θεωρία Πολυπλοκότητας μελετά τους πόρους που απαιτούνται για την επίλυση ενός προβλήματος βάσει ενός συγκεκριμένου αλγορίθμου.

1.3 Εφαρμοσμένη Επιστήμη των Υπολογιστών

Η Εφαρμοσμένη Επιστήμη των Υπολογιστών (Applied Computer Science) μελετά τρόπους εφαρμογής της Θεωρίας των Υπολογιστών για την επίλυση προβλημάτων στον πραγματικό κόσμο. Βασικά επιστημονικά πεδία που εντάσσονται στην Εφαρμοσμένη Επιστήμη των Υπολογιστών είναι:

- Ο σχεδιασμός υλικού για την κατασκευή των υπολογιστών, όπως ο σκληρός δίσκος, η κεντρική μονάδα επεξεργασίας κτλ.
- Ο σχεδιασμός, η ανάπτυξη και η συντήρηση λογισμικού, όπως των λειτουργικών συστημάτων τα οποία συνεργάζονται με το υλικό,

καθώς και των ποικίλων προγραμμάτων που αναπτύσσονται με τη βοήθεια των γλωσσών προγραμματισμού.

- Ο σχεδιασμός πληροφοριακών συστημάτων για τη συλλογή, ανάκτηση, επεξεργασία και αποθήκευση πληροφοριών.
- Η τεχνητή νοημοσύνη, η οποία ερευνά τρόπους ανάπτυξης υπολογιστικών μοντέλων ανθρώπινης γνώσης.
- Ο σχεδιασμός δικτύων υπολογιστών για την παραγωγή, τη λήψη και την προώθηση πληροφοριών.
- Ο σχεδιασμός βάσεων δεδομένων και συστημάτων διαχείρισης βάσεων δεδομένων για την

υποστήριξη πληροφοριακών συστημάτων.

- **Η ασφάλεια των υπολογιστών, δηλαδή το σύνολο των μεθόδων που χρησιμοποιούνται για την προστασία πληροφοριών ή υπηρεσιών από φθορά, αλλοίωση ή μη εξουσιοδοτημένη χρήση.**

Στην συνέχεια του βιβλίου θα μελετηθούν θέματα τόσο της Θεωρητικής (αλγόριθμοι και προγραμματισμός) όσο και της Εφαρμοσμένης Επιστήμης Υπολογιστών (όπως τα Λειτουργικά Συστήματα, τα Πληροφοριακά Συστήματα, τα Δίκτυα Υπολογιστών και η Τεχνητή Νοημοσύνη).

Ανακεφαλαίωση

Η Επιστήμη Υπολογιστών πραγματεύεται δύο μεγάλες θεματικές ενότητες - τη Θεωρητική και την Εφαρμοσμένη - οι οποίες περιλαμβάνουν πολλούς επί μέρους κλάδους με έμφαση τόσο στην διαχείριση πληροφοριών όσο και στην επίλυση προβλημάτων στον πραγματικό κόσμο.

Ξεκινήστε την αναζήτησή σας από την κατηγοριοποίηση του οργανισμού ACM.

Χρήσιμοι Υπερσύνδεσμοι

http://www.acm.org

Association for Computing Machinery

http://www.computer.org/portal/web/guest/home
IEEE-Computer Society

http://aisnet.org
AIS: Association for Information Systems

Λέξεις κλειδιά

Θεωρητική Επιστήμη των Υπολογιστών, Εφαρμοσμένη Επιστήμη των Υπολογιστών.

Ερωτήσεις - Θέματα προς συζήτηση - Δραστηριότητες

1. **Πώς αντιλαμβάνεστε την Επιστήμη των Υπολογιστών;**

2. **Να αναφέρετε τουλάχιστον τρία επιστημονικά πεδία που εντάσσονται στην Εφαρμοσμένη Επιστήμη των Υπολογιστών, αναζητώντας σχετικές πληροφορίες στο διαδίκτυο.**

3. **Να αναφέρετε τρεις τομείς της Θεωρητικής Πληροφορικής οι οποίοι έχουν άμεση εφαρμογή σε προβλήματα που αντιμετωπίζει η Εφαρμοσμένη Πληροφορική.**

4. **Να αναζητήσετε στο Διαδίκτυο, εργαζόμενοι σε ομάδες, όρους που σχετίζονται με την Επιστήμη των Υπολογιστών, τους τομείς της, τα πεδία εφαρμογής καθεμιάς και να συσχετίσετε τις**

έννοιες μεταξύ τους. Με βάση την αναζήτηση αυτή, να γίνει απαρίθμηση των πλέον γνωστών τομέων και ο διαχωρισμός τους σε θεωρητικούς και εφαρμοσμένους.

ΕΝΟΤΗΤΑ 2η

Θέματα Θεωρητικής Επιστήμης των Υπολογιστών

ΚΕΦΑΛΑΙΑ

2.1. Πρόβλημα

2.2. Αλγόριθμοι

2.3. Προγραμματισμός

Πρόβλημα

Στόχοι του κεφαλαίου αυτού είναι να μπορούν οι μαθητές:

- ✓ να περιγράφουν την έννοια του προβλήματος
- ✓ να κατατάσσουν ένα πρόβλημα στην κατηγορία που ανήκει
- ✓ να διακρίνουν την ύπαρξη υπολογιστικών και μη προβλημάτων
- ✓ να αναφέρουν τις φάσεις επίλυσης ενός υπολογιστικού προβλήματος.

Προερωτήσεις

- Συζητήστε με τους συμμαθητές σας και καταγράψτε δύο προβλήματα.
- Ποια είναι τα βασικότερα προβλήματα της ανθρωπότητας;
- Ρωτήστε το συμμαθητή σας ποιο θεωρεί το σημαντικότερο πρόβλημα της ανθρωπότητας και ποιο θεωρεί το σημαντικότερο πρόβλημα που χρήζει αντιμετώπισης στο σχολείο.

2.1.1 Η έννοια του προβλήματος

Οι άνθρωποι, από την πρώτη στιγμή της ύπαρξής τους, ήρθαν αντιμέτωποι με ποικίλα προβλήματα, τόσο στις καθημερινές τους δραστηριότητες όσο και σε διάφορους επιστημονικούς τομείς. Κάνοντας μια αναδρομή στην ιστορία είναι δυνατό να εντοπιστούν πληθώρα προβλημάτων.

- Ο Όμηρος στην Οδύσσεια περιγράφει τα προβλήματα που αντιμετώπιζε ο Οδυσσέας για να φτάσει στην Ιθάκη.
- Το πρόβλημα που κλήθηκε να επιλύσει ο Αρχιμήδης με τη βασιλική κορώνα που οδήγησε στη γνωστή φράση του «Εύρηκα-Εύρηκα».
- Το πρόβλημα μέτρησης του χρόνου, το οποίο αντιμετωπίστηκε με τη χρήση της κλεψύδρας και του εκκρεμούς.

- Τα προβλήματα επιδημιών στην ανθρωπότητα και η αντιμετώπισή τους με εμβόλια.
- Το πρόβλημα του «ιού του 2000» και η αντιμετώπισή του, ώστε τα υπολογιστικά συστήματα να λειτουργήσουν σωστά την 1/1/2000.

Τα προβλήματα δεν είναι απαραίτητα μαθηματικές καταστάσεις που απαιτούν αντιμετώπιση.

Εικόνα 2.1.
Ο κύβος του Ρούμπικ (Rubik)

Όπως φαίνεται, η ύπαρξη προβλημάτων είναι ένα διαχρονικό φαινόμενο. Στην εποχή μας, οι άνθρωποι έρχονται αντιμέτωποι με προβλήματα στον προσωπικό, στον επαγγελματικό και στον κοινωνικό χώρο γενικότερα. Όλοι οι άνθρωποι δεν αντιμετωπίζουν τα ίδια προβλήματα και επιπλέον δίνουν διαφορετική σημασία και βαρύτητα σε αυτά. Ωστόσο υπάρχουν προβλήματα που αναγνωρίζονται από τους περισσότερους ως πολύ σημαντικά.

Τα προβλήματα εκτός από δυσάρεστες ή πιεστικές καταστάσεις που απαιτούν λύση (περιβαλλοντικά προβλήματα, κοινωνικά προβλήματα, προσωπικά προβλήματα κ.ά.) μπορούν να είναι είτε ενδιαφέρουσες προκλήσεις (π.χ. η επίλυση ενός γρίφου ή η νίκη σε ένα παιχνίδι σκάκι), είτε ευκαιρίες για να προκύψει κάτι ωφέλιμο για την κοινωνία μέσω της επίλυσής τους (π.χ. νέα ασφαλέστερα υλικά για την κατασκευή αυτοκινήτων, τρισδιάστατες εκτυπώσεις κ.ά.).

Όλα τα προβλήματα δεν μπορούν να αντιμετωπιστούν με έναν ενιαίο και μοναδικό τρόπο. Επιπλέον, κάθε ξεχωριστό πρόβλημα μπορεί να αντιμετωπίζεται και να επιλύεται με ποικίλους τρόπους, ενώ συγχρόνως μπορεί να έχει πολλές λύσεις (το πρόβλημα οργάνωσης μιας εκπαιδευτικής επίσκεψης).

Με τον όρο Πρόβλημα προσδιορίζεται μια κατάσταση η οποία χρήζει αντιμετώπισης, απαιτεί λύση, η δε λύση της δεν είναι γνωστή, ούτε προφανής.

Η διατύπωση ενός προβλήματος και η αντιμετώπισή του, αποτελούν ζητήματα που απαιτούν ικανότητες ορθολογικής, αναλυτικής και συνθετικής σκέψης, αλλά και σωστό χειρισμό της φυσικής γλώσσας. Επιπλέον, οι δεξιότητες που αποκτούνται από την ενασχόληση με τα προβλήματα (διατύπωση και αντιμετώπισή τους), αποτελούν χρήσιμα εφόδια για κάθε ανθρώπινη δραστηριότητα.

«Η ζωή είναι επίλυση προβλημάτων», (Καρλ Πόππερ, Karl Popper, 1994)

2.1.2 Κατηγορίες Προβλημάτων

Τα προβλήματα ανάλογα με τη δυνατότητα επίλυσης διακρίνονται σε τρεις κατηγορίες.

Ως προς τη δυνατότητα επίλυσης

- Επιλύσιμα
- Μη επιλύσιμα
- Ανοικτά

Εικόνα 2.2.
Κατηγορίες προβλημάτων

Επιλύσιμα είναι εκείνα τα προβλήματα για τα οποία η λύση έχει βρεθεί και έχει διατυπωθεί.

Παραδείγματα επιλύσιμων προβλημάτων είναι η αποψίλωση μιας έκτασης γης (ο καθαρισμός δηλαδή ενός χωραφιού από κάθε είδους βλάστηση), η επίλυση της δευτεροβάθμιας εξίσωσης κ.ά..

Μη επιλύσιμα χαρακτηρίζονται εκείνα τα προβλήματα για τα οποία έχει αποδειχτεί, ότι δεν επιδέχονται λύση.

Το πρόβλημα του τετραγωνισμού του κύκλου με κανόνα και διαβήτη που είχε διατυπωθεί από τους αρχαίους ελληνιστικούς χρόνους είναι ένα τέτοιο πρόβλημα. Παρότι το πρόβλημα επιδέχεται προσεγγιστική λύση, δεν μπορεί να λυθεί με τη χρήση κανόνα και διαβήτη.

Ένα άλλο μη επιλύσιμο πρόβλημα είναι η εύρεση ακέραιων λύσεων οποιασδήποτε διοφαντικής εξίσωσης (ακέραια πολυωνυμική εξίσωση) όπως της $6x + 15y = 4$. Το 1970 αποδείχτηκε ότι το πρόβλημα είναι μη επιλύσιμο.

Ανοικτά ονομάζονται τα προβλήματα για τα οποία η λύση τους δεν έχει ακόμα βρεθεί, ενώ ταυτόχρονα δεν έχει αποδειχτεί, ότι δεν επιδέχονται λύση.

Ένα τέτοιο παράδειγμα είναι το πρόβλημα της ενοποίησης των τεσσάρων πεδίων δυνάμεων (βαρυτικού, ηλεκτρομαγνητικού, ασθενούς πυρηνικού και ισχυρού πυρηνικού), το οποίο προς το παρόν, δεν έχει επιλυθεί. Επίσης, η εικασία του Γκόλντμπαχ (Goldbach, κάθε άρτιος μπορεί να γραφτεί ως άθροισμα δύο πρώτων αριθμών) αποτελεί ένα ανοικτό πρόβλημα αφού δεν έχει ακόμα αποδειχθεί.

