

κεφάλαιο

1

Βασικές Έννοιες

Επιστήμη των Υπολογιστών



Στόχοι

Στόχος του κεφαλαίου είναι οι μαθητές:

- να γνωρίσουν βασικές έννοιες και τομείς της Επιστήμης των Υπολογιστών.



Λέξεις κλειδιά

**Επιστήμη των
Υπολογιστών**

Πληροφορική, Θεωρητική Επιστήμη των Υπολογιστών,
Εφαρμοσμένη Επιστήμη των Υπολογιστών



Εισαγωγικές ερωτήσεις

1. Γνωρίζετε τι είναι Επιστήμη και τι είναι Τεχνολογία; Πιστεύετε ότι η Πληροφορική είναι Επιστήμη ή Τεχνολογία;
2. Γνωρίζετε κλάδους της Επιστήμης των Υπολογιστών;

1. Βασικές έννοιες

1.1 Πληροφορική ή Επιστήμη Υπολογιστών;

Ο όρος «πληροφορική» προέρχεται από τη λέξη πληροφορία. Εμφανίστηκε τη δεκαετία του 1960 στη Γαλλία (Informatique) και τη Γερμανία (Informatik). Από την ετυμολογία της λέξης προκύπτει ο ορισμός:

Πληροφορική είναι η επιστήμη που ασχολείται με την αναπαράσταση, αποθήκευση και επεξεργασία της πληροφορίας.



Η Πληροφορική είναι άρρηκτα συνδεδεμένη με τη μηχανή που λέγεται ηλεκτρονικός υπολογιστής. Γι' αυτό και στον Αγγλόφωνο χώρο, έχει επικρατήσει η ονομασία Επιστήμη Υπολογιστών (Computer Science). Στη νέα επιστήμη όσο σημαντικός είναι ο υπολογιστής (computer), άλλο τόσο σημαντικές είναι οι διαδικασίες υπολογισμού (computing) που μπορεί να κάνει.

Επιστήμη Υπολογιστών είναι η επιστήμη που ασχολείται με τους υπολογιστές και τους υπολογισμούς.



Για μας, στο εξής, Πληροφορική και Επιστήμη Υπολογιστών αναφέρονται στην ίδια επιστήμη. Θεμελιώδεις έννοιες στην Επιστήμη των Υπολογιστών είναι:

- η ίδια η μηχανή, ο υπολογιστής και
- η υπολογιστική διαδικασία που η μηχανή μπορεί να επιτελέσει, το πρόγραμμα.

Ιστορικοί σταθμοί της Επιστήμης Υπολογιστών

300 π.Χ.	Αλγόριθμος του Ευκλείδη για τον υπολογισμό του μέγιστου κοινού διαιρέτη δύο φυσικών αριθμών, ένας από τους αρχαιότερους αλγόριθμους
100 π.Χ.	Μηχανισμός των Αντικυθήρων, μηχανικός υπολογιστής και όργανο αστρονομικών παρατηρήσεων
820 μ.Χ.	Ο al-Khwarizmi, Πέρσης μαθηματικός και αστρονόμος γράφει βιβλίο για τους αλγόριθμους
1822	Ο Charles Babbage σχεδιάζει την Αναλυτική Μηχανή του, την πρώτη μηχανή με αποθηκευμένα προγράμματα.
1940	Ο John von Neumann θέτει τις θεμελιώδεις αρχές σχεδίασης των σύγχρονων ΗΥ, γνωστές ως “αρχιτεκτονική von Neumann”, στο Πανεπιστήμιο του Πρίνστον.
1944	Ο Aiken κατασκευάζει τον Mark I. Χρόνος για πρόσθεση: 1/3 s., για πολλαπλασιασμό: 6 s
1946	Οι Eckert και Maughly κατασκευάζουν τον ENIAC, τον πρώτο ηλεκτρονικό υπολογιστή με 18000 λυχνίες. Χρόνος για πολλαπλασιασμό: 6 ms
1949	Ο Wilkes κατασκευάζει τον EDSAC, τον πρώτο ψηφιακό υπολογιστή γενικής χρήσης με αποθηκευμένο πρόγραμμα (υπολογιστής τύπου von Neumann)
1969	Γέννηση του διαδικτύου με τη δημιουργία του δικτύου ARPANET
1991	Δημιουργία του Παγκόσμιου Ιστού (WWW) στο CERN

Η παραπάνω ιστορική αναδρομή εμφανίζει κυρίως τεχνολογικά επιτεύγματα και όχι επιστημονικά. Μήπως, λοιπόν, η Πληροφορική είναι απλώς η τεχνολογία των υπολογιστών και δεν είναι επιστήμη; Η απάντηση είναι ότι η Πληροφορική είναι δισδιάστατη, είναι και τεχνολογία και επιστήμη. Απλά, ο ηλεκτρονικός υπολογιστής καταλαμβάνει τόσο κυρίαρχο ρόλο σε αυτήν που δεν είναι δυνατό να διαχωρίσουμε τα τεχνολογικά από τα επιστημονικά επιτεύγματα.

Μια από τις δόκιμες διακρίσεις της Επιστήμης Υπολογιστών διακρίνει τη Θεωρητική από την Εφαρμοσμένη Επιστήμη. Στον παρακάτω πίνακα φαίνονται κύριοι τομείς της Επιστήμης Υπολογιστών με διαφορετική απόχρωση ανάλογα με το αν ανήκει στη Θεωρητική ή Εφαρμοσμένη.

Εικόνα 1-1. Ταξινόμηση της Επιστήμης Υπολογιστών από την ACM (Association for Computing Machinery)

Επιστήμη Υπολογιστών (Computer Science)	
Λογισμικό και μηχανική λογισμικού (Software and its engineering)	Γλώσσες προγραμματισμού, προγραμματιστικά υποδείγματα, Λειτουργικά Συστήματα
Θεωρία υπολογισμού (Theory of computation)	Ανάλυση και σχεδίαση αλγορίθμων, Πολυπλοκότητα και υπολογισιμότητα αλγορίθμων
Μαθηματικά της Πληροφορικής (Mathematics of computing)	Διακριτά μαθηματικά, Θεωρία γράφων, Πιθανότητες και στατιστική, Μαθηματική ανάλυση και μαθηματική βελτιστοποίηση
Υλικό (Hardware)	Ολοκληρωμένα κυκλώματα, μνήμες και μέσα αποθήκευσης
Οργάνωση και Αρχιτεκτονική Η/Υ (Computer organization)	Έλεγχος και δοκιμή υλικού, συστήματα πραγματικού χρόνου
Δίκτυα υπολογιστών (Networks)	Αρχιτεκτονικές, πρωτόκολλα και υπηρεσίες δικτύων
Πληροφοριακά συστήματα (Information systems)	Συστήματα διαχείρισης δεδομένων, βάσεις δεδομένων, ανάκτηση πληροφοριών, συστήματα υποστήριξης αποφάσεων
Ασφάλεια και Ιδιωτικότητα (Security and privacy)	Κρυπτογράφηση, συστήματα ασφάλειας δεδομένων
Ανθρωποκεντρικός υπολογισμός (Human-centered computing)	Διεπαφή χρήστη-υπολογιστή, προσβασιμότητα
Μεθοδολογίες υπολογισμού (Computing methodologies)	Παράλληλοι υπολογιστές, Τεχνητή νοημοσύνη, Ρομποτική, Μηχανική όραση, Γραφική Υπολογιστών, Κατανεμημένα συστήματα
Εφαρμογές της Πληροφορικής σε άλλους τομείς (Applied computing)	Φυσικές επιστήμες, Μηχανική, Ηλεκτρονικό εμπόριο, Επιχειρήσεις, Αστρονομία, Αεροναυπηγική, Χημεία, Φυσική, Βιολογία, Εκπαίδευση, Νομικές και κοινωνικές επιστήμες, Καλές τέχνες

1.2 Θεωρητική Επιστήμη των Υπολογιστών

Θεμελιώδεις έννοιες της Θεωρητικής Επιστήμης των Υπολογιστών είναι ο **αλγόριθμος** και το **πρόγραμμα**. Μια μηχανικά εκτελέσιμη υπολογιστική διαδικασία αποτελεί αλγόριθμο. Το θεωρητικό μέρος της Επιστήμης Υπολογιστών ασχολείται με τη σχεδίαση, τη δημιουργία και την αναπαράσταση των αλγορίθμων.

Η μεταφορά του αλγορίθμου σε μια μορφή κατανοητή από τον ηλεκτρονικό υπολογιστή, δηλαδή σε μια **γλώσσα προγραμματισμού**, δημιουργεί ένα πρόγραμμα. Στη Θεωρητική Επιστήμη των Υπολογιστών εξετάζονται τεχνικές σχεδίασης προγραμμάτων και μελετώνται οι γλώσσες προγραμματισμού. Ένας άλλος τομέας της θεωρητικής Πληροφορικής ασχολείται με τα δεδομένα και τους τρόπους οργάνωσης αυτών, τις **δομές δεδομένων**.

Στο θεωρητικό μέρος της Επιστήμης Υπολογιστών ανήκει επίσης ο τομέας της **θεωρίας υπολογισμού**. Ο τομέας αυτός αποτελεί τη μαθηματική βάση της Επιστήμης των Υπολογιστών που έχει ως αντικείμενο να εκφράσει και να μελετήσει, με τη χρήση μαθηματικών μοντέλων, τη λειτουργία των υπολογιστικών μηχανών, την υπολογισσιμότητα των προβλημάτων (αν μπορούν ή όχι να επιλυθούν από υπολογιστή), καθώς και τους υπολογιστικούς πόρους που απαιτούνται για την επίλυση προβλημάτων.

1.3 Εφαρμοσμένη Επιστήμη των Υπολογιστών

Η άρρηκτη σχέση της θεωρητικής επιστήμης με την υπολογιστική μηχανή δημιουργεί πεδία εφαρμογής τόσο στην ίδια την Πληροφορική όσο και σε άλλες Επιστήμες.

Η Εφαρμοσμένη Επιστήμη των Υπολογιστών ασχολείται με το **υλικό** (hardware) και την **οργάνωση** και **αρχιτεκτονική υπολογιστών**. Ασχολείται επίσης με τα **λειτουργικά συστήματα**, προγράμματα μεγάλης κλίμακας που τοποθετούνται ανάμεσα στο υλικό και το λογισμικό του υπολογιστή. Τα **πληροφοριακά συστήματα** συνδυάζουν υλικό, λογισμικό, δεδομένα, ανθρώπους και διαδικασίες. Η διασύνδεση υπολογιστών σε δίκτυα, δημιουργεί έναν άλλο τομέα. Ο τομέας της **τεχνητής νοημοσύνης** ασχολείται με τη δημιουργία προγραμμάτων που κάνουν τον υπολογιστή να προσομοιάζει σε νοήμον ον.

Τέλος, ένα μέρος της Εφαρμοσμένης Επιστήμης των Υπολογιστών διερευνά τις δυνατότητες εφαρμογής των ηλεκτρονικών υπολογιστών σε άλλες επιστήμες, όπως στη Μηχανική, στην Οικονομία, στην Ιατρική, στις Φυσικές επιστήμες, στις Ανθρωπιστικές επιστήμες και στις Καλές τέχνες. Οι εφαρμογές αυτές, συχνά δημιουργούν νέους επιστημονικούς τομείς όπως η Βιοϊατρική.

Θεωρητική Επιστήμη των Υπολογιστών

Επίλυση Προβλημάτων
Αλγόριθμοι
Ανάλυση Αλγορίθμων
Δομές Δεδομένων
Μέθοδοι Προγραμματισμού
Γλώσσες Προγραμματισμού
Μεταγλωττιστές
Θεωρία Αυτομάτων
Θεωρία Υπολογισμών
Θεωρία Πολυπλοκότητας
Τεχνολογία Λογισμικού

Εφαρμοσμένη Επιστήμη των Υπολογιστών

Υλικό Υπολογιστή
Ολοκληρωμένα κυκλώματα
Μικροεπεξεργαστές
Οργάνωση και αρχιτεκτονική υπολογιστών
Δίκτυα υπολογιστών
Λειτουργικά Συστήματα
Επικοινωνία Ανθρώπου-Μηχανής
Πληροφοριακά συστήματα
Συστήματα διαχείρισης βάσεων δεδομένων
Τεχνητή νοημοσύνη
Εφαρμογές σε άλλες επιστήμες

Καλώς ήρθατε στον κόσμο της Πληροφορικής!



Ερωτήσεις - Θέματα για συζήτηση

A. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Η Πληροφορική και η Επιστήμη Υπολογιστών αναφέρονται στην ίδια επιστήμη.
2. Πληροφορική είναι απλώς η τεχνολογία των υπολογιστών και δεν είναι επιστήμη.
3. Η υπολογισσιμότητα των προβλημάτων αφορά στο αν μπορούν ή όχι να επιλυθούν από υπολογιστή.

B. Συμπληρώστε τις προτάσεις με την κατάλληλη λέξη:

1. Θεμελιώδεις έννοιες στην Επιστήμη των Υπολογιστών είναι η ίδια η μηχανή, ο _____ και η _____ διαδικασία.
2. Η Επιστήμη των Υπολογιστών διακρίνεται στη _____ και στην _____ Επιστήμη των Υπολογιστών.
3. Η θεωρία υπολογισμού αποτελεί _____ τη βάση της επιστήμης των υπολογιστών.
4. Η _____ ερευνά τις δυνατότητες εφαρμογής των ηλεκτρονικών υπολογιστών στην Ιατρική.

Γ. Αντιστοιχίστε κάθε στοιχείο της στήλης A με την κατάλληλη έννοια της στήλης B

Στήλη A	Στήλη B
1. Υλικό Υπολογιστή	Α. Θεωρητική Επιστήμη των Υπολογιστών
2. Ανάλυση Αλγορίθμων	
3. Δομές Δεδομένων	
4. Δίκτυα υπολογιστών	
5. Οργάνωση και αρχιτεκτονική υπολογιστών	
6. Γλώσσες Προγραμματισμού	B. Εφαρμοσμένη Επιστήμη των Υπολογιστών
7. Πληροφοριακά συστήματα	
8. Τεχνητή νοημοσύνη	
9. Θεωρία Υπολογισμών	
10. Αλγόριθμοι	
11. Τεχνολογία Λογισμικού	
12. Λειτουργικά Συστήματα	

κεφάλαιο

2

**Θέματα Θεωρητικής
Επιστήμης
των Υπολογιστών**

Η έννοια του προβλήματος



Στόχοι

Στόχοι του κεφαλαίου είναι οι μαθητές:

- να έρθουν σε επαφή με την έννοια του προβλήματος.
- να μπορούν να διακρίνουν τις κατηγορίες των προβλημάτων με κριτήριο τη δυνατότητα επίλυσής τους.
- να ακολουθούν τα στάδια επίλυσης ενός προβλήματος στην αντιμετώπιση υπολογιστικών προβλημάτων.
- να αναγνωρίζουν τις βασικές μεθοδολογίες επίλυσης προβλημάτων και να τις εφαρμόζουν σε δοσμένα προβλήματα.



Λέξεις κλειδιά

Πρόβλημα

Επιλύσιμα - Μη επιλύσιμα - Ανοικτά προβλήματα, Υπολογιστικά προβλήματα, Προβλήματα Απόφασης, Προβλήματα Βελτιστοποίησης, Κατανόηση - Επίλυση προβλήματος, Δεδομένα, Ζητούμενα, Ορθότητα-Πληρότητα-Σαφήνεια προβλήματος, Μεθοδολογίες επίλυσης προβλημάτων



Εισαγωγικές ερωτήσεις

1. Γνωρίζετε όχι ο “τετραγωνισμός του κύκλου” είναι μη επιλύσιμο πρόβλημα;
2. Τι σας δυσκολεύει περισσότερο στην εκφώνηση ενός προβλήματος;
3. Αλλάζετε τρόπο αντιμετώπισης ενός προβλήματος της καθημερινής σας ζωής ανάλογα με τις συνθήκες που το περιβάλλουν;

2. Η έννοια του προβλήματος

2.1 Το πρόβλημα στην επιστήμη των Η/Υ

Τη λέξη πρόβλημα την έχουμε συναντήσει πολλές φορές. Έχουμε λύσει πολλά προβλήματα από την αρχή της σχολικής μας ζωής σε διάφορα μαθήματα όπως τα Μαθηματικά τη Φυσική, τη Χημεία κ.ά.. Όμως προβλήματα αντιμετωπίζουμε και καθημερινά στη ζωή μας όπως, για παράδειγμα, πώς θα πάω στο σχολείο, τί θα φάω σήμερα, πού θα πάω βόλτα με τους φίλους μου, πώς θα τακτοποιήσω το δωμάτιο μου. Τα προβλήματα αυτά θεωρούνται απλά, υπάρχουν όμως και πιο σύνθετα όπως η ρύπανση του περιβάλλοντος, η αντιμετώπιση φυσικών φαινομένων, η αποκωδικοποίηση του DNA, η επαγγελματική σταδιοδρομία, η ανεργία, η θεραπεία ασθενειών κ. ά.

Όλα τα προβλήματα είναι καταστάσεις που πρέπει να αντιμετωπιστούν αλλά δε γνωρίζουμε το πώς. Προβλήματα προκύπτουν όταν ένα εμπόδιο δυσκολεύει την επίτευξη ενός στόχου. Για παράδειγμα, μια βλάβη (το εμπόδιο) σε ένα εργοστάσιο παραγωγής εμποδίζει την εκπλήρωση των παραγγελιών (ο στόχος). Μόλις το εμπόδιο παρακαμφθεί, έχουμε φτάσει στη λύση του προβλήματος.

Με άλλα λόγια, **πρόβλημα** είναι μια κατάσταση που απαιτεί λύση αλλά η λύση της δεν είναι γνωστή ούτε προφανής.

2.2 Κατηγορίες προβλημάτων

Αν παρατηρήσουμε τα προβλήματα που αναφέρθηκαν παραπάνω, διαπιστώνουμε τη διαφορετική τους φύση. Η κατηγοριοποίησή τους μπορεί να γίνει με βάση διάφορα κριτήρια. Αν ως κριτήριο θέσουμε τη δυνατότητα επίλυσής τους, τα προβλήματα διακρίνονται σε:

- **Επιλύσιμα.** Χαρακτηρίζονται τα προβλήματα των οποίων η λύση έχει διατυπωθεί ή η συνάφειά τους με άλλα, ήδη λυμένα, μας επιτρέπει να θεωρούμε βέβαιη την δυνατότητα επίλυσής τους. Παραδείγματα τέτοιων προβλημάτων είναι η επίλυση δευτεροβάθμιων εξισώσεων, ο υπολογισμός του ρεύματος σε ένα κύκλωμα, η κατασκευή μιας γέφυρας, η εξοικονόμηση ενέργειας, η οργάνωση μιας βιβλιοθήκης.
- **Ανοικτά.** Χαρακτηρίζονται τα προβλήματα των οποίων η λύση δεν έχει ακόμα βρεθεί, ούτε έχει αποδειχθεί ότι δεν επιδέχονται λύση. Ως ανοικτά προβλήματα μπορούμε να αναφέρουμε την ανακάλυψη ζωής σε άλλους πλανήτες, τη θεραπεία του καρκίνου, την πρόβλεψη των σεισμών.
- **Μη επιλύσιμα.** Χαρακτηρίζονται τα προβλήματα για τα οποία έχουμε καταλήξει στην παραδοχή ότι δεν μπορούν να λυθούν. Τέτοια προβλήματα είναι ο τετραγωνισμός του κύκλου, το ταξίδι στο παρελθόν, η γήρανση του ανθρώπου.

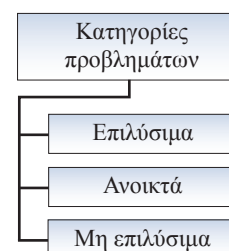
Τα προβλήματα που δίνονται προς επίλυση στον υπολογιστή έχουν να κάνουν με τη διενέργεια υπολογισμών οι οποίοι απαιτούν μια σειρά από λογικές και αριθμητικές πράξεις. Τα προβλήματα αυτά λέγονται υπολογιστικά και είναι αυτά με τα οποία θα ασχοληθούμε κυρίως σε αυτό το βιβλίο.

Στόχος είναι αυτό που έχουμε αποφασίσει ότι πρέπει να επιτύχουμε.

Εμπόδιο είναι αυτό που μας δυσκολεύει στην επίτευξη ενός στόχου.

Ο Jackson (1985) συνοψίζει:

Στόχος + Εμπόδιο = ΠΡΟΒΛΗΜΑ



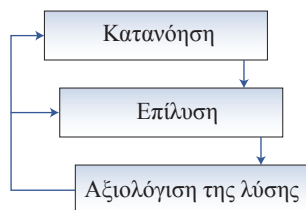
Εικόνα 2-1. Κατηγορίες προβλημάτων με κριτήριο την επιλυσιμότητα

Η εικασία του Goldbach ότι όλοι οι θετικοί άρτιοι αριθμοί μπορούν να γραφούν ως άθροισμα δύο πρώτων αριθμών, παραμένει ανοικτό πρόβλημα

Στα υπολογιστικά προβλήματα ζητούμε να βρούμε την απάντηση που ικανοποιεί τα δεδομένα του προβλήματος. Η πλειοψηφία των προβλημάτων απαιτούν τη διενέργεια πράξεων υπάρχουν όμως, και υπολογιστικά προβλήματα απόφασης ή βελτιστοποίησης. **Απόφασης** είναι ένα πρόβλημα στο οποίο η απάντηση είναι ένα ναι ή ένα όχι. Για παράδειγμα, δεδομένου ότι ο καιρός είναι άστατος, να πάω στο σχολείο με το ποδήλατο; **Βελτιστοποίησης** είναι ένα πρόβλημα στο οποίο αναζητούμε την απάντηση που ικανοποιεί κατά τον καλύτερο τρόπο τα δεδομένα του. Για παράδειγμα, δεδομένου ότι διαθέτω 100 ευρώ, ποιο είναι το καλύτερο κινητό που μπορώ να αγοράσω;

2.3 Διαδικασία επίλυσης προβλημάτων

Η αντιμετώπιση υπολογιστικών προβλημάτων γίνεται με συγκεκριμένη διαδικασία που περιέχει σαφή βήματα.



Εικόνα 2-2. Στάδια επίλυσης προβλήματος



1. Κατανόηση και Παρουσίαση προβλήματος

Σημαντικός παράγοντας για την **κατανόηση** του προβλήματος αλλά και τον καθορισμό της λύσης του είναι ο προσδιορισμός του χώρου του προβλήματος. Ο **χώρος του προβλήματος** καθορίζεται με τη βοήθεια ερωτήσεων («πώς», «πού», «τί», «γιατί», «πότε»). Απαντώντας στα ερωτήματα προσδιορίζουμε τα δεδομένα και τα ζητούμενα του προβλήματος. (Είσοδος - Έξοδος)

Δεδομένο προβλήματος ονομάζεται ένα γνωστό ή αποδεκτό στοιχείο το οποίο χρησιμοποιείται ως βάση ή προϋπόθεση για την επίλυση του προβλήματος.

Ζητούμενο προβλήματος είναι αυτό που ψάχνουμε για να βγούμε από τη δύσκολη κατάσταση στην οποία βρισκόμαστε.

Ανάλογα με τη φύση του προβλήματος τα δεδομένα και τα ζητούμενα μπορεί να είναι αριθμητικά, οικονομικά, λογικά κ.ά. Ανεξάρτητα από το είδος τους, τα δεδομένα και τα ζητούμενα πρέπει να έχουν τρεις (3) σημαντικές ιδιότητες: ορθότητα, πληρότητα και σαφήνεια.

Η **ορθότητα** είναι αναγκαίο να ελέγχεται κάθε φορά που επιδιώκεται η επίλυση ενός προβλήματος. Για παράδειγμα, έστω ότι σας ζητείται να ταξινομήσετε σε αλφαβητική σειρά τα επίθετα των συμμαθητών σας. Αν σας τα έχουν δώσει με ορθογραφικά λάθη, η ταξινόμηση που θα προκύψει θα είναι λανθασμένη. Πριν ξεκινήσουμε την προσπάθεια επίλυσης ενός προβλήματος πρέπει να ελέγξουμε την ορθότητα των δεδομένων.