2.1.3 Υπολογιστικά Προβλήματα

Στις αρχές του 20ου αιώνα, ο Ντέβιντ Χίλμπερτ (David Hilbert) παρουσίασε έναν κατάλογο προβλημάτων, εκ των οποίων το τελευταίο έθετε το ερώτημα «αν υπάρχει αλγόριθμος που μπορεί να αποφασίσει την αλήθεια οποιασδήποτε λογικής πρότασης που αφορούσε τους φυσικούς αριθμούς». Με το ερώτημα αυτό ουσιαστικά ρωτούσε «αν μπορεί να αυτοματοποιηθεί η διαδικασία επίλυσης όλων των μαθηματικών προβλημάτων». Το 1931, το θεώρημα της μη πληρότητας του Κουρτ Γκέντελ (Kurt Gödel) έδειξε ότι, σε οποιαδήποτε γλώσσα που έχει την ισχύ να περιγράψει τις ιδιότητες των φυσικών αριθμών, υπάρχουν αληθείς προτάσεις των οποίων

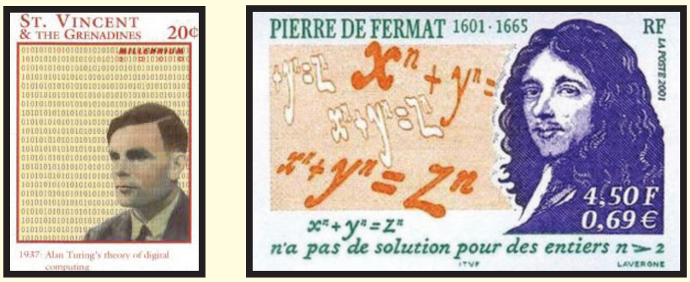
η αλήθεια δεν μπορεί να βεβαιωθεί με κανέναν αλγόριθμο. Με τον τρόπο αυτό, απέδειξε ότι υπάρχουν μερικές συναρτήσεις οι οποίες δεν μπορούν να αναπαρασταθούν από έναν αλγόριθμο, και άρα δεν μπορούν να υπολογιστούν. Στη συνέχεια, ο Άλαν Τιούρινγκ (Alan Turing) όρισε τη μηχανή Turing η οποία είναι ικανή να υπολογίσει οποιαδήποτε υπολογίσιμη συνάρτηση και έδειξε επίσης ότι υπήρχαν μερικές συναρτήσεις τις οποίες καμία μηχανή Turing δεν μπορεί να υπολογίσει.

Από τα παραπάνω φάνηκε ότι τα προβλήματα με βάση τη δυνατότητα επίλυσής τους μέσω του υπολογιστή, μπορούν να διακριθούν σε υπολογιστικά και μη υπολογιστικά.

Οποιοδήποτε πρόβλημα μπορεί να λυθεί και μέσω του υπολογιστή, χαρακτηρίζεται υπολογιστικό πρόβλημα.

Για να λυθεί ένα πρόβλημα με τη βοήθεια του υπολογιστή, χρειάζεται να διατυπωθεί το αντίστοιχο υπολογιστικό πρόβλημα και στη συνέχεια να υλοποιηθεί η επίλυσή του μέσω του υπολογιστή.

Εικόνα 2.3.
David Hilbert
και Kurt Gödel



Εικόνα 2.4.
Alan Turing και
Pierre de Fermat

Παραδείγματα υπολογιστικών προβλημάτων είναι:

- Η επίλυση της δευτεροβάθμιας εξίσωσης.
- Η ταξινόμηση των μαθητών σε αλφαβητική σειρά.
- Η αναζήτηση και ο υπολογισμός της χιλιομετρικά συντομότερης διαδρομής που θα κάνει ένας ταχυδρόμος για να επισκεφθεί δέκα χωριά και να επιστρέψει στο χωριό από όπου ξεκίνησε περνώντας μόνο μία φορά από κάθε χωριό, με βάση έναν δεδομένο χάρτη των χωριών και των δρόμων που συνδέουν τα χωριά.
- Η εύρεση λέξης που να ξεκινά

από ένα γράμμα και να τελειώνει σε ένα άλλο γράμμα.

Από την άλλη, τα μη υπολογιστικά προβλήματα δεν μπορούν να λυθούν από έναν υπολογιστή ή από άλλα μηχανικά μέσα. Για παράδειγμα, καμία μηχανή δεν μπορεί γενικά να αποφανθεί αν ένα δεδομένο πρόγραμμα θα επιστρέψει απάντηση για μια δεδομένη είσοδο, ή αν θα εκτελείται για πάντα.

Το θεώρημα του Πιέρ ντε Φερμά (Pierre de Fermat): «Τρεις θετικοί αριθμοί a, b, c δεν μπορούν να ικανοποιήσουν την εξίσωση $a^n + b^n = c^n$ για κάθε ακέραιο

αριθμό n > 2» διατυπώθηκε από τον ίδιο το 1637 ως σημείωση στο βιβλίο του «Αριθμητικά του Διόφαντου». Η επιτυχής απόδειξη του θεωρήματος δημοσιεύθηκε το 1995.

2.1.4 Διαδικασίες επίλυσης (υπολογιστικού) προβλήματος

Η προσπάθεια αντιμετώπισης και επίλυσης ενός προβλήματος προϋποθέτει αρχικά την πλήρη κατανόηση του προβλήματος. Η **κατανόηση** ενός προβλήματος αποτελεί συνάρτηση δύο παραγόντων, της σωστής διατύπωσης εκ μέρους του δημιουργού του και της αντίστοιχα σωστής ερμηνείας από τη μεριά εκείνου που καλείται να το αντιμετωπίσει.

Η κατανόηση του προβλήματος είναι βασική προϋπόθεση για να ξεκινήσει η διαδικασία ανάλυσης του προβλήματος σε άλλα απλούστερα και ο διαχωρισμός των κύριων στοιχείων του προβλήματος σε σχέση με τα δευτερεύοντα στοιχεία (αφαίρεση). Η **ανάλυση-αφαίρεση** αποτελεί το δεύτερο βήμα στην διαδικασία επίλυσης ενός προβλήματος. Στόχος της ανάλυσης, είναι η διάσπαση του προβλήματος σε άλλα απλούστερα προβλήματα για να είναι εύκολη η αντιμετώπισή τους.

Για να διατυπωθεί ένα πρόβλημα μπορεί να χρησιμοποιηθεί οποιοδήποτε μέσο με συνηθέστερα τον προφορικό ή το γραπτό λόγο. Ο λόγος όμως όταν χρησιμοποιείται για την επικοινωνία, χρειάζεται να χαρακτηρίζεται από σαφήνεια.

Η ανάλυση ενός προβλήματος μπορεί να πραγματοποιηθεί είτε φραστικά είτε διαγραμματικά, όπως φαίνεται στο παράδειγμα 2.1.

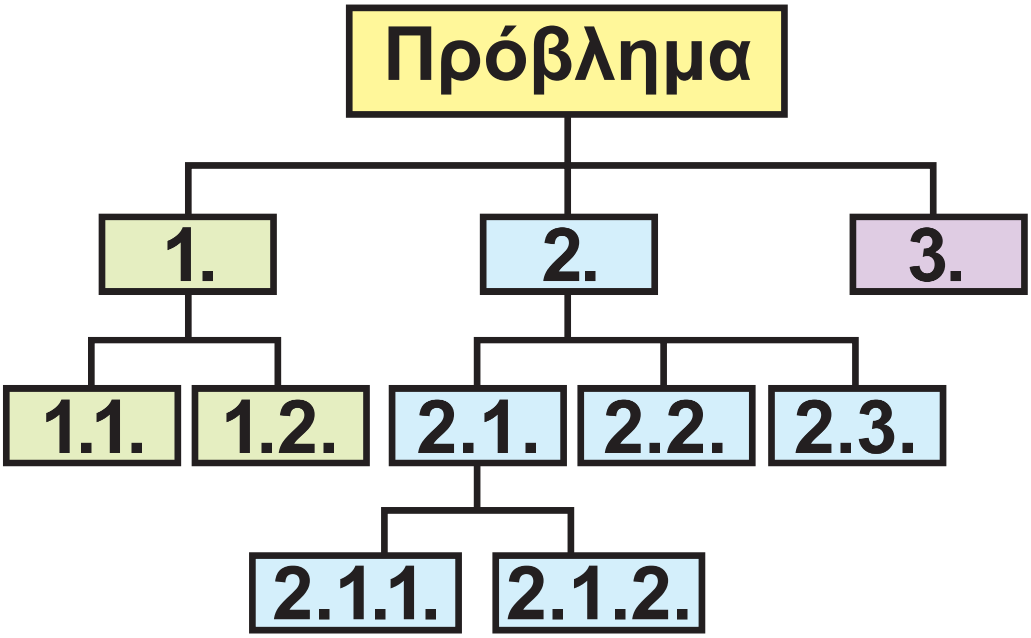
Η άστοχη χρήση ορολογίας και η λανθασμένη σύνταξη, μπορούν να προκαλέσουν παρερμηνείες και παραπλανήσεις. Ωστόσο, παρερμηνείες μπορούν να υπάρξουν ακόμα και σε περιπτώσεις όπου όλοι οι λεξικολογικοί και συντακτικοί κανόνες τηρούνται.

Παράδειγμα 2.1. **Ανάλυση προβλήματος: Εξυπηρέτηση πολιτών από τις υπηρεσίες του δημοσίου.**

Φραστική ανάλυση

1. Προσδιορισμός αναγκών
    - 1.1. Ταχύτερη εξυπηρέτηση πολιτών
    - 1.2. Περιορισμός μετακινήσεων
2. Δράση
    - 2.1. Ανάπτυξη ηλεκτρονικών υπηρεσιών εξυπηρέτησης
        - 2.1.1. Ποιες υπηρεσίες θα είναι διαθέσιμες;
        - 2.1.2. Με ποια διαδικασία θα γίνονται διαθέσιμες;
    - 2.2. Ενημέρωση πολιτών
    - 2.3. Ενημέρωση υπαλλήλων για να συνδράμουν το έργο
3. Εφαρμογή του σχεδίου.

Διαγραμματική ανάλυση



Εικόνα 2.5.
Διαγραμματική αναπαράσταση ανάλυσης προβλήματος

Στη διαγραμματική αναπαράσταση, το αρχικό πρόβλημα αναπαρίσταται από ένα ορθογώνιο παραλληλόγραμμο. Κάθε ένα από τα απλούστερα προβλήματα, αναπαρίσταται επίσης

από ένα ορθογώνιο παραλληλόγραμμο.

Τα παραλληλόγραμμα που αντιστοιχούν στα απλούστερα προβλήματα, σχηματίζονται ένα επίπεδο χαμηλότερα.

Για τη σωστή επίλυση ενός προβλήματος είναι σημαντικός ο επακριβής προσδιορισμός των **δεδομένων** που παρέχει το πρόβλημα και η λεπτομερειακή καταγραφή των **ζητούμενων** που αναμένονται σαν αποτελέσματα της επίλυσης του προβλήματος. Για να βρει κάποιος τα ζητούμενα χρειάζεται να επεξεργαστεί τα δεδομένα.

Επεξεργασία δεδομένων είναι η συστηματική εκτέλεση πράξεων σε δεδομένα.

Δεδομένο είναι μια παράσταση γεγονότων, εννοιών ή εντολών σε τυποποιημένη μορφή που είναι κατάλληλη για επικοινωνία, ερμηνεία ή επεξεργασία από τον άνθρωπο ή από αυτόματα μέσα.

Με τον όρο **ζητούμενο** δηλώνεται οτιδήποτε προκύπτει ή τίθεται ως αντικείμενο έρευνας ή αναζήτησης.

Με τον όρο **πληροφορία** αναφέρεται οποιοδήποτε γνωσιακό στοιχείο προέρχεται από επεξεργασία δεδομένων.

Αφού ολοκληρωθεί η ανάλυση του προβλήματος ακολουθεί το στάδιο της σύνθεσης. Κατά τη **σύνθεση** επιχειρείται η κατασκευή μιας νέας δομής, με την οργάνωση των επιμέρους στοιχείων του προβλήματος. Επιπλέον, η **κατηγοριοποίηση** του προβλήματος είναι ένα εξίσου σημαντικό στάδιο, μέσω του οποίου το πρόβλημα κατατάσσεται σε κάποια κατηγορία, σε μία οικογένεια παρόμοιων προβλημάτων και έτσι διευκολύνεται η επίλυση, αφού παρέχεται η ευκαιρία να προσδιοριστεί το ζητούμενο ανάμεσα σε παρόμοια «αντικείμενα». Τέλος, με

τη γενίκευση, μπορούν να μεταφερθούν τα αποτελέσματα σε άλλες παρεμφερείς καταστάσεις ή προβλήματα.

Παράδειγμα 2.2. **Να διερευνηθεί η εξίσωση $\alpha x + \beta = 0$ ως προς x, για τις διάφορες τιμές του α και β.**

Απάντηση

Υπάρχουν 2 περιπτώσεις: Αν ο συντελεστής της μεταβλητής x είναι διάφορος του μηδενός ($\alpha \neq 0$) ή αν ο συντελεστής της μεταβλητής x είναι ίσος με μηδέν ($\alpha = 0$).

Περίπτωση 1: **Αν $\alpha \neq 0$, τότε η εξίσωση έχει μοναδική λύση την $x = -\dfrac{\beta}{\alpha}$.**

Περίπτωση 2: Αν $\alpha = 0$ τότε υπάρχουν δύο υποπεριπτώσεις: Αν ο σταθερός όρος β είναι διάφορος του μηδενός ($\beta \neq 0$) ή αν είναι ίσος με μηδέν ($\beta = 0$).

Περίπτωση 2.1.
Αν $\beta \neq 0$, η εξίσωση είναι αδύνατη.

Περίπτωση 2.2.
Αν $\beta = 0$, η εξίσωση είναι αόριστη.

Κατανόηση: Δίνονται οι σταθεροί όροι α, β της εξίσωσης και ζητείται η τιμή της μεταβλητής x για τις διάφορες τιμές των α και β.