Η **πληρότητα** πρέπει κι αυτή να ελέγχεται κάθε φορά που επιδιώκεται η επίλυση ενός προβλήματος. Έστω ότι στο παραπάνω παράδειγμα μας ζητάνε να ταξινομηθούν οι συμμαθητές μας με βάση την ημερομηνία γέννησης και μας δίνουν τα στοιχεία στην Εικόνα 2-3. Παρατηρούμε ότι τα δεδομένα είναι ελλιπή (εφόσον στην Παπαδάκη Μιχαέλα υπάρχει ημερομηνία γέννησης) και δεν μπορούμε να επιλύσουμε το πρόβλημα.

Η **σαφήνεια** είναι ένας σημαντικός παράγοντας για την ορθή επίλυση ενός προβλήματος. Τα δεδομένα αλλά και τα ζητούμενα, πρέπει να είναι **σαφή**, δηλαδή δεν υπάρχει ημερομηνία γέννησης) και δεν μπορούμε να επιλύσουμε το πρόβλημα.

Επίθετο	Όνομα	Ημερ. Γέννησης
Παπαδοπούλου	Κυριακή	3/22/98
Παπαδάκης	Κωνσταντίνος	2/25/98
Παπαδάκη	Μιχαέλα	...
Παπαδόπουλος	Μιχάλης	5/20/98

Η σαφήνεια είναι ένας σημαντικός παράγοντας για την ορθή επίλυση ενός προβλήματος. Τα δεδομένα αλλά και τα ζητούμενα, πρέπει να είναι σαφή, δηλαδή δεν απαιτούνται διευκρινιστικές ερωτήσεις από το πρόσωπο που καλείται να λύσει το πρόβλημα.

Για παράδειγμα, έστω ότι μας ζητάει η μητέρα μας να τη βοηθήσουμε στην παρασκευή ενός κέικ. Το επόμενο πρωί, έτοιμοι να ξεκινήσουμε τη διαδικασία παίρνουμε τη συνταγή και διαπιστώνουμε ότι δεν υπάρχουν τα υλικά που χρειάζονται. Το πρόβλημα που αρχικά είχαμε αντιληφθεί είναι διαφορετικό από αυτό που προέκυψε.

Έστω ότι μας ζητείται από μια φίλη μας να αγοράσουμε μια σοκολάτα. Η προθυμία να την εξυπηρετήσουμε μας οδηγεί στο Super Market. Εκεί έχουμε να επιλέξουμε ανάμεσα σε πάρα πολλές μάρκες, ποιότητες αλλά και μεγέθη. Στη συγκεκριμένη περίπτωση αυτό που κάνουμε είναι ή να επιλέξουμε κάποια στην τύχη, ή να ζητήσουμε νέες, πιο σαφείς, οδηγίες, τηλεφωνικά.

Ένας ακόμα σημαντικός παράγοντας για την ορθή επίλυση ενός προβλήματος είναι ο τρόπος παρουσίασής του. Η παρουσίαση του προβλήματος μπορεί να γίνει φραστικά (με λέξεις) ή αλγεβρικά (με μαθηματικά σύμβολα).

Παράδειγμα 2-1. Βρείτε δύο αριθμούς που το άθροισμα τους είναι 78 και το γινόμενο τους 1296.

2. Επίλυση του προβλήματος

Στην προηγούμενη παράγραφο περιγράψαμε τη διαδικασία εύρεσης των δεδομένων και των ζητούμενων μέσα από την παρουσίαση του προβλήματος.

Φυσική Γλώσσα	Γλώσσα της Άλγεβρας
Βρείτε δύο αριθμούς	x, y
που το άθροισμα τους είναι 78	$x+y=78$
και το γινόμενο τους 1296	$x \cdot y=1296$

Προχωρώντας προς το στάδιο της επίλυσης (problem solving) καταλαβαίνουμε ότι σε πολλά προβλήματα η λύση τους δεν είναι άμεσα γνωστή. Η επίλυση περιλαμβάνει την ανάλυση, μοντελοποίηση, σχεδίαση, και υλοποίηση μίας κατάλληλης λύσης του προβλήματος με τον υπολογιστή.

Εικόνα 2-3. Δεδομένα χωρίς πληρότητα

Σαφήνεια δεδομένων

Σαφήνεια ζητούμενων

Η παρουσίαση του προβλήματος καθορίζει σε γενικές γραμμές την επιθυμητή σχέση εισόδου-εξόδου.

Εικόνα 2-4. Παρουσίαση του προβλήματος σε φυσική και αλγεβρική γλώσσα

Βασικές έννοιες στην επίλυση προβλημάτων είναι η δομή, η ανάλυση και η σύνθεση. Με τον όρο **δομή**, εννοούμε τον τρόπο με τον οποίο επιμέρους στοιχεία σχετίζονται και συνδέονται μεταξύ τους ώστε να σχηματίζουν ενιαίο σύνολο. **Ανάλυση** είναι ο διαχωρισμός ενός συνόλου στα συστατικά του στοιχεία. Σύνθεση είναι η τοποθέτηση στοιχείων σε συσχετισμό μεταξύ τους έτσι ώστε να δημιουργείται ένα σύνολο.

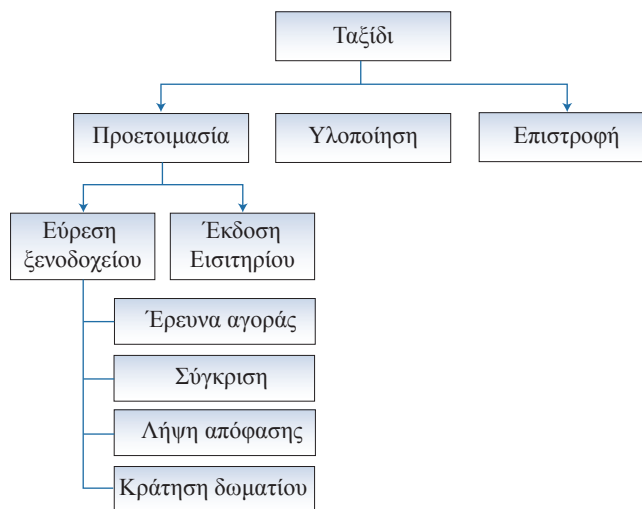
Όταν το ενιαίο σύνολο είναι το πρόβλημα, τα επιμέρους στοιχεία είναι τα μικρότερα προβλήματα στα οποία χωρίζεται, τα υποπροβλήματα. Εφαρμόζοντας την ανάλυση και τη σύνθεση στην επίλυση προβλημάτων δημιουργούνται τρεις βασικές μεθοδολογίες η Αναλυτική (Top Down), η Συνθετική (Bottom Up), και η Μικτή (Mixed).

Αναλυτική (Top Down problem solving) είναι η μεθοδολογία επίλυσης προβλημάτων που βασίζεται στη σχεδίαση από το γενικό στο ειδικό. Η γενική αρχή της είναι ότι για να λύσουμε κάποιο σύνθετο πρόβλημα πρέπει:

1. Να ορίσουμε τα υποπροβλήματα (sub-problems).
2. Να επαναλάβουμε την διαδικασία αυτή για κάθε ένα από τα υποπροβλήματα, όσο αυτό είναι αναγκαίο.
3. Όταν φτάσουμε σε υποπροβλήματα που δεν απαιτούν επιπλέον διάσπαση, προχωράμε στην άμεση επίλυσή τους, τότε έχει επιλυθεί και το αρχικό πρόβλημα.

Παράδειγμα 2-2. Ας δούμε ένα παράδειγμα εφαρμογής της αναλυτικής μεθοδολογίας: Θέλουμε να υλοποιήσουμε ένα ταξίδι στο εξωτερικό.

Εικόνα 2-5. Προετοιμασία ταξιδιού



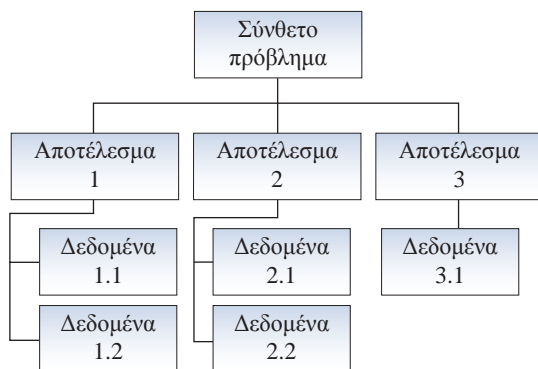
Υπάρχουν πολλοί τρόποι με τους οποίους μπορούμε να αναλύσουμε το πρόβλημα σε υποπροβλήματα. Ένας από όλους είναι να το αναλύσουμε αρχικά σε τρία υποπροβλήματα: προετοιμασία ταξιδιού, υλοποίηση και επιστροφή. Η προετοιμασία του ταξιδιού μπορεί να περιλαμβάνει έκδοση εισιτηρίων και εύρεση ξενοδοχείου. Το πρόβλημα της εύρεσης ξενοδοχείου με τη σειρά του μπορεί να αναλυθεί στα υποπροβλήματα της έρευνας αγοράς, σύγκρισης, λήψης απόφασης και κράτησης δωματίου.

Με τον ίδιο τρόπο θα μπορούσαμε να αναλύσουμε και τα υπόλοιπα υπο-προβλήματα προχωρώντας την ανάλυση σε ακόμα χαμηλότερα επίπεδα. Η διαγραμματική αναπαράσταση της ανάλυσης του προβλήματος δίνεται στην Εικόνα 2.5.

Συνθετική (bottom up problem solving) είναι η μεθοδολογία επίλυσης προβλημάτων που βασίζεται στη σχεδίαση από το ειδικό στο γενικό με τη σύνδεση των δεδομένων. Για το σκοπό αυτό:

1. Αρχίζουμε από δεδομένα που είναι ορθά.
2. Στη συνέχεια, με μια λογικά ορθή διαδικασία, παράγουμε νέα δεδομένα και αποτελέσματα
3. Η διαδικασία τερματίζει, όταν παραχθεί το ζητούμενο του προβλήματος.

Μεικτή (mixed), είναι η μεθοδολογία επίλυσης προβλημάτων που συνδυάζει την αναλυτική και τη συνθετική μέθοδο. Σύμφωνα με τη μεικτή μεθοδολογία, κάποια από τα υποπροβλήματα επιλύονται με την αναλυτική και κάποια με τη συνθετική μέθοδο.



Εικόνα 2-6. Συνθετική μεθοδολογία

Η ανάλυση του προβλήματος μπορεί να αναπαρασταθεί είτε φραστικά είτε διαγραμματικά. Στη διαγραμματική αναπαράσταση, η περιγραφή έχει τη μορφή γενεολογικού δέντρου, στο οποίο κάθε πρόβλημα έχει «παιδιά» τα υποπροβλήματα στα οποία αναλύεται.

Μπορείτε να σκεφτείτε ένα πρόβλημα που να επιλύεται με τη συνθετική ή τη μεικτή μεθοδολογία;

3. Αξιολόγηση της λύσης

Το στάδιο αξιολόγησης της λύσης είναι πολύ σημαντικό. Για να ελέγξουμε εάν έχουμε επιλύσει σωστά ένα πρόβλημα, αρχικά καταγράφουμε υποθετικά δεδομένα σύμφωνα με τις απαιτήσεις του προβλήματος. Στη συνέχεια, εφαρμόζουμε τα βήματα επίλυσης για να εξάγουμε τα αποτελέσματα. Τέλος, συγκρίνουμε τα αποτελέσματα που πήραμε με τα ζητούμενα του προβλήματος. Εάν διαπιστώσουμε λάθος, εντοπίζουμε το τμήμα της λύσης που εκτελεί τη λανθασμένη λειτουργία, το διορθώνουμε και επαναλαμβάνουμε τη διαδικασία ελέγχου, έως ότου τα αποτελέσματα να μη διαφέρουν από τα ζητούμενα. Πολλές φορές το λάθος μπορεί να οφείλεται και στην κατανόηση του προβλήματος. Στην περίπτωση αυτή η διαδικασία επίλυσης επαναλαμβάνεται από την αρχή.



Ερωτήσεις - Θέματα για συζήτηση

A. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Προβλήματα είναι καταστάσεις που πρέπει να αντιμετωπιστούν αλλά δε γνωρίζουμε το πώς.
2. Επιλύσιμα χαρακτηρίζονται τα προβλήματα των οποίων η συνάφειά της λύσης τους με άλλα, ήδη λυμένα, μας επιτρέπει να θεωρούμε βέβαιη την δυνατότητα επίλυσής τους.
3. Ανοικτά χαρακτηρίζονται τα προβλήματα για τα οποία έχουμε καταλήξει στην παραδοχή ότι δεν μπορούν να λυθούν.
4. Όταν τα δεδομένα ενός προβλήματος είναι ελλιπή ή ασαφή, τότε το πρόβλημα χαρακτηρίζεται ως μη επιλύσιμο.
5. Ο έλεγχος των δεδομένων μπορεί να οδηγήσει και πάλι στην είσοδο.
6. Δομή ενός προβλήματος είναι η εύρεση του συνόλου των μερών που το απαρτίζουν.
7. Η ανάλυση των προβλημάτων σε υποπροβλήματα πρέπει να αποφεύγεται γιατί με αυτόν τον τρόπο αντί να λύσουμε ένα πρόβλημα πρέπει να λύσουμε πολλά προβλήματα.
8. Η λύση της εξίσωσης $3x+7=16$ είναι πρόβλημα υπολογιστικό.

B. Σημειώστε την κατηγορία στην οποία ανήκει το κάθε πρόβλημα.

	Επιλύσιμο	Ανοικτό	Μη επιλύσιμο
1. Ο υπολογισμός της υποτείνουσας ενός ορθογώνιου τριγώνου όταν γνωρίζω τις 2 πλευρές του.			
2. Ακριβής πρόγνωση σεισμικών δονήσεων.			
3. Θεραπεία του aids.			
4. Η οργάνωση μιας βιβλιοθήκης			
5. Κατασκευή αυτοκινήτου με αυτόματο «πιλότο».			
6. Η γήρανση του ανθρώπου			
7. Παρασκευή εμβολίου για τον καρκίνο.			
8. Η εικασία του Γκόλντμπαχ (Goldbach, κάθε άρτιος μπορεί να γραφτεί ως άθροισμα δύο πρώτων αριθμών)			
9. Το πρόβλημα του τετραγωνισμού του κύκλου με κανόνα και διαβήτη			
10. Η επίλυση της δευτεροβάθμιας εξίσωσης			

Γ. Συμπληρώστε τις προτάσεις με την κατάλληλη λέξη:

1. Δεδομένο ενός προβλήματος ονομάζεται ένα _____ ή αποδεκτό στοιχείο το οποίο χρησιμοποιείται ως _____ ή προϋπόθεση στην επίλυση προβλημάτων.
2. Πριν την επίλυση ενός προβλήματος, ελέγχεται η _____ των δεδομένων.
3. Οι τρεις βασικές μεθοδολογίες επίλυσης προβλημάτων που χρησιμοποιούνται είναι η _____, η _____, και η _____.

Ασκήσεις - Προβλήματα

1. Να διατυπώσετε με αλγεβρικό τρόπο τα παρακάτω υπολογιστικά :
 - α. Να υπολογιστεί το εμβαδό ενός τριγώνου, αν σας δίνεται ότι το εμβαδό του είναι το μισό του γινομένου της βάσης του επί το ύψος του. β. Η Κασσιανή είναι μεγαλύτερη της Άννα-Μαρίας κατά 10 χρόνια. Η Άννα-Μαρία είναι μικρότερη του Γιώργου κατά 3 χρόνια. Ο Κωνσταντίνος είναι 20 ετών και κατά 1 χρόνο μεγαλύτερος του Γιώργου. Πόσο χρονών είναι η Κασσιανή;
2. «Η Μάρθα θέλει να αγοράσει έναν ηλεκτρονικό υπολογιστή». Ποια είναι τα πιθανά ερωτήματα που θα πρέπει να απαντηθούν προκειμένου να προσδιοριστεί καλύτερα το ζητούμενο στο πρόβλημα της Μάρθας;»
3. Να λυθεί η εξίσωση: $ax^2+bx+c=0$. Αναλύστε σε υποπροβλήματα και παρουσιάστε σχηματικά το πρόβλημα.
4. «Ο Δημήτρης βρίσκεται στο λιμάνι του Πειραιά και θέλει να πάει στη Θεσσαλονίκη». Αναλύστε σε υποπροβλήματα και παρουσιάστε σχηματικά το πρόβλημα. Ποια μέθοδο χρησιμοποιήσατε και γιατί; Με βάση το είδος επίλυσης σε ποια ή ποιες κατηγορίες προβλημάτων θα το εντάσσατε;



Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να περιγράφετε την έννοια του προβλήματος.
- να αναγνωρίζετε και να απαριθμείτε τις κατηγορίες προβλημάτων.
- να διακρίνετε την ύπαρξη υπολογιστικών προβλημάτων και να αναφέρετε τις φάσεις επίλυσής τους.
- να προσδιορίζετε τα δεδομένα και τα ζητούμενα ενός προβλήματος
- να ελέγχετε την πληρότητα, την ορθότητα και την σαφήνεια των δεδομένων και των ζητούμενων.
- να αναλύετε ένα πρόβλημα σε απλούστερα και να διατυπώνετε τα αποτελέσματα της ανάλυσης με τρόπο σαφή.



κεφάλαιο

3

**Θέματα Θεωρητικής
Επιστήμης
των Υπολογιστών**

Αλγόριθμοι



Στόχοι

Στόχοι του κεφαλαίου είναι οι μαθητές:

- να αναγνωρίζουν τη χρησιμότητα των αλγορίθμων στην επίλυση προβλημάτων.
- να μπορούν να περιγράφουν τα βασικά χαρακτηριστικά κάθε αλγόριθμου.
- να διακρίνουν τους διαφορετικούς τύπους αλγορίθμων.
- να συγκρίνουν τις διαφορετικές αναπαραστάσεις αλγορίθμων και να ξεχωρίζουν τα πλεονεκτήματά τους.



Λέξεις κλειδιά

Αλγόριθμος

Χαρακτηριστικά αλγορίθμου (Καθοριστικότητα, Περαιτότητα, Αποτελεσματικότητα, Είσοδος, Έξοδος), Ανάλυση Αλγορίθμων, Θεωρία Υπολογισμού, Υπολογισσιμότητα, Πολυπλοκότητα αλγορίθμων, Σειριακή επεξεργασία, Παράλληλη επεξεργασία, Επαναληπτικοί και Αναδρομικοί αλγόριθμοι, Φυσική Γλώσσα, Διάγραμμα Ροής, Γλώσσες Περιγραφής Αλγορίθμων, Ψευδογλώσσα, Δομές δεδομένων, Εντολές αλγορίθμου, Δομές αλγορίθμων, Εκσφαλμάτωση, Λογικά λάθη, Συντακτικά λάθη, Τεκμηρίωση



Εισαγωγικές ερωτήσεις

1. Ποιος είναι ο αλγόριθμος που χρησιμοποιείτε καθημερινά στην προετοιμασία σας για το σχολείο;
2. Έχετε αναρωτηθεί από τι εξαρτάται η ταχύτητα εντοπισμού των στοιχείων ενός μαθητή σε ένα πρόγραμμα υπολογιστή;
3. Πώς περιγράφετε σε έναν φίλο σας τη λύση που δώσατε σε ένα πρόβλημα της καθημερινότητάς σας;

3. Αλγόριθμοι

3.1 Η έννοια του αλγορίθμου

Στο προηγούμενο κεφάλαιο είδαμε ότι για την επίλυση ενός προβλήματος μπορούν να ακολουθηθούν διάφορες τεχνικές οι οποίες έχουν κατηγοριοποιηθεί και συστηματοποιηθεί. Παρόλα αυτά, η διεργασία αντιμετώπισης προβλημάτων παραμένει μια υψηλά διαισθητική λειτουργία του ανθρώπινου εγκεφάλου. Αναρωτηθήκατε το πώς εκτελούνται αριθμητικές πράξεις στο μυαλό μας με τόση ευκολία; Το πώς είμαστε σίγουροι ότι το αποτέλεσμα τους είναι σωστό; Στο μυαλό μας, με θαυμαστό τρόπο, μέσα από τη γνώση, δημιουργούνται διαδικασίες επίλυσης προβλημάτων που καταγράφονται σε αυτό ως σωστές. Με την εμφάνιση των ηλεκτρονικών υπολογιστών, το στοίχημα του ανθρώπου ήταν να μεταδώσει αυτή τη θαυμαστή ιδιότητα του εγκεφάλου του στον υπολογιστή. Η διαδικασία επίλυσης ενός προβλήματος οδηγεί στη δημιουργία ενός αλγορίθμου.

Οι αλγόριθμοι στην καθημερινή ζωή της Μαρίας

- Βάλε ελαφριά ρούχα. Πήγαινε στο μπαλκόνι που έχει ησυχία. Κατέβασε την τέντα. Πάρε ένα χυμό να δροσιστείς και φυσικά το βιβλίο σου, είπε η μητέρα μου.

Όχι πάλι, σκέφτηκα.... Εντολές, εντολές, εντολές.... Δε μου φτάνει το πρόβλημα μου, έχω και τους αλγόριθμους της μητέρας μου! Ξεκίνησα το διάβασμα. Ορισμός: \sqrt{x} είναι ο αριθμός y για τον οποίο ισχύει $y^2 = x$. Ωραία σκέφτηκα. Η ρίζα του 9 είναι το 3. Η ρίζα του 49 είναι το 7. Ναι, αλλά πώς θα υπολογίσω τη ρίζα του 2 ή τη ρίζα του 2371; Για τη λύση θα πρέπει να σκέφτομαι και να συνδυάζω ιδιότητες. Θα έχω όμως και τότε αποτέλεσμα; Αποτελεσματικότερο το κομπιουτεράκι της μητέρας μου. Εύκολο και σίγουρο. Πληκτρολόγησα το 2, πάτησα το σύμβολο \sqrt{x} και να το αποτέλεσμα 1.4142135623730950488016887242097 !!!!

Πώς το βρήκε τόσο γρήγορα; Πώς το σκέφτηκε; Γιατί εγώ πρέπει να διαβάζω, να συμπληρώνω φυλλάδια, να κάνω τόσο πολύπλοκους συνδυασμούς στο μυαλό μου και να μην μπορώ να υπολογίσω ΟΛΕΣ τις ρίζες, όπως το κομπιουτεράκι; Θα πρέπει, σκέφτηκα, να υπάρχει μια συγκεκριμένη γενική διαδικασία που αν την ακολουθήσω, θα μπορώ να υπολογίσω οποιαδήποτε ρίζα. Πρέπει να υπάρχει ένας αλγόριθμος!

Για επιβεβαίωση των σκέψεών μου έκανα αναζήτηση στον παγκόσμιο ιστό, στη μηχανή που λειτουργεί με τον εξυπνότερο και γρηγορότερο αλγόριθμο αναζήτησης. Φράση αναζήτησης: «Αλγόριθμος για τον υπολογισμό της τετραγωνικής ρίζας ενός αριθμού» και πρώτο αποτέλεσμα «Η προσεγγιστική μέθοδος του Newton για τον υπολογισμό της τετραγωνικής ρίζας αριθμού».

Αν y μια τιμή κατά προσέγγιση της \sqrt{x} , τότε το αποτέλεσμα της πράξης $\frac{y + \frac{x}{y}}{2}$ δίνει μια καλύτερη προσέγγιση, πιο κοντά στην πραγματική τιμή της ρίζας.

Αρχισα να το εφαρμόζω, ελπίζοντας να βρω την τιμή που μου έβγαλε το κομπιουτεράκι.

Πρώτη φορά $\frac{2 + \frac{2}{2}}{2} = 1.5$, δεύτερη φορά $\frac{1.5 + \frac{2}{1.5}}{2} = 1.4166$, τρίτη φορά... $\frac{1.4166 + \frac{2}{1.4166}}{2} = 1.41421568$

Το εφάρμοσα μόλις τρεις φορές και παρατήρησα ότι το αποτέλεσμα σύγκλιε με την τιμή που είχε βγάλει το κομπιουτεράκι. Είχα ανακαλύψει μέσα από την παρατήρηση και την έρευνα την έννοια του αλγορίθμου, μια μηχανικά εκτελέσιμη υπολογιστική διαδικασία, μέσω της οποίας, αν ξέρεις πράξεις και την ακολουθήσεις πιστά, υπολογίζεις μια ποσότητα. Με αυτό τον τρόπο θα μπορούσα να υπολογίσω οποιαδήποτε τετραγωνική ρίζα.