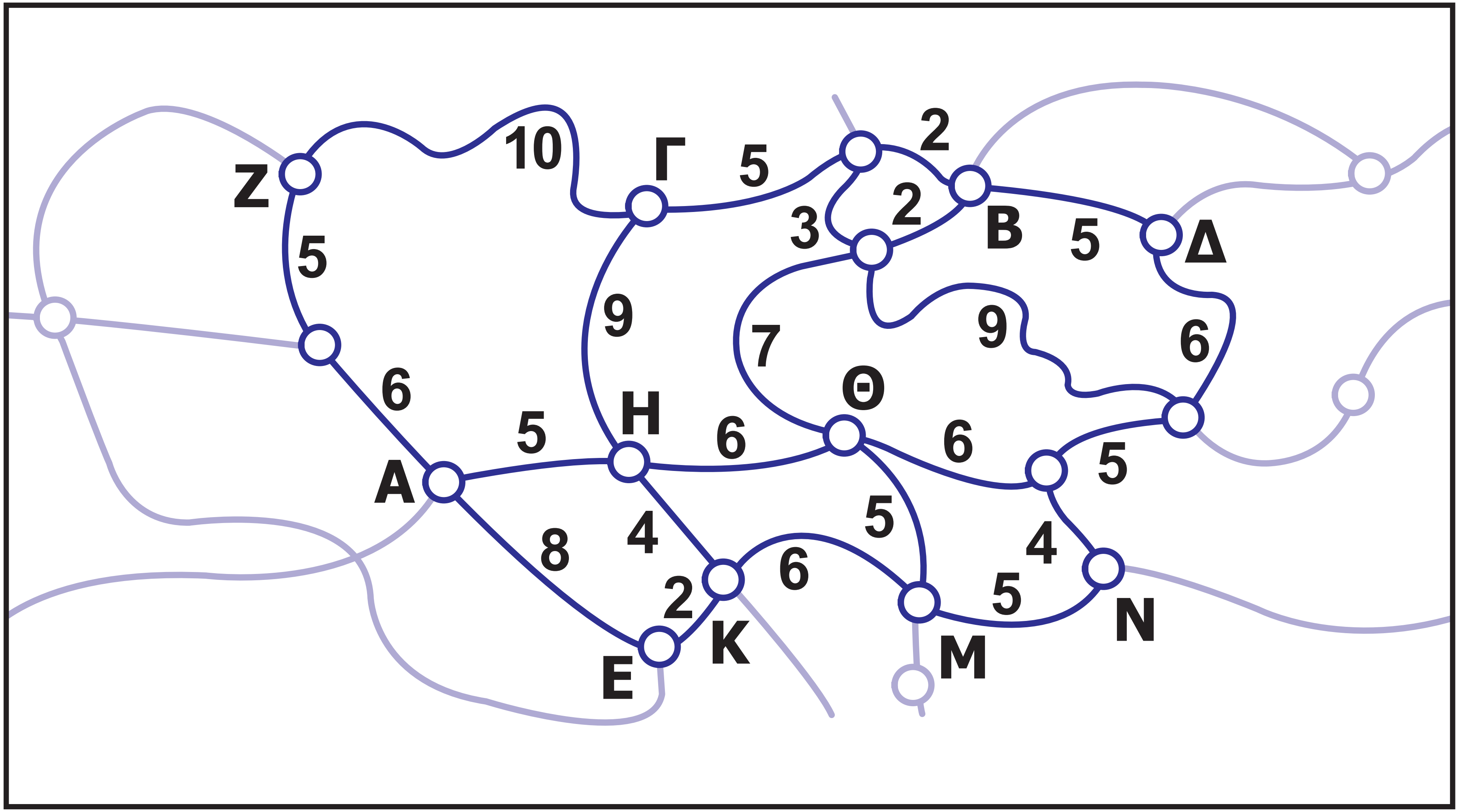
Ανάλυση: Το πρόβλημα διασπάται αρχικά σε δύο υποπροβλήματα. Στο πρώτο ο συντελεστής α της μεταβλητής x είναι διάφορος του μηδενός. Στο δεύτερο ο συντελεστής είναι μηδέν. Το δεύτερο υποπρόβλημα διασπάται σε δύο υποπροβλήματα: Ο σταθερός όρος β είναι ίσος με μηδέν ή είναι διάφορος του μηδενός.

Σύνθεση: Η εξίσωση είτε έχει μοναδική λύση, είτε είναι αδύνατη, είτε είναι αόριστη.

Κατηγοριοποίηση και γενίκευση: όλες οι πρωτοβάθμιες εξισώσεις αντιμετωπίζονται με αυτή την προσέγγιση.

Παράδειγμα 2.3. Δίνεται ο ακόλουθος χάρτης διαδρομών (εικόνα 2.6) που συνδέει ορισμένες πόλεις. Ο χάρτης δείχνει το χρόνο που απαιτείται για τη μετακίνηση από πόλη σε πόλη.

1. Ποια διαδρομή είναι η συντομότερη από την πόλη Α στην πόλη Β;
2. Σε ποια πόλη θα συναντηθούν τρεις φίλοι ώστε κανένας να μην κινηθεί περισσότερο από δεκαπέντε λεπτά αν βρίσκονται στις πόλεις Γ, Δ και Ε αντίστοιχα και τα τρένα τους ξεκινούν όλα στις 19:00;



Εικόνα 2.6. Χάρτης διαδρομών

Το παράδειγμα 2.3 εντάσσεται σε μία γενικότερη κατηγορία προβλημάτων εύρεσης συντομότερης διαδρομής.

Απάντηση

1. Όπως φαίνεται στο χάρτη, υπάρχουν πολλές διαδρομές για να μεταβεί κάποιος από την πόλη Α στην πόλη Β. Από αυτές τις διαδρομές χρειάζεται να υπολογιστεί η συντομότερη με βάση το χρόνο που απαιτείται για τη μετακίνηση από πόλη σε πόλη. Για το σκοπό αυτό, μετριούνται οι αποστάσεις μεταξύ των πόλεων, από όπου προκύπτει ότι η συντομότερη με βάση το χρόνο διαδρομή είναι 20 λεπτά.

2. Με παρόμοιο τρόπο, μετριούνται οι σχετικές αποστάσεις. Η συνάντηση θα γίνει στην πόλη Θ, αφού ο άνθρωπος από την πόλη Γ θα κάνει 15 λεπτά, ο άνθρωπος από την πόλη Δ θα κάνει 14 λεπτά και ο άνθρωπος από την πόλη Ε θα κάνει 12 λεπτά.

- **Επίλυση προβλήματος**
  - Κατανόηση
  - Ανάλυση-Αφαίρεση
  - Σύνθεση
  - Κατηγοριοποίηση
  - Γενίκευση

Εικόνα 2.7.
Στάδια επίλυσης προβλήματος

Ανακεφαλαίωση

Στο κεφάλαιο αυτό παρουσιάστηκε η διαχρονικότητα του προβλήματος και επιχειρήθηκε να γίνει σαφής η ανεξαρτησία της λύσης του από τον υπολογιστή. Κατηγοριοποιήθηκαν τα προβλήματα ως προς τη δυνατότητα επίλυσης και επιπλέον ως προς τη δυνατότητα επίλυσης με τον υπολογιστή. Επισημάνθηκαν βασικά στοιχεία στη διαδικασία επίλυσης ενός προβλήματος (κατανόηση, καθορισμός δεδομένων και ζητουμένων, ανάλυση-αφαίρεση, σύνθεση, κατηγοριοποίηση και γενίκευση).

Το 1976 το θεώρημα των 4 χρωμάτων αποδείχθηκε. Για την απόδειξη χρησιμοποιήθηκε ένας υπολογιστής για 1200 ώρες.

Λέξεις κλειδιά

Πρόβλημα, Επιλύσιμα, Ανοικτά, Υπολογιστικά, Επίλυση προβλήματος, Κατανόηση, Ανάλυση, Αφαίρεση, Σύνθεση, Κατηγοριοποίηση, Γενίκευση

Ερωτήσεις - Θέματα προς συζήτηση - Δραστηριότητες

1. Να αναφέρετε τις κατηγορίες προβλημάτων ως προς τη δυνατότητα επίλυσης και να δώσετε

παραδείγματα προβλημάτων από κάθε κατηγορία.

2. Εργαστείτε σε ομάδες. Κάθε ομάδα θα θέτει ένα πρόβλημα στην άλλη ομάδα, η οποία καλείται να ανακαλύψει την κατηγορία στην οποία ανήκει το πρόβλημα.

3. Να διερευνήσετε την εξίσωση $αx^2 + βx + γ = 0$ ως προς x για τις διάφορες τιμές των α, β και γ.

4. Μπορεί κάθε χάρτης να χρωματιστεί με τέσσερα χρώματα το πολύ, ώστε οι γειτονικές χώρες να είναι χρωματισμένες διαφορετικά;

5. Να χαρακτηρίσετε με Σωστό ή Λάθος τις παρακάτω προτάσεις:

Α. Κάθε επιλύσιμο πρόβλημα εί-
ναι υπολογιστικό.
Β. Η κατανόηση προηγείται της
επίλυσης.
Γ. Όλα τα προβλήματα μπορούν
να λυθούν με τη βοήθεια του
υπολογιστή.
Δ. Η ανάλυση του προβλήματος
βοηθάει στην επίλυσή του.
Ε. Υπάρχουν μη υπολογιστικά
μαθηματικά προβλήματα.

Αλγόριθμοι

Στόχοι του κεφαλαίου αυτού είναι να μπορούν οι μαθητές:

- ✓ να περιγράφουν την έννοια του αλγορίθμου και να διακρίνουν την ύπαρξη συγκεκριμένων χαρακτηριστικών που χρειάζεται να έχει ένας αλγόριθμος
- ✓ να αναγνωρίζουν βασικές έννοιες στην Ανάλυση Αλγορίθμων
- ✓ να αναγνωρίζουν τις διάφορες μορφές αναπαράστασης αλγορίθμου
- ✓ να αναφέρουν τους βασικούς τύπους και δομές δεδομένων

- ✓ να διακρίνουν τις βασικές εντολές και δομές που χρησιμοποιούνται σε έναν αλγόριθμο
- ✓ να προσδιορίζουν τον τρόπο λειτουργίας των δομών δεδομένων
- ✓ να εκπονούν απλούς αλγορίθμους
- ✓ να εντοπίζουν και να διορθώνουν τα λογικά λάθη ενός αλγορίθμου
- ✓ να εξηγούν την ανάγκη δημιουργίας της κατάλληλης τεκμηρίωσης.

Προερωτήσεις

- Ξέρεις ότι ήδη έχεις χρησιμοποιήσει πολλούς αλγορίθμους;

- **Μπορείς να περιγράψεις πώς βρίσκεις το Μέγιστο Κοινό Διαιρέτη δύο αριθμών;**
- **Τι κάνεις για να εντοπίσεις μία λέξη σε ένα λεξικό;**

2.2.1 Ορισμός αλγορίθμου

Η λέξη αλγόριθμος (algorithm) προέρχεται από μια μελέτη του Πέρση μαθηματικού Μοχάμεντ Ιμπν Μουσά Αλ Χουαρίζμι (Muhammad ibn Mūsā al-Khwārizmī), που έζησε περί το 825 μ.Χ. (Εικόνα 2.8). Παρόλα αυτά, η ύπαρξη και η ηλικία μερικών αλγορίθμων αριθμεί χιλιάδες χρόνια. Σήμερα, το πεδίο της μελέτης των αλγορίθμων (το οποίο καλείται θεωρία αλγορίθμων) είναι

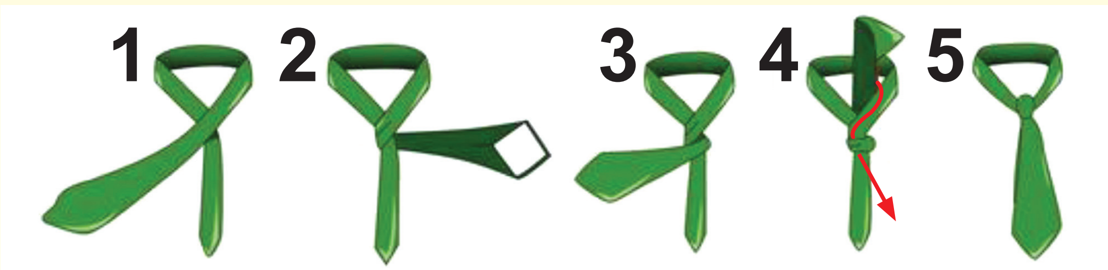
ένα ιδιαίτερα ευρύ πεδίο έρευνας. Γενικά,

Αλγόριθμος είναι μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος.

Η μελέτη του Αλ Χουαρίζμι μεταφράστηκε στα λατινικά και άρχιζε με τη φράση «Algoritmi dixit...» (ο αλγόριθμος λέει ....).

Εικόνα 2.8. Ο Αλ Χουαρίζμι

Η έννοια του αλγορίθμου δεν συνδέεται αποκλειστικά και μόνο με προβλήματα της Πληροφορικής. Για παράδειγμα, το δέσιμο της γραβάτας αποτελεί ένα πρόβλημα, για την επίλυση του οποίου χρειάζεται να εκτελεστεί μια πεπερασμένη σειρά ενεργειών (Εικόνα 2.9.). Η αλληλουχία των ενεργειών οδηγεί στο επιθυμητό αποτέλεσμα. Η αλληλουχία δεν είναι απαραίτητα μοναδική για την επίτευξη αυτού του, αφού, υπάρχουν και άλλοι τρόποι για το δέσιμο της γραβάτας.



Εικόνα 2.9. Αλγόριθμος δεσίματος γραβάτας

Ο όρος αλγόριθμος επέζησε επί χίλια χρόνια ως σπάνιος όρος, που σήμαινε κάτι σαν «συστηματική διαδικασία αριθμητικών χειρισμών». Τη σημερινή του αξία απόκτησε στις αρχές του 20ού αιώνα με την ανάπτυξη της θεωρίας αλγορίθμων και φυσικά με τους υπολογιστές.

Ιστορικά, ένας από τους πρώτους αλγορίθμους, είναι ο αλγόριθμος για την εύρεση του Μέγιστου Κοινού Διαιρέτη (ΜΚΔ) δύο ακεραίων αριθμών x και y.