Η έννοια του αλγορίθμου είναι θεμελιώδης για την επιστήμη της Πληροφορικής. Στα Μαθηματικά δίνουμε ορισμούς που περιγράφουν το «τι είναι». Διατυπώνουμε γενικές ιδιότητες και θεωρήματα. Συνδυάζοντάς τα, λύνουμε ειδικά προβλήματα (π.χ. τον υπολογισμό της ρίζας του 2). Στην Πληροφορική κατασκευάζουμε αλγορίθμους που περιγράφουν το «πώς υπολογίζεται», δηλαδή δημιουργούμε συγκεκριμένες υπολογιστικές διαδικασίες οι οποίες εκτελούνται μηχανικά. Στη συνέχεια, τις εφαρμόζουμε για να λύσουμε γενικά προβλήματα (π.χ. υπολογισμός της ρίζας οποιουδήποτε αριθμού).



Αλγόριθμος είναι η ακριβής περιγραφή μιας σειράς βημάτων που απαιτούνται για την επίλυση ενός προβλήματος.

Ο αλγόριθμος αποτελεί τη βάση για τη δημιουργία ενός προγράμματος που θα εκτελεστεί από έναν ηλεκτρονικό υπολογιστή. Η έννοιά του είναι γενικότερη από εκείνη του προγράμματος και δε συνδέεται αποκλειστικά με την Πληροφορική. Μπορεί να εκφραστεί ακόμα και σε φυσική γλώσσα, Ελληνικά ή Αγγλικά.

Μια συνταγή μαγειρικής είναι ένας αλγόριθμος. Οι οδηγίες του καθηγητή για την επίλυση της δευτεροβάθμιας εξίσωσης είναι, επίσης, αλγόριθμος. Στην επίλυση προβλημάτων μέσω αλγορίθμων, υπάρχουν τρεις (3) διαφορετικοί διακριτοί ρόλοι:

- Ο λύτης, αυτός που καλείται να αντιμετωπίσει το πρόβλημα σχεδιάζοντας τον αλγόριθμο.
- Ο εκτελεστής, αυτός που εφαρμόζει πιστά τις εντολές του αλγορίθμου που έφτιαξε ο λύτης.
- Ο χρήστης, αυτός που ενεργοποιεί τον αλγόριθμο, καλώντας τον εκτελεστή να λύσει, όποτε θέλει, το πρόβλημα.

Επίλυση προβλήματος με ΗΥ

Λύτης = Προγραμματιστής

Εκτελεστής = Υπολογιστής

Χρήστης = Χρήστης Η/Υ



Ερωτήσεις - Θέματα για συζήτηση

A. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Η έννοια του αλγορίθμου είναι θεμελιώδης για την επιστήμη της Πληροφορικής.
2. Η έννοια του αλγορίθμου είναι γενικότερη από εκείνη του προγράμματος.
3. Η έννοια του αλγορίθμου συνδέεται αποκλειστικά με την Πληροφορική.
4. Μια συνταγή μαγειρικής είναι ένας αλγόριθμος
5. Στην επίλυση προβλημάτων μέσω αλγορίθμων, υπάρχουν δύο διαφορετικοί διακριτοί ρόλοι

B. Συμπληρώστε τις προτάσεις με την κατάλληλη λέξη:

1. Ο αλγόριθμος αποτελεί τη βάση για τη δημιουργία ενός

2. Οι διακριτοί ρόλοι στην επίλυση προβλημάτων μέσω αλγορίθμων είναι ο _____, ο _____ και ο _____ .

Γ. Αντιστοιχίστε κάθε στοιχείο της στήλης Α με την κατάλληλη έννοια της στήλης Β

1.

Στήλη Α	Στήλη Β
α. Μαθηματικά	1. ορισμοί
	2. ιδιότητες
β. Πληροφορική	3. θεωρήματα
	4. υπολογιστικές διαδικασίες
	5. αλγόριθμοι

2. Επίλυση προβλήματος με υπολογιστή

Στήλη Α	Στήλη Β
α. Λύτης	1. Χρήστης Η/Υ
β. Εκτελεστής	2. Προγραμματιστής
γ. Χρήστης	3. Υπολογιστής

Ασκήσεις - Προβλήματα

Αναζητήστε τον αλγόριθμο του Ευκλείδη για τον υπολογισμό του Μέγιστου Κοινού Διαιρέτη. Μελετήστε τη λειτουργία του και εφαρμόστε τον για τον υπολογισμό του ΜΚΔ των αριθμών 32 και 40.



Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να ορίζετε και να περιγράφετε την έννοια του αλγορίθμου
- να προσδιορίζετε τη θέση του αλγορίθμου στην καθημερινή ζωή
- να αναγνωρίζετε τη σημασία της αλγοριθμικής σκέψης στην επίλυση προβλημάτων και τη λήψη αποφάσεων
- να δίνετε παραδείγματα αλγορίθμων είτε για απλά καθημερινά προβλήματα είτε για απλά υπολογιστικά προβλήματα.
- να αναγνωρίζετε τους ρόλους του λύτη, του εκτελεστή και του χρήστη στη δημιουργία ενός αλγορίθμου.



3.2 Χαρακτηριστικά αλγορίθμου

Ο αλγόριθμος αποτελείται από μια σειρά εντολών που, αν εφαρμοσθούν πιστά, οδηγούν στην επίλυση ενός προβλήματος. Είναι, όμως, οποιαδήποτε σειρά από εντολές ένας αλγόριθμος; Η απάντηση είναι όχι. Οι εντολές ενός αλγορίθμου, αλλά και ο αλγόριθμος ως ολότητα, θα πρέπει να ικανοποιούν κάποια κριτήρια:

- **Είσοδος:** είναι τα στοιχεία που χρειάζεται ο αλγόριθμος για να εκτελεσθεί.
- **Έξοδος:** είναι τα στοιχεία που παράγει ο αλγόριθμος, τα αποτελέσματά του.
- **Καθοριστικότητα:** κάθε εντολή θα πρέπει να είναι μονοσήμαντη, δηλαδή να καθορίζει με απόλυτη σαφήνεια και ακρίβεια τον τρόπο εκτέλεσής της σε κάθε δυνατή περίπτωση.
- **Περατότητα:** εκτελώντας τα βήματα του αλγορίθμου, θα πρέπει να φθάνουμε σε πέρας (τέλος) σε κάθε δυνατή περίπτωση.
- **Αποτελεσματικότητα:** κάθε εντολή θα πρέπει να είναι διατυπωμένη με απλό τρόπο, ώστε να μπορεί να εκτελεσθεί.

Ας εξετάσουμε τα αλγοριθμικά κριτήρια μέσα από ένα πρόβλημα.

Πρόβλημα: πώς φτιάχνουμε ένα κέικ σοκολάτας;

Αλγόριθμος: Αρκεί να ανοίξουμε ένα βιβλίο ζαχαροπλαστικής και να ακολουθήσουμε πιστά οποιαδήποτε συνταγή.

Είσοδος: τα υλικά **Έξοδος:** το ζεστό ζουμερό κέικ

Καθοριστικότητα: η εντολή «Ψήσε σε φούρνο ήδη προθερμασμένο σε θερμοκρασία 175 βαθμούς Κελσίου για ακριβώς 45 λεπτά» είναι απόλυτα σαφής, ενώ η εντολή «Ψήσε σε φούρνο μέχρι να φουσκώσει» σίγουρα δεν είναι!

Περατότητα: Μια εντολή επανάληψης «Ψήσε στο φούρνο μέχρι να σταματήσει η γη να κινείται» θα ήταν καταστροφική για τον αλγόριθμο και το κέικ. Ο αλγόριθμος δε θα τελειώσει ποτέ και το κέικ θα καιγόταν.

Αποτελεσματικότητα: Η εντολή «Ανοίξε το φούρνο και πιάσε το καυτό κέικ με τα δύο σου γυμνά χέρια», αν και απόλυτα σαφής, δε θα ήταν καταστροφική για το κέικ αλλά θα ήταν για τα χέρια μας και κατά συνέπεια για τον αλγόριθμο!

Είναι φανερό από το παραπάνω παράδειγμα ότι τα κριτήρια της καθοριστικότητας και αποτελεσματικότητας είναι σχετικά με αυτόν που θα κληθεί να εφαρμόσει τον αλγόριθμο. Κάθε εκτελεστής έχει διαφορετικές δυνατότητες, διαφορετικά δυνατά και αδύνατα σημεία. Ο άνθρωπος, για παράδειγμα, μπορεί να ακούσει, να διαβάσει, μπορεί να μιλήσει αλλά ακόμα και να εφαρμόσει τύπους, να σκεφθεί, να αυτοσχεδιάσει. Ο υπολογιστής, από την άλλη, μπορεί να εισάγει από το πληκτρολόγιο, να εμφανίσει, να κάνει αριθμητικές και λογικές πράξεις. Κάθε εκτελεστής έχει διαφορετικό σύνολο κατανοητών και εφικτών εντολών. Ο άνθρωπος έχει ένα ευρύ σύνολο εκτελέσιμων εντολών. Ο υπολογιστής μπορεί να εκτελεί μόνο βασικές πράξεις. Σε κάθε περίπτωση, μια σειρά εντολών θα πρέπει να λαμβάνει υπόψη το μελλοντικό εκτελεστή της, προκειμένου να ικανοποιεί τα αλγοριθμικά κριτήρια.

Ερωτήσεις - Θέματα για συζήτηση



A. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Μια σειρά από εντολές είναι ένας αλγόριθμος.
2. Οι εντολές ενός αλγορίθμου πρέπει να ικανοποιούν κάποια κριτήρια.
3. Ένας αλγόριθμος είναι μια σειρά εντολών.
4. Ο υπολογιστής μπορεί να εκτελεί μόνο βασικές πράξεις.
5. Μια σειρά εντολών πρέπει να λαμβάνει υπόψη το μελλοντικό εκτελεστή της, προκειμένου να ικανοποιεί τα αλγοριθμικά κριτήρια.
6. Ο άνθρωπος έχει ευρύτερο σύνολο εκτελέσιμων εντολών από τον υπολογιστή.

B. Αντιστοιχίστε κάθε στοιχείο της στήλης A με την κατάλληλη έννοια της στήλης B:

Στήλη A	Στήλη B
Είσοδος	κάθε εντολή πρέπει να είναι απλά διατυπωμένη και εκτελέσιμη
Έξοδος	δεδομένα
Καθοριστικότητα	ο αλγόριθμος φθάνει πάντα σε τέλος
Περατότητα	κάθε εντολή πρέπει να είναι μονοσήμαντη
Αποτελεσματικότητα	αποτελέσματα

Γ. Συμπληρώστε τις προτάσεις με την κατάλληλη λέξη:

1. Ο αλγόριθμος αποτελείται από μια σειρά _____ που, αν εφαρμοστούν πιστά, οδηγούν στην _____ ενός προβλήματος.
2. Τα κριτήρια της _____ και _____ είναι σχετικά με αυτόν που θα κληθεί να εφαρμόσει τον αλγόριθμο.

Δ. Ερωτήσεις

1. Δίνεται η εντολή «Πολλαπλασίασε την ακτίνα επί 2 και επί 3.14» σε φυσική γλώσσα. Έχει καθοριστικότητα; Έχει αποτελεσματικότητα;
2. Ποιος έχει ευρύτερο σύνολο εκτελέσιμων εντολών, ο άνθρωπος ή ο υπολογιστής;
3. Συζητήσετε στην τάξη το πώς επηρεάζει ο εκτελεστής (άνθρωπος ή υπολογιστής) την αποτελεσματικότητα ενός αλγορίθμου.



Ασκήσεις - Προβλήματα

1. Περιγράψτε με φυσική γλώσσα αλγόριθμο υπολογισμού της περιφέρειας και του εμβαδού ενός κύκλου. Δώστε τον σε συμμαθητή σας να τον εκτελέσει και συζητήστε στην τάξη αν ικανοποιεί τα αλγοριθμικά κριτήρια.



Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να αναγνωρίζετε πότε μια σειρά από εντολές αποτελεί αλγόριθμο
- να αναγνωρίζετε τη σημασία του να τερματίζει και να δίνει αποτελέσματα ένας αλγόριθμος.
- να αναγνωρίζετε το ρόλο του λύτη και του εκτελεστή στην αποτελεσματικότητα ενός αλγορίθμου.