«Ο ευκλείδειος αλγόριθμος είναι ο παππούς όλων των αλγορίθμων, αφού είναι ο παλαιότερος μη τετριμμένος αλγόριθμος που χρησιμοποιείται ακόμα και σήμερα.» (Ντόναλντ Κνουθ, Donald Knuth, 1981)

Παράδειγμα 2.4. Να βρεθεί ο Μέγιστος Κοινός Διαιρέτης (ΜΚΔ) δύο θετικών ακεραίων αριθμών x και y.

Απάντηση

Ο αλγόριθμος περιγράφεται σε ομιλούμενη γλώσσα ως εξής: Θέσε στο z τον διαιρέτη. Αν $z = 0$, τότε ΜΚΔ είναι ο x. Αν $z \neq 0$ τότε διαίρεσε το x με το y, και έστω z το υπόλοιπο και επανάλαβε τη διαίρεση με τους ακέραιους y και z αντί για x και y μέχρι το z να γίνει 0.

Για παράδειγμα προκειμένου να βρεθεί ο ΜΚΔ δύο μη μηδενικών

αριθμών, π.χ. των 78 και 27, σύμφωνα με τον αλγόριθμο μπορείτε να κάνετε τις ακόλουθες ενέργειες:

Βρείτε το υπόλοιπο της διαίρεσης του 78 με το 27. Το υπόλοιπο είναι 24. Ελέγξτε αν είναι 0. Στην περίπτωση αυτή δεν είναι 0. Αφού δεν είναι 0, βρείτε το υπόλοιπο της διαίρεσης του 27 με το 24. Το υπόλοιπο είναι 3. Ελέγξτε αν είναι 0. Στην περίπτωση αυτή δεν είναι 0. Αφού δεν είναι 0, βρείτε το υπόλοιπο της διαίρεσης του 24 με το 3. Το υπόλοιπο είναι 0. Αφού το υπόλοιπο είναι 0, βρέθηκε ο ΜΚΔ. Ο ΜΚΔ είναι 3.

Ο αλγόριθμος αυτός μπορεί να εκφραστεί και με κωδικοποιημένο τρόπο ως εξής:

1. **Αλγόριθμος** Ευκλείδης
2. **Διάβασε** x, y
3. z ← y
4. **Όσο** z ≠ 0 **επανάλαβε**
5. z ← x **mod** y
6. x ← y
7. y ← z
8. **Τέλος_επανάληψης**
9. **Εμφάνισε** x
10. **Τέλος** Ευκλείδης

Για την περιγραφή του αλγορίθμου χρησιμοποιείται μια γλώσσα στην οποία το όνομα του αλγορίθμου (Ευκλείδης) καθορίζει την αρχή και το τέλος του.

Οι τιμές εισόδου (x, y) δίνονται με την εντολή Διάβασε, ενώ ο ΜΚΔ είναι η τιμή που παίρνει τελικά η μεταβλητή x, η οποία εμφανίζεται.

Στον ευκλείδειο αλγόριθμο, δεν έχει σημασία αν θα είναι η τιμή του y μικρότερη από την τιμή του x. Έτσι, στην αρχική εκτέλεση του αλγορίθμου έχει επιλεγεί ο πρώτος αριθμός να είναι μεγαλύτερος του δευτέρου. Στη συνέχεια θα εκτελεστεί ο αλγόριθμος με τους ίδιους αριθμούς, όπου ο πρώτος αριθμός θα είναι μικρότερος του δευτέρου.

Απαραίτητο είναι οι τιμές των δύο μεταβλητών να είναι ακέραιες και μία εκ των δύο μεταβλητών να είναι διάφορη του μηδενός. Στην περίπτωση αρνητικών τιμών ο ΜΚΔ προκύπτει από τη μελέτη των απολύτων τιμών των δύο αριθμών.

Η εκτέλεση του ευκλείδειου αλγορίθμου, μπορεί να πραγματοποιηθεί από κάποιον που δεν χρειάζεται απαραίτητα να έχει κατασκευάσει τον αλγόριθμο. Επιπλέον, ο αλγόριθμος εφόσον κωδικοποιηθεί σε γλώσσα προγραμματισμού, μπορεί να εκτελεστεί από τον υπολογιστή.

Η εύρεση του ΜΚΔ είναι μια επαναληπτική διαδικασία που συνεχίζεται όσο το υπόλοιπο (**mod**) της διαίρεσης x διά του y είναι διάφορο του μηδενός. Η επαναληπτική διαδικασία έχει την έννοια «όσο ισχύει η συνθήκη (δηλαδή όσο $z \neq 0$) επαναλάμβανε τη διαδικασία, αλλιώς μην εκτελείς άλλο τη διαδικασία και συνέχισε στο επόμενο βήμα». Οι εντολές του τύπου $x \leftarrow y$ δεν έχουν την έννοια της ισότητας, αλλά της εκχώρησης τιμής του δεξιού μέλους στη μεταβλητή του αριστερού μέλους. Αυτό σημαίνει ότι μετά την εκτέλεση της εντολής η μεταβλητή του αριστερού μέλους θα έχει τιμή ίση με αυτή του δεξιού μέλους.

Με βάση τα παραπάνω, ο ευκλείδειος αλγόριθμος για τον υπολογισμό του ΜΚΔ δύο θετικών ακεραίων αριθμών, περιγράφεται πλήρως με ακρίβεια και σαφήνεια. Συνεπώς, αν το ζητούμενο είναι να υπολογιστεί με τη χρήση του αλγόριθμου Ευκλείδης, ο ΜΚΔ των αριθμών 27 και 78, τότε θα μπορούσε να αξιοποιηθεί η αρίθμηση των γραμμών του αλγορίθμου και να πραγματοποιηθεί εκτέλεσή του στο χαρτί. Σε κάθε επανάληψη υπολογίζονται οι τιμές

των x, y και z, οι οποίες παρουσιάζονται στον πίνακα 2.1. Με την ολοκλήρωση προκύπτει ότι ο ΜΚΔ των αριθμών 27 και 78 είναι ο αριθμός 3.

Πίνακας 2.1. Εικονική εκτέλεση του ευκλείδειου αλγορίθμου

Αριθμός εντολής	x	y	z	z ≠ 0	Έξοδος
2	27	78			
3			78		
4				Αληθής	
5			27		
6	78				
7		27			
4				Αληθής	
5			24		

6	27				
7		24			
4				Αληθής	
5			3		
6	24				
7		3			
4				Αληθής	
5			0		
6	3				
7		0			
4				Ψευδής	
9					3

Η εκτέλεση ενός αλγορίθμου πραγματοποιείται αφού αριθμηθούν οι γραμμές του αλγορίθμου. Για κάθε εντολή που εκτελείται, καταγράφετε τον αριθμό της γραμμής και το αποτέλεσμα της εκτέλεσης στο αντίστοιχο κελί.

Η αρίθμηση των γραμμών του αλγορίθμου είναι απαραίτητη μόνο για την εκτέλεσή του.

Οι εντολές της γραμμής 1 και 10 είναι δηλωτικές εντολές και δεν αποτυπώνονται στον πίνακα 2.1.

Ο παραπάνω αλγόριθμος, μπορεί να απαντήσει όχι μόνο στη συγκεκριμένη ερώτηση, «να βρεθεί ο ΜΚΔ των 27 και 78», αλλά σε όλες τις παρόμοιες ερωτήσεις. Λύνει, δηλαδή, ένα πρόβλημα. Κάθε μία από τις ερωτήσεις αυτές λέγεται στιγμιότυπο του προβλήματος. Έτσι, η εύρεση του ΜΚΔ των 27 και 78 είναι ένα στιγμιότυπο του προβλήματος της εύρεσης του ΜΚΔ δύο θετικών ακεραίων. Δηλαδή, αν εκτελεστούν τα βήματα του αλγορίθμου, θα ολοκληρωθεί η διαδικασία έχοντας πάρει τη σωστή απάντηση για οποιοδήποτε ζευγάρι θετικών ακεραίων.

Ωστόσο, ένα θεωρητικό ερώτημα που προκύπτει είναι το ακόλουθο: «γιατί ο αλγόριθμος λύνει οποιοδήποτε στιγμιότυπο του προβλήματος;» Συνήθως, για να λύνει πραγματικά ο αλγόριθμος ένα πρόβλημα, χρειάζεται να μπορεί να αποδειχτεί η ορθότητά του με αυστηρό τρόπο. Στην περίπτωση του ευκλείδειου αλγορίθμου, αποδεικνύεται από τον ίδιο τον Ευκλείδη στο έβδομο βιβλίο των «Στοιχείων» του.

Ο Ευκλείδης έζησε περίπου από το 330 έως το 275 π.Χ. και έγραψε το έργο «Τα Στοιχεία» που αποτελείται από 13 βιβλία, τα οποία επιχειρούν να συστηματοποιήσουν τις μαθηματικές γνώσεις της εποχής του.

Ο ευκλείδειος αλγόριθμος έχει πολλές θεωρητικές και πρακτικές εφαρμογές. Μέσω αυτού μπορούν να δημιουργηθούν αρκετοί παραδοσιακοί μουσικοί ρυθμοί. Χρησιμοποιείται στην κρυπτογραφία, στο ηλεκτρονικό εμπόριο, στην επίλυση διοφαντικών εξισώσεων κ.α.

2.2.2 Χαρακτηριστικά αλγορίθμου

Κάθε αλγόριθμος είναι σημαντικό να έχει ορισμένα χαρακτηριστικά προκείμενου να θεωρείται πλήρης.

Χαρακτηριστικά ενός αλγορίθμου

1. Καθοριστικότητα (Definiteness)
2. Περατότητα (Finiteness)
3. Αποτελεσματικότητα (Effectiveness)
4. Είσοδος (Input)
5. Έξοδος (Output)

Καθοριστικότητα: Κάθε εντολή ενός αλγορίθμου χρειάζεται να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της.

Κατά τη διαίρεση δύο ακεραίων αριθμών, το χαρακτηριστικό της καθοριστικότητας ικανοποιείται αν έχει ληφθεί υπόψη και η περίπτωση που ο διαιρέτης είναι μηδέν.

Περατότητα: Κάθε αλγόριθμος πρέπει να τελειώνει μετά από πεπερασμένα βήματα εκτέλεσης των εντολών του.

Ένας αλγόριθμος για να διαθέτει το χαρακτηριστικό της περατότητας, χρειάζεται να προσδιορίζει τη λύση ενός προβλήματος μετά από ένα συγκεκριμένο αριθμό βημάτων και να μην εκτελείται ατέρμονα (δηλαδή χωρίς να τελειώνει).

Αποτελεσματικότητα: Κάθε εντολή ενός αλγορίθμου χρειάζεται να είναι διατυπωμένη απλά και κατανοητά, ώστε να μπορεί να εκτελεστεί επακριβώς και σε πεπερασμένο μήκος χρόνου.

Κατά τη διαίρεση δύο ακέραιων αριθμών, ο αλγόριθμος διαθέτει το χαρακτηριστικό της αποτελεσματικότητας, αφού οι ακέραιοι αναπαρίστανται ακριβώς και υπάρχει αλγόριθμος για τη διαίρεσή τους (Ευκλείδεια Διαίρεση) σε πεπερασμένο χρόνο. Αν όμως επιχειρείται η διαίρεση δύο πραγματικών αριθμών που ο καθένας αναπαρίσταται από άπειρα δεκαδικά ψηφία, τότε ο αλγόριθμος δεν διαθέτει το χαρακτηριστικό της αποτελεσματικότητας, αφού δεν μπορεί να αναπαρασταθεί πλήρως και να εκτελεστεί ακριβώς η συγκεκριμένη διαίρεση.

Είσοδος: Κάθε αλγόριθμος χρειάζεται να δέχεται ένα σύνολο μεταβλητών εισόδου (που μπορεί να είναι και το κενό σύνολο), οι οποίες αποτελούν τα δεδομένα του αλγορίθμου.

Η είσοδος των μεταβλητών μπορεί να επιτευχθεί με κατάλληλες εντολές, οι οποίες θα παρουσιαστούν σε επόμενες παραγράφους. Στην περίπτωση του αλγόριθμου Ευκλείδης που παρουσιάστηκε στο παράδειγμα 2.4, αυτό επιτυγχάνεται με την εντολή **Διάβασε x, y.**

Όπως θα παρουσιαστεί σε επόμενη παράγραφο, η είσοδος σε ένα αλγόριθμο μπορεί να επιτευχθεί με την εντολή Διάβασε, την εντολή Δεδομένα και την εντολή εκχώρησης. Με την εντολή εκχώρησης δημιουργούνται δεδομένα μέσα στον αλγόριθμο (κενό σύνολο μεταβλητών εισόδου).

Έξοδος: Κάθε αλγόριθμος χρειάζεται να δημιουργεί κάποιο αποτέλεσμα.

Το αποτέλεσμα του αλγορίθμου, η έξοδός του, είναι μία ή περισσότερες μεταβλητές ή/και σταθερές τιμές. Η έξοδος μπορεί να επιτευχθεί με κατάλληλες εντολές, οι οποίες θα παρουσιαστούν σε επόμενες παραγράφους. Στην περίπτωση του αλγόριθμου Ευκλείδης που παρουσιάστηκε στο παράδειγμα 2.4, αυτό επιτυγχάνεται με την εντολή **Εμφάνισε x.**

2.2.3 Ανάλυση Αλγορίθμων, Θεωρία Υπολογισμού, Πολυπλοκότητα Αλγορίθμων, Υπολογισιμότητα Αλγορίθμων.

Η ανάλυση της συμπεριφοράς ενός αλγορίθμου για συνθήκες παρόμοιες με αυτές που εμφανίζονται στην πράξη και η ικανοποιητική απόδοσή του, παρέχει τη δυνατότητα να πραγματοποιηθεί η υλοποίηση και η εφαρμογή του. Αν δεν επιτυγχάνεται ικανοποιητική απόδοση τότε απαιτείται ο σχεδιασμός ενός τροποποιημένου αλγορίθμου.

Η Θεωρία Υπολογισμού (Theory of computation) είναι το πεδίο της πληροφορικής που ασχολείται τόσο με το πρόβλημα ύπαρξης λύσης ενός προβλήματος όσο και αποδοτικότητας των αλγορίθμων για την επίλυση των προβλημάτων με βάση ένα δεδομένο μοντέλο υπολογισμού.

Η πιο συνηθισμένη ανάλυση ενός αλγορίθμου αφορά το χρόνο εκτέλεσης. Ο χρόνος συνήθως υπολογίζεται σαν συνάρτηση του αριθμού των στοιχειωδών βημάτων που εκτελούνται στον αλγόριθμο.

Το πεδίο της θεωρίας υπολογισμού, διαιρείται σε δύο κλάδους: τη Θεωρία Υπολογισιμότητας (Computability Theory) και τη Θεωρία Πολυπλοκότητας (Computational Complexity Theory). Συνεπώς, για κάθε αλγόριθμο απαιτείται ανάλυση, μέσω της οποίας:

α) τεκμηριώνεται η ορθότητά του, δηλαδή αν ο αλγόριθμος κάνει

πραγματικά τη δουλειά για την οποία έχει σχεδιαστεί και

β) μετριέται ποσοτικά η απόδοσή του, σε σχέση με διάφορα είδη υπολογιστικών πόρων, όπως είναι ο χρόνος και η μνήμη που απαιτείται για την εκτέλεσή του.

Η ανάλυση ενός αλγορίθμου είναι η εκτίμηση του πλήθους των υπολογιστικών πόρων που απαιτεί η εκτέλεση του αλγορίθμου.

Η Θεωρία Υπολογισιμότητας ερευνά αν και πόσο αποδοτικά κάποια προβλήματα μπορούν να επιλυθούν με συγκεκριμένα υπολογιστικά μοντέλα.

Η Θεωρία Πολυπλοκότητας μελετά τους πόρους που απαιτούνται για την επίλυση ενός προβλήματος βάσει ενός συγκεκριμένου αλγορίθμου.

Για παράδειγμα, ο αλγόριθμος Ευκλείδης χρησιμοποιεί τόση μνήμη όση χρειάζεται για να αποθηκευτούν οι τρεις ακέραιοι x, y, z. Ωστόσο, η εύρεση του πλήθους των εντολών που εκτελούνται δεν είναι τόσο τετριμμένη. Κάθε φορά που πραγματοποιείται η επανάληψη εκτελούνται τέσσερα βήματα (έλεγχος της συνθήκης Όσο...επανάλαβε, υπολογισμός του z και δύο εντολές εκχώρησης). Επομένως, απαιτούνται 4×α βήματα, όπου α είναι ο

αριθμός των επαναλήψεων. Πόσες, όμως, θα είναι οι επαναλήψεις;

Ο Lame απέδειξε ότι ο αριθμός των βημάτων του ευκλείδειου αλγορίθμου για δύο αριθμούς είναι πάντα μικρότερος ή ίσος του πενταπλασίου των ψηφίων του μεγαλύτερου από τους δύο αριθμούς.

$s \geq 4{,}785 \times \log_{10} n + 1{,}6723$

Το ότι για $x = 27$ και $y = 78$ γίνονται τέσσερις επαναλήψεις, δεν είναι πολύ διαφωτιστικό. Θα ήταν χρήσιμο να μπορεί να υπολογιστεί ο αριθμός των επαναλήψεων για κάθε ζευγάρι ακεραίων x, y.

Έχει υπολογιστεί, ότι ο αριθμός των επαναλήψεων είναι περίπου ίσος με τον λογάριθμο του x (ή του y). Δηλαδή, ο ευκλείδειος αλγόριθμος για να υπολογίσει το ΜΚΔ των x και y, κάνει 4×logx βήματα. Συνήθως παραλείπονται σταθερές, όπως το 4 και έτσι προκύπτει ότι η πολυπλοκότητα του αλγόριθμου είναι O(logx) (διαβάζεται «η πολυπλοκότητα του αλγορίθμου είναι της τάξης logx»).

Η πολυπλοκότητα ενός αλγορίθμου δίνει ένα μέτρο της χρονικής καθυστέρησης του αλγορίθμου για την επίλυση ενός προβλήματος.

Οι περισσότεροι αλγόριθμοι έχουν πολυπλοκότητα χρόνου που ανήκει σε μια από τις κατηγορίες που φαίνονται στον Πίνακα 2.2. Με n συμβολίζεται το μέγεθος του προβλήματος και εξαρτάται από το πρόβλημα. Για παράδειγμα, αν ζητείται να ταξινομηθούν k αριθμοί, τότε το μέγεθος του προβλήματος είναι k, επομένως n = k.

Ο συμβολισμός Ο (O-notation), όπου το Ο είναι το αρχικό γράμμα της αγγλικής λέξης order και διαβάζεται «όμικρον κεφαλαίο», χρησιμοποιείται για να αναδείξει τη γενική συμπεριφορά (την τάξη) του αλγορίθμου.

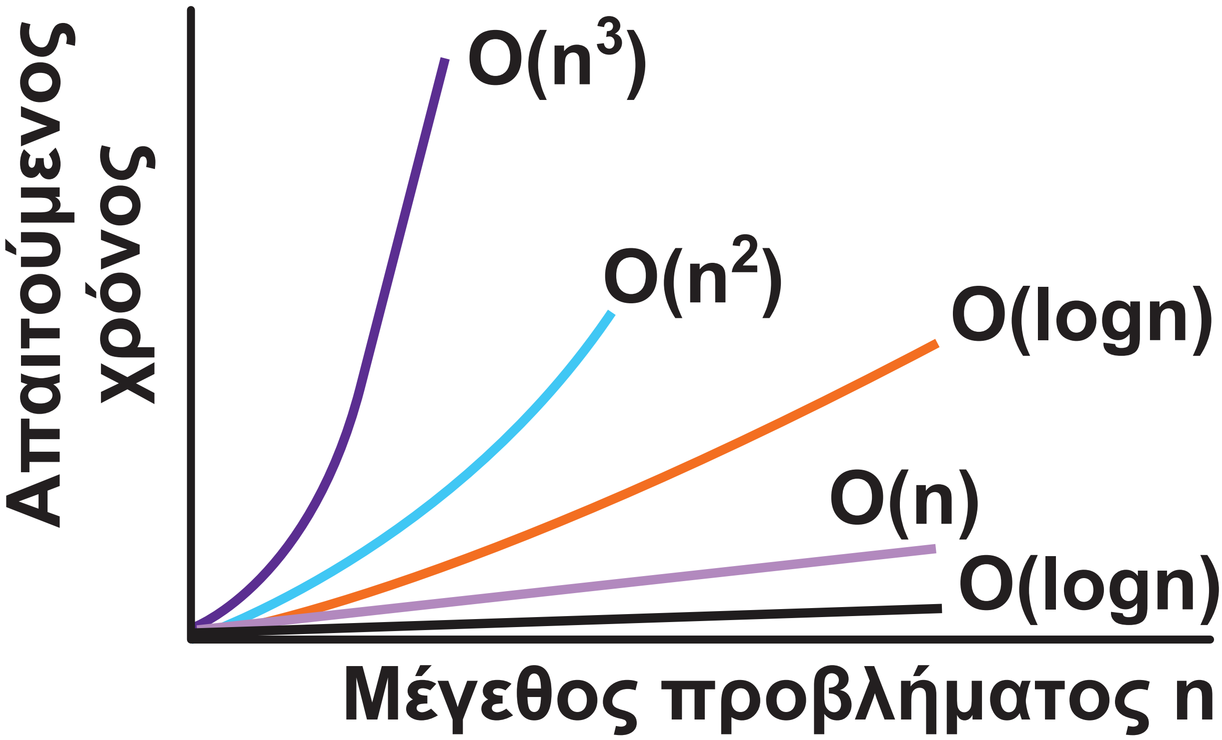
Πίνακας 2.2:

Πολυπλοκότητα	Ονομασία
$O(1)$	Σταθερή
$O(\log n)$	Λογαριθμική
$O(n)$	Γραμμική
$O(n\log n)$	
$O(n^2)$	Τετραγωνική
$O(n^3)$	Κυβική
$O(2^n)$	Εκθετική

Κατηγορίες πολυπλοκότητας αλγορίθμων

	Παρατηρήσεις
	Κάθε βήμα εκτελείται μια φορά ή το πολύ μερικές φορές
	Αν δεν σημειώνεται αλλιώς, οι λογάριθμοι είναι δυαδικοί (Δυαδική Αναζήτηση)
	Σειριακή αναζήτηση
	Γρήγορη Ταξινόμηση
	Ταξινόμηση με επιλογή
	Πολλαπλασιασμός πινάκων
	Μερισμός συνόλου

Στην εικόνα 2.10 έχουν παρασταθεί οι καμπύλες των κυριότερων συναρτήσεων πολυπλοκότητας, ενώ στον Πίνακα 2.3 παρουσιάζονται οι χρόνοι εκτέλεσης για διάφορες πολυπλοκότητες και μεγέθη προβλημάτων.



Εικόνα 2.10: Καμπύλες των κυριότερων συναρτήσεων πολυπλοκότητας.

Υπολογισιμότητα:
Τι μπορεί να υπολογιστεί; Μπορεί ένας υπολογιστής να λύσει οποιοδήποτε πρόβλημα δοθέντος αρκετού χρόνου και χώρου;

Πολυπλοκότητα
Πόσο γρήγορα μπορεί να λυθεί ένα πρόβλημα; Πόσος χώρος (μνήμη) χρειάζεται για να λυθεί ένα πρόβλημα;

Γενικά, τα δεδομένα συνιστούν το μέγεθος της εισόδου ενός αλγορίθμου. Για παράδειγμα στην ταξινόμηση, το πλήθος των αντικειμένων που θα ταξινομηθούν δίνει το μέγεθος του προβλήματος.

Πίνακας 2.3. Χρόνοι εκτέλεσης για διάφορες πολυπλοκότητες και

Τάξη αλγορίθμου	Μέγεθος
	4
O(logn)	6×10^{-10} sec
O(nlogn)	2×10^{-8} sec
O(n)	4×10^{-8} sec
$O(n^2)$	2×10^{-8} sec
$O(n^3)$	64×10^{-8} sec
$O(2^n)$	16×10^{-8} sec
O(n!)	24×10^{-8} sec

μεγέθη προβλημάτων	
Προβλήματος	
16	64
$1,6\times10^{-8}$	$1,8\times10^{-8}$
19×10^{-8}	$1,2\times10^{-6}$
16×10^{-8}	64×10^{-8}
$2,6\times10^{-6}$	4×10^{-5}
4×10^{-5}	3×10^{-3}
65×10^{-5}	5×10^{3} χρόνια
58 ώρες	4×10^{73} χρόνια

Η συνάρτηση O καλείται πολυωνυμική αν είναι έκφραση πολυωνύμου ως προς n, διαφορετικά καλείται μη-πολυωνυμική, αν περιέχει όρους όπως 2^n, n! κτλ. Στην πράξη οι τρεις τελευταίες πολυπλοκότητες χρησιμοποιούνται μόνο για προβλήματα μικρού μεγέθους.

Μετά την επίλυση ενός προβλήματος με ένα αλγόριθμο γίνεται προσπάθεια να σχεδιαστεί ένας νέος αλγόριθμος με μεγαλύτερη ταχύτητα εύρεσης της λύσης του προβλήματος. Για παράδειγμα ένας αλγόριθμος ταξινόμησης με συγκρίσεις (ο οποίος περιγράφεται σε επόμενο κεφάλαιο) απαιτεί $O(n^2)$ συγκρίσεις. Υπάρχουν όμως ταχύτεροι αλγόριθμοι, όπως ο αλγόριθμος γρήγορης ταξινόμησης (Quicksort), που απαιτεί $O(n\log n)$ συγκρίσεις.

Η μελέτη αυτού του ζητήματος γίνεται στο πλαίσιο της **Θεωρίας Πολυπλοκότητας**. Η έρευνα στην περιοχή αυτή προσπαθεί να βρει τους εγγενείς περιορισμούς στην ταχύτητα των αλγορίθμων για τη λύση συγκεκριμένων προβλημάτων. Έτσι αποδεικνύεται, για παράδειγμα, ότι δεν είναι δυνατόν να ταξινομηθούν n ακέραιοι με λιγότερο από nlogn συγκρίσεις και συνεπώς ο αλγόριθμος γρήγορης ταξινόμησης είναι ουσιαστικά βέλτιστος, όσον αφορά την ταχύτητα ταξινόμησης.

Για το πρόβλημα της ταξινόμησης ενός πίνακα δεν έχει βρεθεί μέχρι σήμερα ταχύτερος αλγόριθμος από τον αλγόριθμο γρήγορης ταξινόμησης, ενώ για το πρόβλημα του υπολογισμού του ΜΚΔ δεν έχει βρεθεί ταχύτερος αλγόριθμος από εκείνον του Ευκλείδη. Αυτό οδηγεί, μοιραία, στο ερώτημα: Μήπως δεν υπάρχει καλύτερος αλγόριθμος, μήπως δηλαδή τα προβλήματα αυτά έχουν μία εγγενή πολυπλοκότητα;

2.2.4 Βασικοί τύποι αλγορίθμων

Ο ορισμός του αλγορίθμου που δόθηκε στην αρχή αυτού του κεφαλαίου, συμφωνεί με τη φιλοσοφία των περισσότερων υπολογιστών σήμερα, που διαθέτουν μία Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ) στην οποία οι εντολές εκτελούνται με σειρά, η μία μετά την άλλη. Για το λόγο αυτό ονομάζονται σειριακοί αλγόριθμοι. Όμως η ύπαρξη προβλημάτων στα οποία απαιτείται πολύ μεγάλος χρόνος για τον υπολογισμό της λύσης ενός προβλήματος, δημιούργησε την ανάγκη εύρεσης αλγορίθμων, όπου ορισμένα ή μία σειρά από βήματα αυτών των αλγορίθμων θα μπορούσαν να εκτελούνται παράλληλα (ταυτόχρονα). Σε αυτή την περίπτωση, η εκτέλεση του ενός βήματος

δεν εξαρτάται από την ολοκλήρωση της εκτέλεσης του προηγούμενου. Αλγόριθμοι αυτής της μορφής ονομάζονται **παράλληλοι** αλγόριθμοι και η υλοποίησή τους γίνεται με την ύπαρξη πολλαπλών ΚΜΕ στο σύστημα του υπολογιστή.