3.3 Ανάλυση αλγορίθμων

Οι αλγόριθμοι προκύπτουν από την ανάγκη εύρεσης λύσεων σε προβλήματα που μας απασχολούν και φυσικά μπορούμε να έχουμε πολλές λύσεις για το ίδιο πρόβλημα άρα και πολλούς αλγόριθμους. Πώς όμως μπορούμε να αξιολογήσουμε τους διαφορετικούς αλγόριθμους και να εντοπίσουμε ποιος είναι καλύτερος έναντι κάποιου άλλου; Υπάρχουν στοιχεία σε έναν αλγόριθμο που να μπορούν να μετρηθούν και να υπολογιστεί πόσο αποδοτικός είναι; Στα ερωτήματα αυτά προσπαθεί να απαντήσει ο τομέας της επιστήμης των υπολογιστών που ασχολείται με την **ανάλυση αλγορίθμων**.

Τα δύο στοιχεία ενός αλγορίθμου που μπορούν να μετρηθούν και να μας δώσουν μια εκτίμηση της αποδοτικότητάς του είναι ο χρόνος που χρειάζεται για να εκτελεστεί και ο χώρος, δηλαδή η μνήμη, που απαιτείται για να λειτουργήσει σωστά. Μεγαλύτερη έμφαση δίνουμε στη σωστή χρήση του χρόνου και όχι του χώρου, καθώς το κόστος των κυκλωμάτων μνήμης των υπολογιστών μειώνεται συνεχώς με την εξέλιξη της τεχνολογίας.

Η εκτίμηση του χρόνου εκτέλεσης ενός αλγορίθμου μπορεί να γίνει με δύο τρόπους: α) πειραματικά, χρονομετρώντας το διάστημα από τη στιγμή που ξεκινά να εκτελείται ένας αλγόριθμος μέχρι τη στιγμή που θα τερματίσει, β) θεωρητικά, προσπαθώντας να βρούμε μια μαθηματική σχέση που να συνδέει το χρόνο εκτέλεσης με το πλήθος των δεδομένων εισόδου. Επειδή οι αλγόριθμοι μπορούν να εφαρμόζονται σε διαφορετικό πλήθος δεδομένων κάθε φορά και φυσικά σε μηχανήματα διαφορετικών δυνατοτήτων, η πειραματική προσέγγιση δε μας δίνει πολλά στοιχεία για την αποδοτικότητα του αλγορίθμου. Από την άλλη πλευρά, η θεωρητική προσέγγιση της εύρεσης μιας μαθηματικής σχέσης μας καλύπτει καλύτερα διότι έτσι μπορούμε να εκτιμήσουμε την αποδοτικότητα ενός αλγορίθμου ανεξάρτητα από το πλήθος των δεδομένων εισόδου ή της υπολογιστικής μηχανής που χρησιμοποιούμε.

Ας δούμε όμως καλύτερα ένα παράδειγμα. Έστω ότι έχουμε έναν τηλεφωνικό κατάλογο που περιλαμβάνει 128000 ονόματα και τηλέφωνα. Θέλουμε να φτιάξουμε έναν αλγόριθμο που, εάν του δίνουμε ένα όνομα ως είσοδο, αυτός να μας επιστρέφει το τηλέφωνο που αντιστοιχεί σε αυτό. Μια πρώτη λύση στο πρόβλημά μας είναι να αρχίσουμε να διαβάζουμε από την αρχή όλα τα ονόματα του τηλεφωνικού καταλόγου και μόλις βρούμε το όνομα που μας ενδιαφέρει να επιστραφεί το τηλέφωνό του. Η λύση αυτή είναι ιδιαίτερα χρονοβόρα, χρησιμοποιείται όμως στους υπολογιστές και ονομάζεται **σειριακή αναζήτηση**.

Προσπαθώντας να αναλύσουμε τον αλγόριθμό μας παρατηρούμε ότι εάν το όνομα που δίνουμε προς αναζήτηση βρίσκεται στις πρώτες σελίδες του τηλεφωνικού καταλόγου, ο αλγόριθμός μας βρίσκει το αντίστοιχο τηλέφωνο γρήγορα, ενώ, εάν το όνομα που δίνουμε βρίσκεται στις τελευταίες σελίδες καθυστερεί σημαντικά. Εάν συμφωνήσουμε ότι ο χρόνος ανάγνωσης και σύγκρισης ενός ονόματος με αυτό που ψάχνουμε είναι ίσος με ένα δευτερόλεπτο, τότε ο χρόνος εκτέλεσης του αλγορίθμου θα ισούται με τη θέση του ονόματος στον τηλεφωνικό κατάλογο. Έτσι, εάν αναζητούμε το 10ο στη σειρά

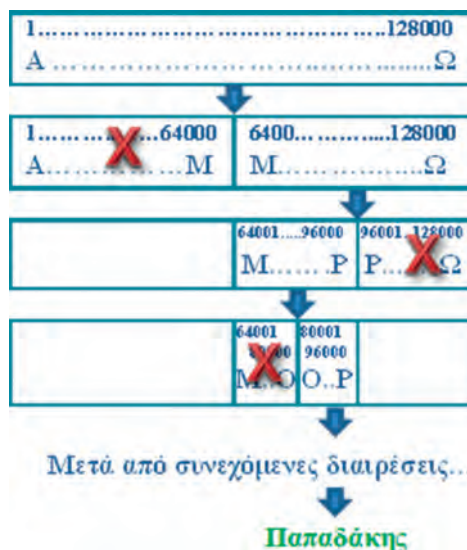
1. Αβαράκης 2103456789	X
2. Αβραμίδης 2105678234	X
3. ...	X
4. ...	X
5. ...	X

.....
.....

127996. ...	X
127997. ...	X
127998. ...	X
127999. Ψωμάς 2102575677	X
128000. Ωραίου 2108674462	✓

Εικόνα 3-1. Σειριακή αναζήτηση του ονόματος «Ωραίου»

όνομα, ο χρόνος εκτέλεσης θα είναι 10 δευτερόλεπτα. Αυτό που μας ενδιαφέρει συνήθως σε έναν αλγόριθμο είναι ο χρόνος της χειρότερης περίπτωσης που στο συγκεκριμένο παράδειγμα είναι 128000 δευτερόλεπτα και αντιστοιχεί στην αναζήτηση του τελευταίου ονόματος του τηλεφωνικού καταλόγου. Γενικεύοντας, αν ο τηλεφωνικός μας κατάλογος είχε n ονόματα, τότε ο χρόνος χειρότερης περίπτωσης θα συμβολίζονταν με $O(n)$ θέλοντας να δείξουμε τη γραμμική σχέση ανάμεσα στο πλήθος των δεδομένων εισόδου και του χρόνου εκτέλεσης του αλγορίθμου. Ο συμβολισμός $O()$ λοιπόν μας δίνει ένα πάνω όριο για τον χρόνο εκτέλεσης ενός αλγορίθμου και αποτελεί ένα πολύ καλό μέτρο της αποδοτικότητάς του.



Εικόνα 3-2. Αναδική Αναζήτηση του ονόματος «Παπαδάκης»

Μια δεύτερη λύση στο ίδιο πρόβλημα εκμεταλλεύεται την ιδιότητα της αλφαβητικής ταξινόμησης των ονομάτων του τηλεφωνικού καταλόγου. Όταν μας δίνεται ένα όνομα προς αναζήτηση, τότε το συγκρίνουμε με το όνομα που βρίσκεται στη μέση του τηλεφωνικού καταλόγου. Εάν το όνομα που ψάχνουμε προηγείται αλφαβητικά του μεσαίου ονόματος, τότε απορρίπτουμε αμέσως όλα τα ονόματα που βρίσκονται μετά τη μέση του τηλεφωνικού καταλόγου, ενώ, εάν το όνομα που ψάχνουμε έπεται αλφαβητικά του μεσαίου ονόματος, τότε απορρίπτουμε όλα τα ονόματα που βρίσκονται πριν τη μέση του τηλεφωνικού καταλόγου. Με μία και μόνη σύγκριση έχουμε «ελέγξει» τα μισά ονόματα! Στη συνέχεια εφαρμόζουμε την τακτική μας στα 64000 πρώτα ή στα 64000 τελευταία ονόματα που έμειναν, συγκρίνοντας πάλι το μεσαίο όνομα με αυτό που αναζητούμε. Μετά τη νέα σύγκριση θα μας μείνουν προς έλεγχο 32000 ονόματα και συνεχίζουμε με την ίδια λογική μέχρι να βρούμε τελικά το όνομα που ψάχνουμε. Τα βήματα που θα εκτελέσει ο αλγόριθμός μας, στη χειρότερη περίπτωση, είναι μόλις 17! Ο αριθμός αυτός μπορεί να προκύψει μαθηματικά υπολογίζοντας τον δυαδικό λογάριθμο του 128000 ($\log_2 128000$) καθώς ο αλγόριθμός μας βασίστηκε σε συνεχόμενες διαιρέσεις του πλήθους των δεδομένων

Οι περισσότεροι αλγόριθμοι έχουν χρονική πολυπλοκότητα που ανήκει σε μια από τις κατηγορίες: $O(1)$, $O(\log n)$, $O(n)$, $O(n \log n)$, $O(n^2)$, $O(2^n)$

του προβλήματός μας με το 2. Στη γενική περίπτωση λοιπόν που ο αριθμός των ονομάτων του τηλεφωνικού καταλόγου ισούται με n , ο χρόνος χειρότερης περίπτωσης θα συμβολιζόταν με $O(\log n)$ (με $\log n$ συμβολίζουμε χάριν συντομίας τον δυαδικό λογάριθμο του n). Ο αλγόριθμος που μόλις περιγράψαμε ονομάζεται δυαδική αναζήτηση και αποτελεί τον γρηγορότερο αλγόριθμο αναζήτησης, όταν τα δεδομένα μας είναι ταξινομημένα.

Πλήθος δεδομένων προβλήματος					
Πολυπλοκότητα αλγορίθμου	10	100	1000	10000	100000
$O(\log n)$	3,3 ns	6,6 ns	10 ns	13,3 ns	16,6 ns
$O(n)$	10 ns	100 ns	1000 ns	10000 ns	100000 ns
$O(n^2)$	100 ns	10000 ns	0,001 sec	0,1 sec	10 sec
$O(n^3)$	1000 ns	0,001 sec	1 sec	100 sec	278 ώρες
$O(2^n)$	1024 ns	401969368413 αιώνες			

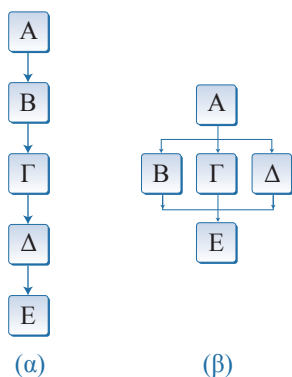
Εικόνα 3-3. Χρόνοι εκτέλεσης αλγορίθμων με χρόνο εκτέλεσης εντολής 1 ns

Η **πολυπλοκότητα αλγορίθμων** είναι ο ένας από τους δύο κλάδους του τομέα της επιστήμης των υπολογιστών που ονομάζεται **θεωρία υπολογισμού**. Ο τομέας αυτός εξετάζει αρχικά εάν μπορεί να λυθεί ένα πρόβλημα χρησιμοποιώντας κάποιο αλγόριθμο και, σε δεύτερη φάση, πόσο αποδοτικά μπορεί να λυθεί το συγκεκριμένο πρόβλημα. Ο κλάδος που αναζητά τι μπορεί να υπολογισθεί και τι όχι ονομάζεται **υπολογισμότητα**. Για την απόδειξη της υπολογισμότητας ή όχι κάποιου προβλήματος οι επιστήμονες της πληροφορικής χρησιμοποιούν διάφορα υπολογιστικά μοντέλα με το ισχυρότερο από αυτά να είναι η λεγόμενη **Μηχανή Τιούρινγκ** από το όνομα του μεγάλου βρετανού μαθηματικού Άλαν Τιούρινγκ που θεωρείται ο πατέρας της επιστήμης των υπολογιστών.

Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να περιγράφετε τι είναι η αποδοτικότητα ενός αλγορίθμου
- να αναγνωρίζετε και να συσχετίζετε τις έννοιες «πολυπλοκότητα» και «υπολογισμότητα» αλγορίθμου στην επίλυση ενός προβλήματος





Εικόνα 3-4. Σειριακή και παράλληλη εκτέλεση αλγορίθμου

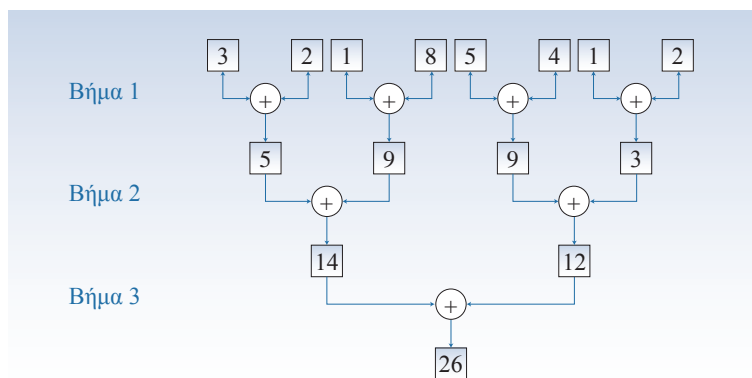
Αν θέλουμε να προσθέσουμε, χίλιους (1000) αριθμούς πόσα βήματα απαιτούνται με παράλληλη επεξεργασία; Πώς επηρεάζεται η απάντηση από το πλήθος των διαθέσιμων επεξεργαστών.

3.4 Βασικοί τύποι αλγορίθμων

Αλγόριθμοι σειριακής και παράλληλης επεξεργασίας

Οι εντολές ενός αλγορίθμου γράφονται η μια κάτω από την άλλη και εκτελούνται ακολουθιακά. Σε κάποιους αλγόριθμους, τμήματα εντολών μπορούν να εκτελεστούν ταυτόχρονα. Ας θεωρήσουμε έναν αλγόριθμο που αποτελείται από 5 τμήματα, Α έως Ε. Στην Εικόνα 3-4 (α), βλέπουμε την εκτέλεση των τμημάτων του αλγορίθμου με σειριακή επεξεργασία. Αν τα τμήματα Β, Γ, Δ δεν εξαρτώνται το ένα από το άλλο, τότε μπορούν να εκτελεστούν ταυτόχρονα, μειώνοντας δραστικά το χρόνο εκτέλεσης. Στην Εικόνα 3-4 (β), βλέπουμε την εκτέλεση των τμημάτων του αλγορίθμου με παράλληλη επεξεργασία.

Για παράδειγμα, μπορούμε να εφαρμόσουμε τη σειριακή και παράλληλη επεξεργασία στον υπολογισμό του αθροίσματος 8 τυχαίων αριθμών. Ο αλγόριθμος σειριακής επεξεργασίας απαιτεί 8 βήματα εκτέλεσης: Στην Εικόνα 3-5 βλέπουμε την εφαρμογή ενός αλγορίθμου παράλληλης επεξεργασίας για την επίλυση του ίδιου προβλήματος. Η πρόσθεση του πρώτου με τον δεύτερο, του τρίτου με τον τέταρτο, του πέμπτου με τον έκτο και του έβδομου με τον όγδοο γίνονται ταυτόχρονα σε ένα βήμα.



Εικόνα 3-5. Παράλληλη εκτέλεση αλγορίθμου

Ο αλγόριθμος σειριακής επεξεργασίας απαιτεί επτά (7) βήματα, ενώ ο αλγόριθμος παράλληλης επεξεργασίας, μόλις τρία (3).

Βήμα 1: Πρόσθεσε τον πρώτο με το δεύτερο αριθμό.

Βήμα 2: Πρόσθεσε το αποτέλεσμα με τον τρίτο αριθμό.

...

Βήμα 7: Πρόσθεσε το αποτέλεσμα με τον όγδοο αριθμό.



Αλγόριθμος σειριακής επεξεργασίας είναι ένας αλγόριθμος του οποίου τα βήματα εκτελούνται ακολουθιακά το ένα μετά το άλλο, από έναν επεξεργαστή.

Αλγόριθμος παράλληλης επεξεργασίας είναι ένας αλγόριθμος του οποίου τα βήματα μπορούν να εκτελούνται ταυτόχρονα από διαφορετικούς επεξεργαστές, μειώνοντας το χρόνο εκτέλεσης.



Επαναληπτικοί αλγόριθμοι

Ένα από τα μεγάλα πλεονεκτήματα της χρήσης των υπολογιστών για την επίλυση προβλημάτων είναι η δυνατότητα επανάληψης μιας διαδικασίας πολλές φορές. Ο άνθρωπος, έστω και αν μπορεί να εκτελέσει μια ενέργεια, κουράζεται με την επανάληψή της, κάνει λάθη. Δεν ισχύει το ίδιο για τον υπολογιστή. Γι' αυτό και οι επαναληπτικοί αλγόριθμοι είναι πολύ διαδεδομένοι στους υπολογιστές.

Ο υπολογισμός του αθροίσματος δύο αριθμών είναι μια απλή πρόσθεση. Όταν, όμως, έχουμε χίλιους αριθμούς, τότε η πράξη της πρόσθεσης πρέπει να εκτελεστεί χίλιες φορές. Αυτή είναι μια επαναληπτική διαδικασία, η οποία μπορεί να εκφραστεί ως εξής:

Σε τρεις γραμμές ο αλγόριθμος έχει ολοκληρωθεί. Ο βρόχος μείωσε δραστικά το πλήθος των απαιτούμενων εντολών χωρίς να επηρεάσει την αποτελεσματικότητα του αλγορίθμου.

Μια επαναληπτική διαδικασία λέγεται **βρόχος**.

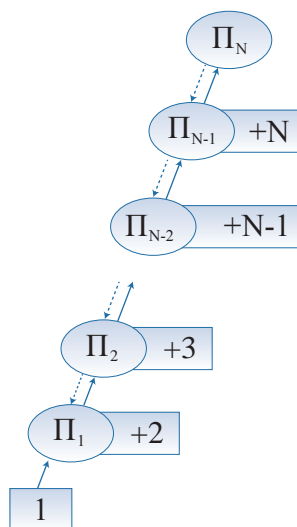
Ένας βρόχος που δεν σταματά ποτέ λέγεται **ατέρμων**.

Αναδρομικοί αλγόριθμοι

Η αναδρομή είναι μια ιδιαίτερα διαισθητική αλγοριθμική μέθοδος. Η κατανόηση των αναδρομικών αλγορίθμων δεν είναι εύκολη υπόθεση. Απαιτεί ιδιαίτερο τρόπο σκέψης που κατακτάται μόνο με έντονη εξάσκηση. Ένας αναδρομικός αλγόριθμος χρησιμοποιεί τον ίδιο του τον εαυτό.



Αναδρομικός είναι ο αλγόριθμος που καλεί άμεσα ή έμμεσα τον εαυτό του μία ή περισσότερες φορές, επιλύοντας κάθε φορά ένα πρόβλημα της ίδιας φύσης με το αρχικό, αλλά μικρότερου μεγέθους.



Εικόνα 3-6. Αναδρομικός αλγόριθμος

Αν συμβολίσω το γινόμενο $1*2*...*N$ με Π_N , τότε προκύπτει ο εξής μαθηματικός αναδρομικός ορισμός:

$\Pi_N = 1$, αν $N=1$

$\Pi_N = \Pi_{N-1} * N$, αν $N>1$

με τη χρήση του οποίου μπορώ να υπολογίσω αναδρομικά οποιοδήποτε γινόμενο Π_N .

Για παράδειγμα, ο υπολογισμός του γινομένου $\Pi = 1 \cdot 2 \cdot 3 \cdot \dots \cdot N$ μπορεί να γίνει αναδρομικά. Ας προσπαθήσουμε να λύσουμε ένα στιγμιότυπο του προβλήματος, τον υπολογισμό του $1 \cdot 2 \cdot \dots \cdot 6$ και μετά να γενικεύσουμε τη λύση, αντικαθιστώντας το 6 με N . Αναπτύσσουμε το εξής σκεπτικό: Για να υπολογίσω το γινόμενο $1 \cdot 2 \cdot 3 \cdot \dots \cdot 6$, αρκεί να πολλαπλασιάσω τον αριθμό 6 με το γινόμενο $1 \cdot 2 \cdot \dots \cdot 5$. Για να υπολογίσω το γινόμενο $1 \cdot 2 \cdot 3 \cdot \dots \cdot 5$, αρκεί να πολλαπλασιάσω τον αριθμό 5 με το γινόμενο $1 \cdot 2 \cdot \dots \cdot 4$. Συνεχίζοντας την απλούστευση του προβλήματος, θα φθάσω στο πρόβλημα υπολογισμού του γινομένου $1 \cdot 2$ που είναι τετριμμένο. Ακολουθώντας, τώρα, την αντίθετη φορά, έχοντας υπολογίσει το $1 \cdot 2$, υπολογίζω το $(1 \cdot 2) \cdot 3$, μετά το $((1 \cdot 2) \cdot 3) \cdot 4$, κ.ο.κ., μέχρι να φθάσω στο $((1 \cdot 2) \cdot \dots) \cdot 6$. Μπορώ να γενικεύσω το παραπάνω σκεπτικό για οποιοδήποτε ακέραιο N .

Η δημιουργία και ο έλεγχος ορθότητας αναδρομικών αλγορίθμων είναι ακόμα πιο δύσκολες διεργασίες που απαιτούν προγραμματιστική εμπειρία. Παρόλ' αυτά, η αναδρομή είναι πολύ ισχυρή μέθοδος, που δίνει τη δυνατότητα να περιγράψουμε με σύντομο τρόπο τη λύση δύσκολων προβλημάτων.



Ερωτήσεις - Θέματα για συζήτηση

A. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Σε κάποιους αλγόριθμους, τμήματα εντολών μπορούν να εκτελεστούν ταυτόχρονα.
2. Η εκτέλεση τμημάτων ενός αλγόριθμου με παράλληλη επεξεργασία, μειώνει δραστικά το χρόνο εκτέλεσης.
3. Η αξιοποίηση της παραλληλίας μπορεί να εφαρμοσθεί σε προβλήματα που απαιτούν ομοιόμορφη επεξεργασία πολλών δεδομένων.
4. Η παράλληλη επεξεργασία μπορεί να εφαρμοσθεί σε όλα τα προβλήματα.

B. Συμπληρώστε τις προτάσεις με την κατάλληλη λέξη:

1. Αλγόριθμος σειριακής επεξεργασίας είναι ένας αλγόριθμος του οποίου τα βήματα εκτελούνται _____ το ένα μετά το άλλο, από έναν _____.
2. Αλγόριθμος παράλληλης επεξεργασίας είναι ένας αλγόριθμος του οποίου τα βήματα μπορούν να εκτελούνται _____ από διαφορετικούς _____, μειώνοντας το χρόνο εκτέλεσης.
3. Για τον υπολογισμό του αθροίσματος 8 τυχαίων αριθμών, ο αλγόριθμος σειριακής επεξεργασίας απαιτεί _____ βήματα, ενώ ο αλγόριθμος παράλληλης επεξεργασίας, μόλις _____.

Ασκήσεις - Προβλήματα

1. Αναλύστε τον υπολογισμό της αριθμητικής παράστασης $((x-y)*(x+y))/((2+x)*y)$ σε απλές εντολές μιας πράξης. Ποιες εντολές μπορούν να εκτελεστούν ταυτόχρονα; Πόσα βήματα απαιτούνται για τη σειριακή εκτέλεση από έναν επεξεργαστή; Πόσα βήματα απαιτούνται κατ' ελάχιστο για την παράλληλη επεξεργασία από τρεις επεξεργαστές;
2. Αναζητήσετε το «Πρόβλημα των πύργων του Ανόι». Επιλύστε το πρόβλημα για 3 δίσκους και εκφράστε τη λύση του με φυσική γλώσσα. Χρησιμοποιήστε τη λύση που κατασκευάσατε, για να λύσετε το πρόβλημα για 4 δίσκους. Γενικεύσετε τη λύση για οποιοδήποτε πλήθος δίσκων. Εκφράσετε τη λύση σας αναδρομικά.

**Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:**

- να αναγνωρίζετε την ύπαρξη συγκεκριμένων χαρακτηριστικών και τύπων αλγορίθμων
- να διακρίνετε έναν αλγόριθμο σειριακής επεξεργασίας από έναν αλγόριθμο παράλληλης επεξεργασίας
- να αναγνωρίζετε τις αλληλοεξαρτήσεις μεταξύ των εντολών ενός σειριακού αλγορίθμου και να εντοπίζετε ποια βήματα ενός αλγορίθμου που μπορούν να εκτελεστούν παράλληλα.
- να προσδιορίζετε την έννοια της επαναληπτικής διαδικασίας και του βρόχου
- να αναγνωρίζετε πότε ένα πρόβλημα απαιτεί επαναληπτικό αλγόριθμο για την επίλυσή του.
- να διατυπώνετε σε φυσική γλώσσα έναν επαναληπτικό αλγόριθμο.
- να αναγνωρίζετε πότε ένας αλγόριθμος είναι αναδρομικός.