Σειριακοί λέγονται οι αλγόριθμοι που χρησιμοποιούν μία κεντρική μονάδα επεξεργασίας και οι εντολές τους εκτελούνται σε σειρά η μία μετά την άλλη.

Παράλληλοι χαρακτηρίζονται οι αλγόριθμοι που χρησιμοποιούν πολλαπλές κεντρικές μονάδες επεξεργασίας όπου ορισμένες ή μία σειρά από εντολές εκτελούνται παράλληλα (ταυτόχρονα).

Παράδειγμα 2.5. Έστω ότι υπάρχει ένας πίνακας που έχει ως περιεχόμενο τους αριθμούς:

1	2	3	4
6	9	8	3

Ο πίνακας του παραδείγματος 2.5 είναι μονοδιάστατος πίνακας που έχει τέσσερα στοιχεία, στις θέσεις 1, 2, 3, 4. Στη θέση 1 το περιεχόμενο του πίνακα είναι 6, στη θέση 2 το περιεχόμενο του πίνακα είναι 9 κτλ.

Στόχος είναι να τοποθετηθούν οι αριθμοί σε αύξουσα σειρά από το μικρότερο στο μεγαλύτερο (αύξουσα ταξινόμηση). Η διαδικασία ταξινόμησης θα επιχειρηθεί με σειριακή και παράλληλη επεξεργασία.

Απάντηση

➢ **Σειριακά**

Εντοπίζεται το μικρότερο στοιχείο του πίνακα (στην περίπτωση αυτή είναι το 3) και αντιμετατίθεται με το στοιχείο της πρώτης θέσης.

Εντοπίζεται το μικρότερο από τα υπόλοιπα στοιχεία του πίνακα (που είναι το 6) και αντιμετατίθεται με το στοιχείο της δεύτερης θέσης.

Εντοπίζεται το μικρότερο από τα υπόλοιπα στοιχεία του πίνακα (που είναι το 8), το οποίο όμως είναι το τρίτο στοιχείο του πίνακα, οπότε η ταξινόμηση έχει ολοκληρωθεί.

- **Παράλληλα**

Συγκρίνονται ταυτόχρονα με δύο διαφορετικούς επεξεργαστές το 1ο με το 2ο στοιχείο και το 3ο με το 4ο. Αν δεν είναι σωστά διαταγμένα, αντιμετατίθενται.

Συγκρίνονται ταυτόχρονα με δύο διαφορετικούς επεξεργαστές το 1ο με το 3ο στοιχείο και το 2ο με το 4ο. Αν δεν είναι σωστά διαταγμένα, αντιμετατίθενται.

Τώρα το μικρότερο από όλα τα στοιχεία είναι στη σωστή θέση (την

1η) και το μεγαλύτερο επίσης στη σωστή θέση (την 4η). Ωστόσο τα δύο μεσαία στοιχεία δεν είναι βέβαιο ότι είναι σωστά διαταγμένα. Οπότε απαιτείται μια ακόμη σύγκριση αυτών των δύο από έναν επεξεργαστή και ολοκληρώνεται η ταξινόμηση.

Σειριακή επεξεργασία

Αρχικός	6	9	8	3
1η	6	9	8	3
2η	6	9	3	8
3η	3	8	6	9
Τελικός	3	6	8	9

Παράλληλη επεξεργασία

Δεν μπορούν να λυθούν όλα τα προβλήματα κάνοντας χρήση παράλληλου υπολογισμού.
Το σε βάθος σκάψιμο για να ανοιχτεί ένα χαντάκι δεν μπορεί να υλοποιηθεί με παράλληλο αλγόριθμο, αφού το δεύτερο μέτρο δεν είναι προσβάσιμο αν δεν τελειώσει πρώτα το πρώτο μέτρο.

Οι αλγόριθμοι επιλύουν προβλήματα. Υπάρχουν απλά και σύνθετα προβλήματα. Λίγα απλά προβλήματα μπορούν να επιλυθούν με διαδοχική εκτέλεση μερικών βημάτων, αφού τα περισσότερα προβλήματα απαιτούν την εκτέλεση ορισμένων συγκεκριμένων βημάτων πολλές φορές. Αυτοί οι αλγόριθμοι αποκαλούνται **επαναληπτικοί.**

Ενδιαφέρον ζήτημα αποτελεί ο εντοπισμός του καλύτερου τρόπου υποδιαίρεσης των προβλημάτων, για να είναι εφικτή η επεξεργασία τους από πολλούς επεξεργαστές παράλληλα.

Για παράδειγμα, επαναληπτικοί αλγόριθμοι είναι ο ευκλείδειος αλγόριθμος, καθώς και ο αλγόριθμος ταξινόμησης με επιλογή που περιγράφηκε πιο πριν.

Παράδειγμα 2.6. Να αναπτυχθεί αλγόριθμος ο οποίος θα διαβάζει τον αριθμό Ν και θα υπολογίζει και θα εμφανίζει το Ν παραγοντικό (συμβολισμός: Ν!).

Το Ν! ορίζεται ως το γινόμενο των ακέραιων αριθμών 1, 2 έως Ν.
Δηλαδή
$N! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot (N-1) \cdot N$
Αν $N = 5$, το $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$

Ο αλγόριθμος υπολογισμού του Ν! με μια επανάληψη βρίσκει το γινόμενο (βλέπε παράδειγμα 2.20).

Όμως το Ν! μπορεί να οριστεί και με άλλο τρόπο, που αποκαλείται αναδρομικός, ως εξής:

$$\begin{cases} N! = N \cdot (N-1)! \text{ για } N \geq 1 & (1) \\ 0! = 1 & (2) \end{cases}$$

Από τη σχέση (1) φαίνεται ότι το παραγοντικό του Ν ορίζεται χρησιμοποιώντας το παραγοντικό του (Ν – 1). Ο όρος αναδρομικότητα εδώ εκφράζει, ότι για να βρεθεί η τιμή του Ν! πρέπει να βρεθεί η τιμή του (Ν – 1)!, η τιμή του οποίου χρειάζεται την τιμή του (Ν – 2)! κ.ο.κ.

Έτσι το 5! κάνει διαδοχικά:

$$5! = 4!\cdot 5 = 3!\cdot 4\cdot 5 = 2!\cdot 3\cdot 4\cdot 5 =$$
$$= 1!\cdot 2\cdot 3\cdot 4\cdot 5 = 0!\cdot 1\cdot 2\cdot 3\cdot 4\cdot 5 =$$
$$= 1\cdot 1\cdot 2\cdot 3\cdot 4\cdot 5 = 120$$

Φαίνεται ότι ο υπολογισμός του Ν! με αναδρομικό τρόπο είναι πιο πολύπλοκος από τον επαναληπτικό. Ωστόσο σε άλλες περιπτώσεις και ιδίως σε μερικά δύσκολα προβλήματα η αναδρομή διευκολύνει σημαντικά.

Αλγόριθμοι που υλοποιούν μια αναδρομική σχέση, αποκαλούνται **αναδρομικοί** και μερικά παραδείγματα παρουσιάζονται στην παράγραφο 2.2.7.6.

2.2.5 Αναπαράσταση αλγορίθμου

Προκειμένου να επιτευχθεί η «ακριβής περιγραφή» ενός αλγορίθμου, χρησιμοποιείται κάποια γλώσσα που μπορεί να περιγράφει σειρές ενεργειών με τρόπο αυστηρό, χωρίς ασάφειες και διφορούμενα. Τέτοιες γλώσσες είναι οι γλώσσες προγραμματισμού (με την προϋπόθεση ότι η σημασιολογία τους είναι αυστηρά διατυπωμένη), μαθηματικά μοντέλα, κάποιες συμβολικές γλώσσες που χρησιμοποιούν αυστηρά καθορισμένους κανόνες περιγραφής, καθώς και κατάλληλα διαμορφωμένα υποσύνολα των φυσικών (ομιλούμενων) γλωσσών. Η αναπαράσταση των αλγορίθμων μπορεί να πραγματοποιηθεί με:

- **Φυσική γλώσσα** όπου η αναπαράσταση γίνεται με την ομιλούμενη γλώσσα, μέσω της οποίας περιγράφονται τα βήματα επίλυσης του προβλήματος. Ωστόσο, με τη φυσική γλώσσα μπορούν να παρατηρηθούν ασάφειες στις οδηγίες.

Όταν περιγράφεται στην ομιλούμενη γλώσσα ο τρόπος με τον οποίο θα μπορέσει κάποιος να επισκεφθεί ένα μουσείο, τότε ο αλγόριθμος έχει διατυπωθεί με φυσική γλώσσα.

Οι φυσικές γλώσσες, είναι οι γλώσσες που μιλούν οι άνθρωποι, ενώ οι τεχνητές γλώσσες έχουν αναπτυχθεί κυρίως για διευκόλυνση της επικοινωνίας ιδεών.

- **Ψευδοκώδικα ή ψευδογλώσσα** η οποία είναι μια υποθετική γλώσσα για την αναπαράσταση αλγορίθμων με στοιχεία από κάποιες γλώσσες προγραμματισμού, παραλείποντας λεπτομέρειες που δεν είναι ουσιαστικές για την ανθρώπινη κατανόηση του αλγορίθμου.

- **Γλώσσα προγραμματισμού** η οποία είναι μια τεχνητή γλώσσα, που έχει αναπτυχθεί για να δημιουργεί ή να εκφράζει προγράμματα για τον υπολογιστή. Η αναπαράσταση των αλγορίθμων με γλώσσα προγραμματισμού μπορεί να γίνει είτε με οπτικές είτε με κειμενικές γλώσσες προγραμματισμού.

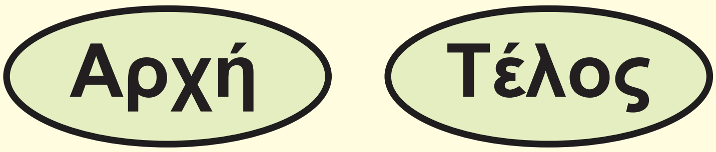
- Στις **οπτικές γλώσσες προγραμματισμού**, η αναπαράσταση των αλγορίθμων γίνεται μέσα από το γραφικό χειρισμό προγραμματιστικών στοιχείων.

- Στις **κειμενικές γλώσσες προγραμματισμού**, η αναπαράσταση των αλγορίθμων γίνεται με τη χρήση σειρών κειμένου που περιλαμβάνουν λέξεις, αριθμούς και σημεία στίξης.

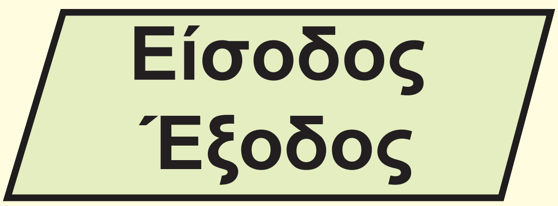
- **Μεθοδολογίες διαγραμματικής αναπαράστασης αλγορίθμων** που συνιστούν έναν γραφικό τρόπο παρουσίασης του αλγόριθμου. Από τις διάφορες μεθοδολογίες διαγραμματικής αναπαράστασης αλγορίθμων που έχουν επινοηθεί η πιο διαδεδομένη είναι το διάγραμμα ροής, όπου η περιγραφή και η αναπαράσταση των αλγορίθμων γίνεται με τη χρήση γεωμετρικών σχημάτων - συμβόλων, όπου το καθένα δηλώνει μια συγκεκριμένη ενέργεια ή λειτουργία.

Τα κυριότερα χρησιμοποιούμενα γεωμετρικά σχήματα - σύμβολα στα διαγράμματα ροής είναι τα ακόλουθα:

- Η έλλειψη, που δηλώνει την αρχή και το τέλος του αλγορίθμου.



- Το πλάγιο παραλληλόγραμμο, που δηλώνει είσοδο ή έξοδο στοιχείων.



- Το ορθογώνιο παραλληλόγραμμο, που δηλώνει την εκτέλεση μιας ή περισσότερων πράξεων.

- Ο ρόμβος, που δηλώνει μία ερώτηση με δύο εξόδους για απάντηση.

- Στα διαγράμματα ροής, εκτός των παραπάνω σχημάτων, χρησιμοποιείται και το βέλος, το οποίο δείχνει τη ροή εκτέλεσης του αλγορίθμου.

Παράδειγμα 2.7. Να αναπτυχθεί αλγόριθμος με φυσική γλώσσα, με διάγραμμα ροής και με ψευδογλώσσα, ο οποίος θα διαβάζει τις τιμές δύο μεταβλητών και θα αντιμεταθέτει το περιεχόμενό τους. Στη συνέχεια θα εμφανίζει ως αποτέλεσμα το περιεχόμενο των μεταβλητών μετά την αντιμετάθεση.

Να εκτελεστεί ο αλγόριθμος για τις τιμές 8 και 12.