3.5 Αναπαράσταση αλγορίθμου

Η αναπαράσταση ενός αλγορίθμου μπορεί να γίνει με διάφορους τρόπους. Ο τρόπος περιγραφής σε συνδυασμό με αυτόν που θα κληθεί να τον εκτελέσει, επηρεάζουν την ικανοποίηση ή μη των αλγοριθμικών κριτηρίων. Επιπλέον, ο ίδιος αλγόριθμος, αν εκφρασθεί με διαφορετικούς τρόπους, μπορεί να ικανοποιεί ή όχι κάποια αλγοριθμικά κριτήρια. Για την αναπαράσταση των αλγορίθμων χρησιμοποιούνται διάφοροι τρόποι όπως η φυσική γλώσσα, το διάγραμμα ροής, οι γλώσσες περιγραφής αλγορίθμων και οι γλώσσες προγραμματισμού.

Η **φυσική γλώσσα** αποτελεί τον πιο απλό και ανεπεξέργαστο τρόπο παρουσίασης ενός αλγορίθμου, που με απλά λόγια και ελεύθερες εκφράσεις περιγράφουμε τα βήματα. Ωστόσο, ο συγκεκριμένος τρόπος έκφρασης ενέχει αυξημένη πιθανότητα λάθους ή ασάφειας. Η φυσική γλώσσα ενδείκνυται μόνο σε καθημερινά προβλήματα στα οποία η ακρίβεια και η σαφήνεια δεν παίζουν σημαντικό ρόλο στην καθοριστικότητα και αποτελεσματικότητα του τρόπου επίλυσής τους.

Το **διάγραμμα ροής** είναι η αναπαράσταση του αλγορίθμου με τη χρήση γεωμετρικών σχημάτων. Είναι ο πλέον εποπτικός τρόπος παρουσίασης. Σε ένα διάγραμμα ροής μπορούμε, με μια ματιά, να αναγνωρίσουμε τις λογικές δομές που περιλαμβάνει ο αλγόριθμος. Από την άλλη, η χρήση σχημάτων περιορίζει τη δυνατότητα έκφρασης σε ένα σχετικά μικρό χώρο, όπως είναι μια εκτυπώσιμη σελίδα. Αυτό καθιστά σχεδόν αδύνατη την αναπαράσταση ενός αλγορίθμου εκατοντάδων ή χιλιάδων εντολών με ένα διάγραμμα ροής. Ο περιορισμός αυτός δεν υπάρχει σε μια γλώσσα. Τα σχήματα που χρησιμοποιούνται στα διαγράμματα του παρόντος βιβλίου και η έννοιά τους, φαίνονται στον παρακάτω πίνακα:

Σχήμα	Έννοια
	Ροή (οδηγεί στην επόμενη εντολή)
	Έλλειψη
	Πλάγιο - παραλληλόγραμμο
	Ορθογώνιο παραλληλόγραμμο
	Ρόμβος
	Αρχή - Τέλος αλγορίθμου
	Είσοδος - Έξοδος
	Επεξεργασία - Εκτέλεση πράξεων
	Συνθήκη

Εικόνα 3-7. Σύμβολα διαγραμμάτων ροής

Οι **γλώσσες περιγραφής αλγορίθμων** αποτελούν ενδιάμεσο στάδιο αναπαράστασης. Προσεγγίζουν τις γλώσσες προγραμματισμού, είναι όμως απλούστερες. Στόχος τους είναι η κωδικοποιημένη αποτύπωση αλγορίθμων σε κείμενο, χωρίς, όμως, να ενδιαφέρουν οι λεπτομέρειες υλοποίησής τους από έναν υπολογιστή. Γι' αυτό και αναφέρονται ως ψευδογλώσσες ή ψευδοκώδικες.

Οι γλώσσες προγραμματισμού αποτελούν τις γλώσσες επικοινωνίας του ανθρώπου με τον υπολογιστή. Η προσπάθεια του ανθρώπου να αναθέσει την επίλυση προβλημάτων σε υπολογιστές οδήγησε στη δημιουργία γλωσσών που έχουν περιορισμένο σύνολο κωδικοποιημένων εντολών με αυστηρούς κανόνες σύνταξης. Με τη χρήση τους, ο άνθρωπος δίνει εντολές στον υπολογιστή και αυτός τις εκτελεί.

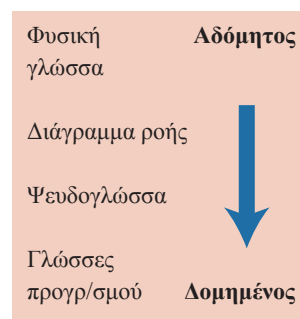
Ας δούμε τον αλγόριθμο υπολογισμού περιφέρειας και εμβαδού ενός κύκλου. Στην Εικόνα 3-8, ο ίδιος αλγόριθμος έχει εκφρασθεί με τρεις διαφορετικούς τρόπους. Ανάμεσα στους τρεις τρόπους υπάρχει αντιστοιχία των εντολών. Γενικά, είναι εύκολο να μετατρέψουμε έναν αλγόριθμο από μια μορφή αναπαράστασης σε μια άλλη. Αν γνωρίζουμε καλά μια γλώσσα προγραμματισμού, τότε είναι εύκολο να μάθουμε και μια δεύτερη, αρκεί να έχουμε κατανοήσει την αντιστοιχία των εντολών τους.

Φυσική Γλώσσα	Ψευδογλώσσα	Διάγραμμα ροής
Αρχή αλγορίθμου	Αλγόριθμος Κύκλος	Αρχή
Ρώτα να μάθεις πόσο είναι η ακτίνα.	Γράψε 'Δώσε ακτίνα'	Διάβασε ακτίνα
Πάρε την ακτίνα.	Διάβασε ακ	περιφέρεια ← ακτίνα*2*3.14
Πολλαπλασίασε την ακτίνα με το 2 και ότι βρεις με το 3.14. Το αποτέλεσμα είναι η περιφέρεια του κύκλου.	περ ← ακ*2*3.14	εμβαδό ← ακτίνα^2*3.14
Ύψωσε την ακτίνα στο 2 και πολλαπλασίασε ότι βρεις με το 3.14. Το αποτέλεσμα είναι το εμβαδό του κύκλου.	εμβ ← ακ^2*3.14	Γράψε περιφέρεια, εμβαδό
Πες την περιφέρεια και το εμβαδό που υπολόγισες.	Γράψε περ, εμβ	Τέλος
Τέλος αλγορίθμου	Τέλος Κύκλος	

Εικόνα 3-8. Τρεις αναπαράστασεις του ίδιου αλγορίθμου

Ποιος από τους τρόπους αναπαράστασης είναι ο καλύτερος; Πόσο εύκολη είναι η δημιουργία διαγραμμάτων στον υπολογιστή; Ένα πρόγραμμα σε γλώσσα προγραμματισμού μπορεί να κατανοηθεί και εκτελεστεί από οποιονδήποτε άνθρωπο; Ένα κείμενο σε φυσική γλώσσα μπορεί να εκτελεστεί άμεσα από τον υπολογιστή; Οι απαντήσεις δεν είναι άμεσες ούτε εύκολες. Η φύση και το μέγεθος του προβλήματος, καθορίζουν τον καταλληλότερο τρόπο αναπαράστασης.

Συνοψίζοντας τους τρόπους αναπαράστασης αλγορίθμων, παρατηρούμε ότι ο βαθμός δόμησης του τρόπου αναπαράστασης αυξάνεται καθώς πηγαίνουμε από τον πλούτο και την ελευθερία της φυσικής γλώσσας προς τους αυστηρούς λεξικολογικούς και συντακτικούς κανόνες των γλωσσών προγραμματισμού. **Η περιγραφή αλγορίθμου σε φυσική γλώσσα είναι ο λιγότερο δομημένος, ενώ οι γλώσσες προγραμματισμού ο πιο δομημένος τρόπος αναπαράστασης αλγορίθμων.**





Ερωτήσεις - Θέματα για συζήτηση

A. Γράψτε τον αριθμό της πρότασης και δίπλα το γράμμα Σ εάν θεωρείτε ότι η πρόταση είναι σωστή ή το γράμμα Λ εάν θεωρείτε ότι είναι λανθασμένη.

1. Ένας αλγόριθμος μπορεί να εκφρασθεί με πολλούς διαφορετικούς τρόπους.
2. Η φυσική γλώσσα αποτελεί τον πιο απλό και ανεπεξέργαστο τρόπο παρουσίασης ενός αλγορίθμου.
3. Ένα κείμενο σε φυσική γλώσσα μπορεί να εκτελεστεί άμεσα από τον υπολογιστή.
4. Η περιγραφή του αλγορίθμου σε φυσική γλώσσα περιέχει απλά λόγια και ελεύθερες εκφράσεις.
5. Η περιγραφή του αλγορίθμου σε διάγραμμα ροής περιέχει σχήματα.
6. Η αναπαράσταση του αλγορίθμου με σχήματα είναι ο λιγότερο εποπτικός τρόπος παρουσίασης.
7. Οι γλώσσες προγραμματισμού είναι οι γλώσσες επικοινωνίας του ανθρώπου με τον υπολογιστή.
8. Ο καλύτερος τρόπος αναπαράστασης είναι η γλώσσα προγραμματισμού.
9. Ένα πρόγραμμα σε γλώσσα προγραμματισμού C μπορεί να κατανοηθεί και εκτελεστεί από οποιονδήποτε άνθρωπο.

B. Αντιστοιχίστε κάθε στοιχείο της στήλης A με την κατάλληλη έννοια της στήλης B

1.

Στήλη A	Στήλη B
Φυσική γλώσσα	Πιο δομημένος
Διάγραμμα ροής	Πιο αδόμητος
Γλώσσες περιγραφής αλγορίθμων	Πιο εποπτικός

2.

Στήλη A	Στήλη B
Βέλος	Επεξεργασία
Έλλειψη	Είσοδος - Έξοδος
Πλάγιο παραλληλόγραμμο	Συνθήκη
Ορθογώνιο παραλληλόγραμμο	Ροή (επόμενη εντολή)
Ρόμβος	Αρχή - Τέλος αλγορίθμου

Γ. Συμπληρώστε τις προτάσεις με την κατάλληλη λέξη:

1. Τα Κινέζικα είναι μια _____ γλώσσα.
2. Η _____ αποτελεί τον πιο απλό και ανεπεξέργαστο τρόπο παρουσίασης ενός αλγορίθμου.
3. Το _____ λεί μια εύληπτη και παραστατική παρουσίαση αλγορίθμου σε δισδιάστατη μορφή.
4. Οι _____ αποτελούν κωδικοποιημένη αποτύπωση αλγορίθμου με κείμενο.
5. Οι _____ αποτελούν αποτύπωση αλγορίθμου με περιορισμένο σύνολο κωδικοποιημένων εντολών, κατανοητών από τον υπολογιστή.
6. Οι γλώσσες περιγραφής αλγορίθμων αναφέρονται και ως _____

Ασκήσεις - Προβλήματα

1. Γράψτε αλγόριθμο που δέχεται τον προφορικό και το γραπτό βαθμό ενός μαθητή υπολογίζει το μέσο όρο του και εξάγει αποτέλεσμα προαγωγής. Ο μαθητής προάγεται αν έχει μέσο όρο πάνω από 9.5, διαφορετικά παραπέμπεται για επανεξέταση. Εκφράστε τον αλγόριθμο και με τους τρεις τρόπους αναπαράστασης, φυσική γλώσσα, διάγραμμα ροής και Ψευδογλώσσα.



Όταν ολοκληρώσετε την ενότητα ελέγξτε αν είστε σε θέση:

- να αναγνωρίζετε τις διάφορες μορφές αναπαράστασης του αλγορίθμου.
- να συγκρίνετε και να επιλέγετε τον κατάλληλο τρόπο αναπαράστασης ανάλογα με το πρόβλημα και τον αλγόριθμο.
- μο
- να εντοπίζετε και να περιγράφετε τη λειτουργία σχολίων σε απλά παραδείγματα αλγορίθμων.