Απάντηση

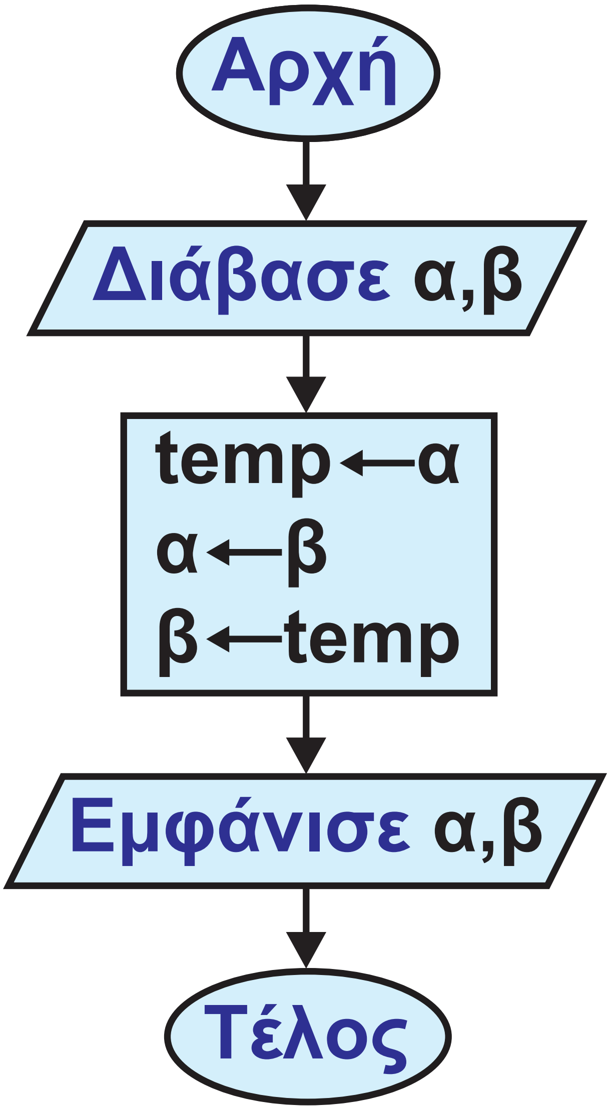
Φυσική γλώσσα: Αφού εισαχθούν οι τιμές δύο μεταβλητών α και β, να δώσετε το περιεχόμενο της μεταβλητής α και σε μία νέα μεταβλητή temp (προσωρινή). Στη συνέχεια, να δώσετε το περιεχόμενο της μεταβλητής β στη μεταβλητή α και τέλος να δώσετε το περιεχόμενο της μεταβλητής temp και στη μεταβλητή β.

Ψευδογλώσσα

1. **Αλγόριθμος** Αντιμετάθεση
2. **Διάβασε** α, β
3. temp ← α
4. α ← β
5. β ← temp
6. **Εμφάνισε** α, β
7. **Τέλος** Αντιμετάθεση

Αρ. Εντ.	α	β	temp	Έξοδος
2	8	12		
3			8	
4	12			
5		8		
6				12 8

Διάγραμμα ροής



Ο αλγόριθμος Αντιμετάθεση αντιμεταθέτει το περιεχόμενο των δύο μεταβλητών και παρουσιάζεται με τρεις τρόπους αναπαράστασης.

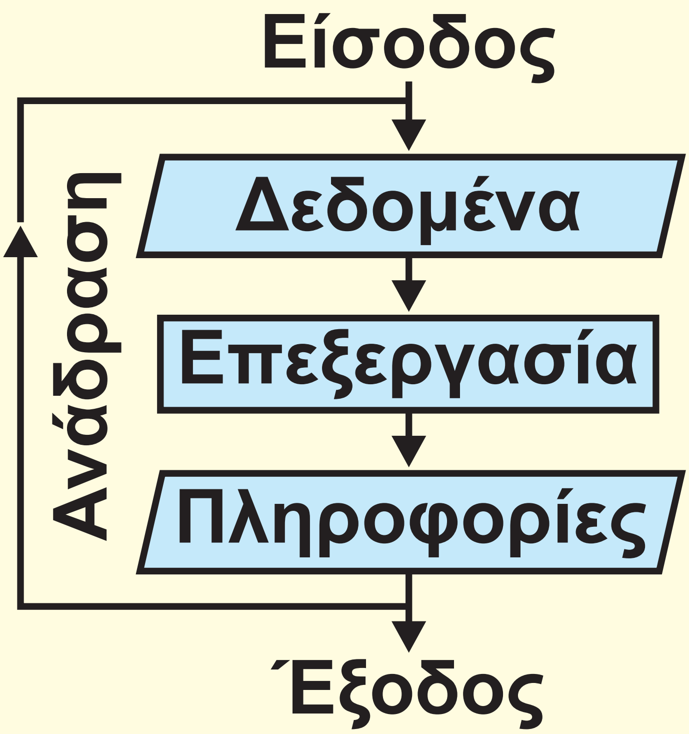
Στην ψευδογλώσσα έχουν καταγραφεί οι ενέργειες που περιγράφονται στο πλαίσιο της φυσικής γλώσσας σε κωδικοποιημένη μορφή, ενώ τέλος έχει δημιουργηθεί και το αντίστοιχο διάγραμμα ροής.

2.2.6 Δεδομένα και αναπαράσταση τους

Ένας αλγόριθμος λαμβάνει κάποια δεδομένα από την είσοδο, τα επεξεργάζεται μέσα από μια σειρά βημάτων και δίνει ως έξοδο τα αποτελέσματα.

Η επεξεργασία δεδομένων, η οποία στην πράξη πραγματοποιείται μέσω αλγορίθμων, αναφέρεται στην εκτέλεση διαφόρων πράξεων/ λειτουργιών πάνω στα δεδομένα. Το αποτέλεσμα της επεξεργασίας δεδομένων είναι η πληροφορία. Έτσι, θα μπορούσε κανείς γενικεύοντας να πει ότι ένας αλγόριθμος μετατρέπει τα δεδομένα σε πληροφορία. Για παράδειγμα, ένας τηλεφωνικός αριθμός και ένα ονοματεπώνυμο

αποτελούν δεδομένα. Δεν παρέχουν όμως καμία πληροφορία. Πληροφορία παράγεται μόνο αν σχετισθούν μεταξύ τους, ότι δηλαδή κάποιο τηλέφωνο ανήκει σε κάποιο συγκεκριμένο συνδρομητή. Με βάση τις πληροφορίες λαμβάνονται αποφάσεις και γίνονται διάφορες ενέργειες. Στη συνέχεια οι ενέργειες αυτές παράγουν νέα δεδομένα, αυτά νέες πληροφορίες, οι τελευταίες νέες αποφάσεις και ενέργειες κ.ο.κ. όπως φαίνεται στην εικόνα 2.11. Η διεργασία αυτή μπορεί να επαναλαμβάνεται, οι πληροφορίες που παρήχθησαν, να επανατροφοδοτούν μέσω ανάδρασης την είσοδο για επανάληψη του κύκλου κ.ο.κ..



Εικόνα 2.11. Σχέση δεδομένων και πληροφοριών

Η θεωρία αλγορίθμων, μελετά τα δεδομένα κυρίως από τις σκοπιές του υλικού και των γλωσσσών προγραμματισμού.

Όσον αφορά τις γλώσσες προγραμματισμού, κάθε γλώσσα μπορεί να υποστηρίζει τη χρήση διαφόρων τύπων δεδομένων. Κάθε γλώσσα έχει συγκεκριμένους τύπους δεδομένων, ενώ μπορούν να δημιουργηθούν νέοι τύποι ορισμένοι από το χρήστη. Οι πιο συνήθεις τύποι δεδομένων είναι οι ακόλουθοι (εικόνα 2.12):

Ακέραιος τύπος: για την αναπαράσταση ακεραίων αριθμών.
Πραγματικός τύπος: για την αναπαράσταση πραγματικών αριθμών.
Λογικός τύπος: για την αναπαράσταση λογικών δεδομένων.
Αλφαριθμητικός τύπος: για την αναπαράσταση αλφαριθμητικών δεδομένων.

Εικόνα 2.12. Τύποι Δεδομένων

Σε κάθε τύπο δεδομένων μπορούν να εφαρμοστούν διαφορετικές πράξεις. Επομένως, κατά τον σχεδιασμό ενός αλγορίθμου έχει σημασία το είδος των τύπων δεδομένων που υποστηρίζονται.

Το υλικό επιτρέπει την αποθήκευση των δεδομένων ενός προγράμματος

στην κύρια μνήμη ή και στις περιφερειακές συσκευές ενός υπολογιστή με διάφορες μορφές. Με σκοπό την πρόσβαση στα δεδομένα που βρίσκονται στη βοηθητική μνήμη είναι πιθανό να χρησιμοποιούνται διαφορετικοί αλγόριθμοι για την επεξεργασία τους. Επομένως, το υλικό του υπολογιστή έχει επίδραση στο είδος των αλγορίθμων που θα χρησιμοποιηθούν.

Τα δεδομένα μπορεί να είναι απλές μεταβλητές, οι οποίες λαμβάνουν μία τιμή κάθε φορά (απλά δεδομένα) ή μπορούν να αποθηκεύονται ως μία δομή δεδομένων.

Ο σκληρός δίσκος και η μνήμη flash, αποτελούν παραδείγματα βοηθητικής μνήμης του υπολογιστή.

Δομή δεδομένων (data structure) είναι ένα σύνολο αποθηκευμένων δεδομένων, τα οποία είναι έτσι οργανωμένα, ώστε να υπόκεινται σε συγκεκριμένες απαιτούμενες επεξεργασίες.

Ο όρος αναφέρεται σε ένα σύνολο δεδομένων μαζί με ένα σύνολο λειτουργιών που επιτρέπονται στα δεδομένα αυτά. Οι δομές δεδομένων είναι πολύ στενά συνδεδεμένες με την έννοια του αλγορίθμου. Είναι πολύ χαρακτηριστική η ακόλουθη «σχέση» που διατύπωσε ο Νικλάους Βιρθ (Niklaus Wirth), δημιουργός της γλώσσας Pascal:

Αλγόριθμοι + Δομές Δεδομένων =
= Προγράμματα

Η ανωτέρω σχέση έχει την έννοια ότι αν κάποιος διαθέτει τον κατάλληλο αλγόριθμο και τις δομές δεδομένων, οι οποίες θα χρησιμοποιηθούν, είναι εντελώς άμεση η μετατροπή και υλοποίησή του σε πρόγραμμα σε γλώσσα υπολογιστή. Κάθε δομή δεδομένων αποτελείται, στην πιο γενική της μορφή, από ένα σύνολο στοιχείων ή κόμβων.

Οι πιο ευρέως χρησιμοποιούμενες δομές δεδομένων είναι ο πίνακας, η στοίβα, η ουρά, η λίστα, το δένδρο και ο γράφος.

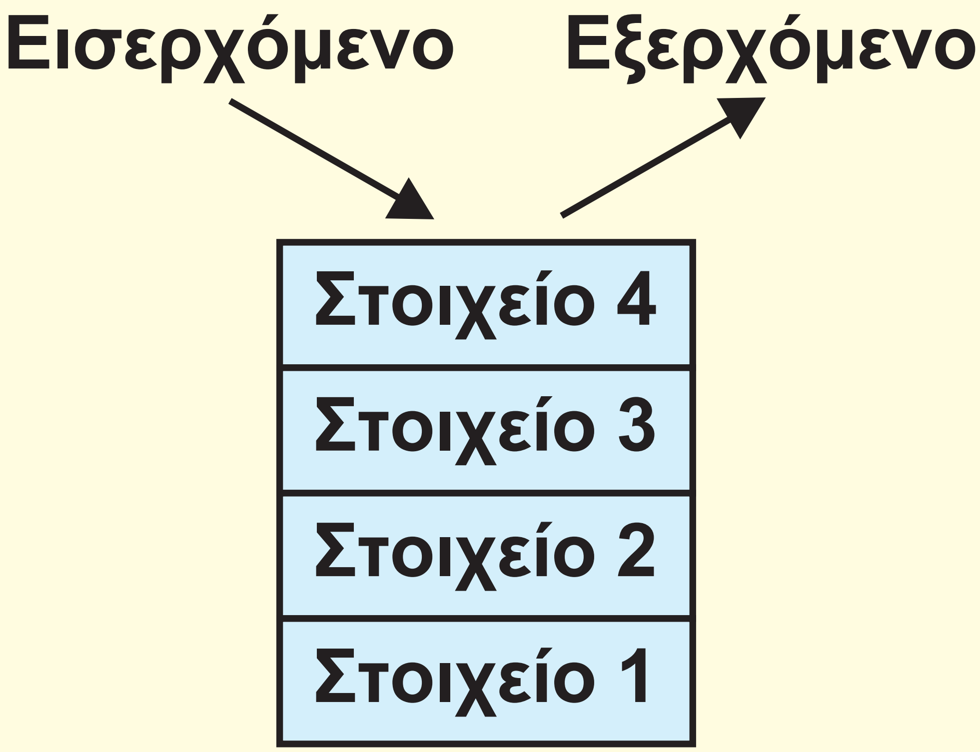
Ο **πίνακας** (table) αποτελείται από ένα σύνολο ομοειδών απλών στοιχείων, καθένα από τα οποία καθορίζεται με τη βοήθεια ενός ή περισσοτέρων δεικτών. Ένας πίνακας μπορεί να είναι μίας, δύο ή περισσοτέρων διαστάσεων, ανάλογα με το πλήθος δεικτών που χρειάζονται για να καθοριστεί η θέση του. Στην εικόνα 2.13, παρουσιάζεται ένας δισδιάστατος πίνακας Α με 4 γραμμές και 3 στήλες. Στο παράδειγμα 2.5 είχε παρουσιαστεί ένας μονοδιάστατος πίνακας.

Δισδιάστατος Πίνακας Α 4x3		
Στοιχείο 1,1	Στοιχείο 1,2	Στοιχείο 1,3
Στοιχείο 2,1	Στοιχείο 2,2	Στοιχείο 2,3
Στοιχείο 3,1	Στοιχείο 3,2	Στοιχείο 3,3
Στοιχείο 4,1	Στοιχείο 4,2	Στοιχείο 4,3

Εικόνα 2.13. Δισδιάστατος πίνακας

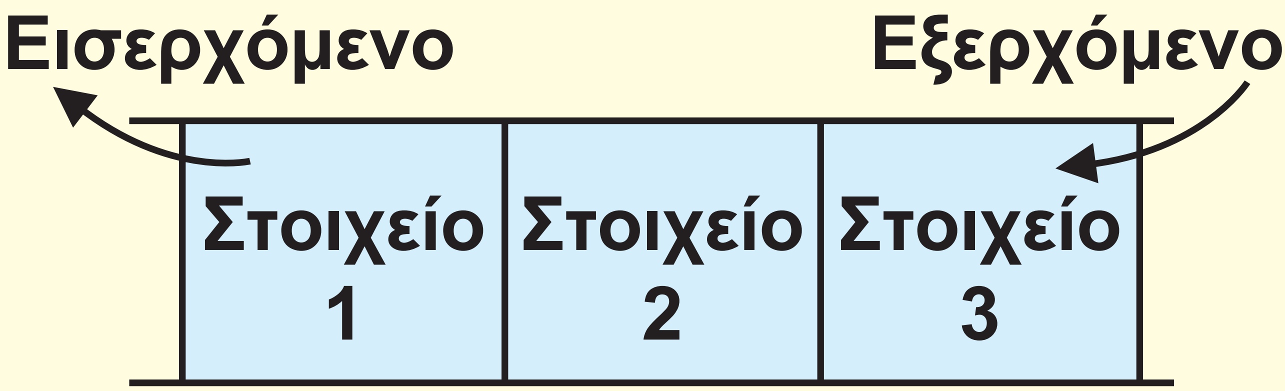
Μία **στοίβα** (stack) είναι μια γραμμική διάταξη στοιχείων, στην οποία εισάγονται και εξάγονται στοιχεία μόνο από το ένα άκρο (εικόνα 2.14). Η λειτουργία της εισαγωγής αποκαλείται **ώθηση** (push) και της εξαγωγής **απώθηση** (pull ή pop). Η φιλοσοφία εισαγωγής και εξαγωγής των

στοιχείων ονομάζεται LIFO (Last In, First Out), δηλαδή το τελευταίο εισαγόμενο δεδομένο εξέρχεται και πρώτο.



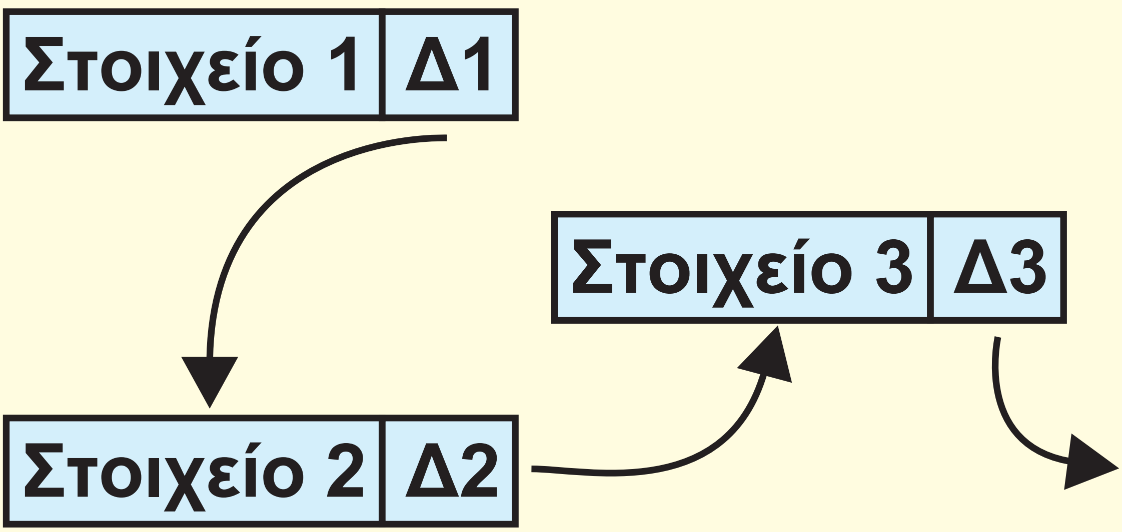
Εικόνα 2.14. Στοίβα

Μια **ουρά** (queue) αποτελεί μια γραμμική διάταξη στοιχείων, στην οποία εισάγονται νέα στοιχεία από ένα άκρο και εξάγονται υπάρχοντα στοιχεία από το άλλο άκρο (εικόνα 2.15). Η λειτουργία της ουράς αποκαλείται **FIFO** (First In, First Out), δηλαδή το στοιχείο το οποίο εισάγεται πρώτο στην ουρά εξέρχεται και πρώτο.



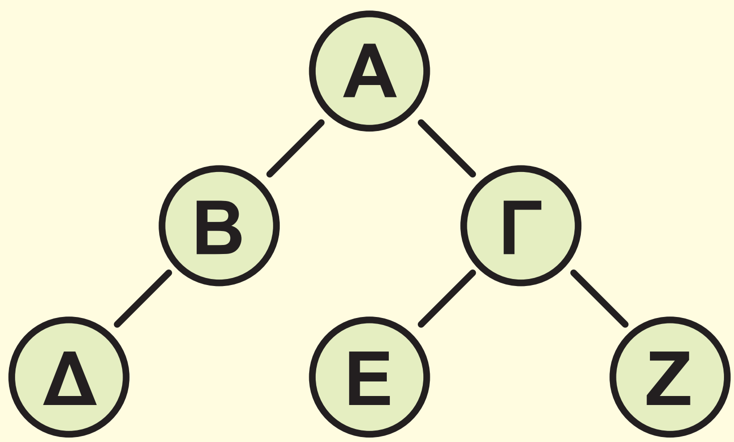
Εικόνα 2.15. Ουρά

Σε μια (συνδεσμική) **λίστα** (linked list) τα στοιχεία φαίνονται «λογικά» ότι είναι γραμμικά διατεταγμένα, χωρίς όμως αυτό να σημαίνει ότι βρίσκονται σε συνεχόμενες θέσεις της μνήμης του υπολογιστή (εικόνα 2.16). Ανεξάρτητα από τη θέση που καταλαμβάνει στη μνήμη ένα δεδομένο, συσχετίζεται με το επόμενό του με τη βοήθεια κάποιου **δείκτη** (pointer).



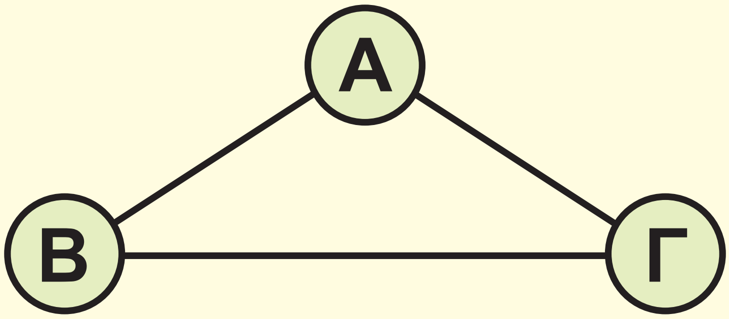
Εικόνα 2.16. Λίστα

Το **δένδρο** (tree) είναι μη γραμμική δομή που αποτελείται από ένα σύνολο κόμβων, οι οποίοι συνδέονται με ακμές (εικόνα 2.17). Υπάρχει μόνο ένας κόμβος, από τον οποίο μόνο ξεκινούν ακμές, που ονομάζεται **ρίζα** (root). Σε όλους τους άλλους κόμβους καταλήγει μία ακμή και ξεκινούν καμία, μία ή περισσότερες. Οι κόμβοι στους οποίους καταλήγουν μόνο ακμές, ονομάζονται **φύλλα**.



Εικόνα 2.17. Δένδρο

Ο **γράφος** (graph) αποτελεί τη πιο γενική δομή δεδομένων μια και αποτελείται από κόμβους και ακμές χωρίς όμως κάποια ιεράρχηση.



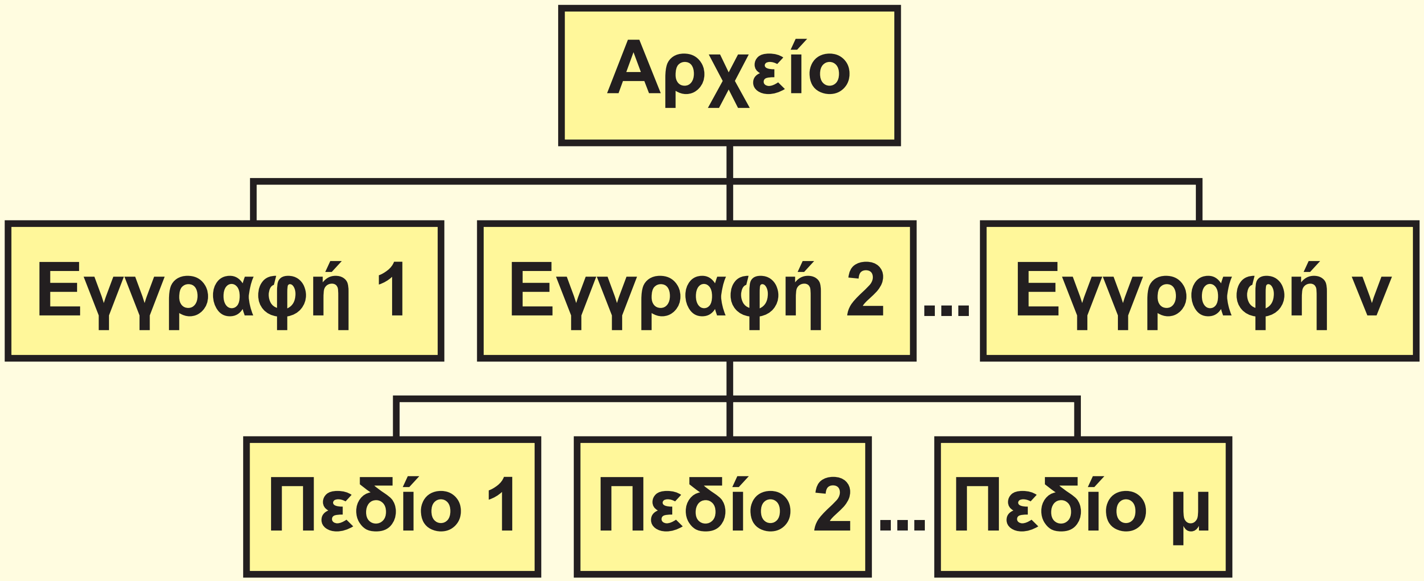
Εικόνα 2.18. Γράφος

Υπάρχουν διάφοροι τρόποι διάκρισης των δομών δεδομένων. Διακρίνονται σε **στατικές** και **δυναμικές**. Οι στατικές δομές έχουν σταθερό μέγεθος και μπορούν να κατακρατήσουν συγκεκριμένο πλήθος στοιχείων. Αντίθετα οι δυναμικές δομές δεν έχουν σταθερό μέγεθος και το πλήθος των στοιχείων τους μπορεί να μεγαλώνει ή να μικραίνει καθώς στη δομή εισάγονται νέα δεδομένα ή διαγράφονται άλλα.

Για παράδειγμα, η εγγραφή ενός μαθητή μπορεί να αποτελείται από το όνομα, το επώνυμο, τον αριθμό κινητού τηλεφώνου, τη διεύθυνση αλληλογραφίας, την ηλεκτρονική διεύθυνση, τη φωτογραφία του κ.ά., που καλούνται πεδία της εγγραφής.

Οι δομές δεδομένων διακρίνονται επίσης σε **γραμμικές** και **μη γραμμικές**. Στις γραμμικές δομές μπορεί να ορισθεί κάποια σχέση διάταξης για δύο οποιαδήποτε διαδοχικά στοιχεία τους. Αυτό σημαίνει ότι κάποιο στοιχείο θα είναι πρώτο και κάποιο τελευταίο. Οποιοδήποτε από τα υπόλοιπα θα έπεται από το προηγούμενό του και θα προηγείται από το επόμενό του. Στις μη γραμμικές δομές δεν μπορεί να οριστεί μια σχέση διάταξης όπως η παραπάνω. Τέτοιες δομές είναι τα δένδρα και οι γράφοι. Στα δένδρα ένας κόμβος έχει έναν προηγούμενο αλλά πιθανόν πολλούς επόμενους. Στους γράφους κάθε κόμβος μπορεί να έχει πολλούς προηγούμενους και πολλούς επόμενους.

Τέλος διάκριση των δομών μπορεί να γίνει και ανάλογα με το είδος της χρησιμοποιούμενης μνήμης (κύρια ή βοηθητική). Οι δομές δεδομένων βοηθητικής μνήμης αποκαλούνται **αρχεία δεδομένων** (data files). Ένα αρχείο απαρτίζεται από έναν αριθμό ομοειδών εγγραφών (records). Κάθε εγγραφή διαθέτει ορισμένα πεδία (fields), που περιέχουν δεδομένα για μια οντότητα (π.χ. μαθητής), όπως φαίνεται στην εικόνα 2.19.



Εικόνα 2.19. Δομή αρχείου

Περιεχόμενα