

Εισαγωγή στις Αρχές της επιστήμης των Η/Υ

B Λυκείου

Επιστήμη των υπολογιστών

μελετά

Θεωρητικά
θεμέλια

Φύση
πληροφοριών

Φύση
αλγορίθμων

Φύση
υπολογισμών

Και τις
τεχνολογικές
εφαρμογές
τους

Από τις
σκοπιές

Σχεδίασης

Ανάπτυξης

Υλοποίησης

Διερεύνησης

ανάλυσης

Η Επιστήμη των υπολογιστών
διακρίνεται σε

Θεωρητική

Εφαρμοσμένη

Επιστήμη
Υπολογιστών

Θεωρητική
ερευνά τον σχεδιασμό

Εφαρμοσμένη

αλγορίθμων

Υπολ.
μεθόδων

Που χρησιμοποιούνται για

Άντληση
πληροφοριών

Επεξεργασία
πληροφοριών

Αποθήκευση
πληροφοριών

Ανάλυση
πληροφοριών

ΕΠΙΣΤΗΜΗ ΤΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

Εφαρμοσμένη

Θεωρητική
(βασικές έννοιες)

Ανάλυση αλγορίθμων

Θεωρία
υπολογισιμότητας

Θεωρία
πολυπλοκότητας

Θεωρία
γλ.προγραμματισμού

ΕΠΙΣΤΗΜΟΛΟΓΙΑ

Θεωρητική

Εφαρμοσμένη
(επιστημονικά πεδία)

Κατασκευή Η/Υ

Λογισμικό

Πληροφοριακά
συστήματα

Τεχνητή νοημοσύνη

Δίκτυα

Λειτ. συστήματα

Βάσεις δεδομένων

Ασφάλεια Η/Υ

2.1. Πρόβλημα

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

Π.χ

- Μέτρηση χρόνου (στην αρχαιότητα)
- Κοινωνικά προβλήματα (ναρκωτικά, ανεργία)
- Φυσικά φαινόμενα (σεισμοί, πλημμύρες, επιδημίες)
- Ενεργειακό πρόβλημα
- Προστασία φυσικού περιβάλλοντος (τρύπα όζοντος)
- Τεχνολογικά προβλήματα (Millennium Bug)

2.1.2 Κατηγορίες Προβλημάτων

Βάσει της δυνατότητας επίλυσης τους:

- **Επιλύσιμα**
εκείνα τα προβλήματα για τα οποία η λύση έχει βρεθεί και έχει διατυπωθεί. **π.χ επίλυση της δευτεροβάθμιας εξίσωσης**
- **Μη επιλύσιμα**
τα προβλήματα για τα οποία έχει αποδειχτεί, ότι δεν επιδέχονται λύση. **Π.χ Το πρόβλημα του τετραγωνισμού του κύκλου**
- **Ανοικτά**
τα προβλήματα για τα οποία η λύση τους δεν έχει ακόμα βρεθεί, ενώ ταυτόχρονα δεν έχει αποδειχτεί, ότι δεν επιδέχονται λύση.

2.1.3 Υπολογιστικά Προβλήματα

Τα προβλήματα με βάση τη δυνατότητα επίλυσής τους μέσω του υπολογιστή διακρίνονται 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.

2.1.3 Υπολογιστικά Προβλήματα

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.

2.1.3 Υπολογιστικά Προβλήματα

Τα μη υπολογιστικά προβλήματα δεν μπορούν να λυθούν από έναν υπολογιστή ή από άλλα μηχανικά μέσα

2.1.4 Διαδικασίες επίλυσης (υπολογιστικού) προβλήματος



Επίλυση προβλήματος - Κατανόηση

Η κατανόηση ενός προβλήματος αποτελεί συνάρτηση δύο παραγόντων :

- **Σωστής διατύπωσης** από τον δημιουργό του προβλήματος
- **Σωστής ερμηνείας** από αυτόν που θα λύσει το πρόβλημα

Επίλυση προβλήματος – Ανάλυση Αφαίρεση

Με την Ανάλυση διασπάμε το πρόβλημα σε άλλα απλούστερα προβλήματα για να είναι εύκολη η αντιμετώπισή τους.

Η ανάλυση γίνεται **φραστικά** ή **διαγραμματικά**.

Πρόβλημα : Εξυπηρέτηση πολιτών από τις υπηρεσίες του δημοσίου

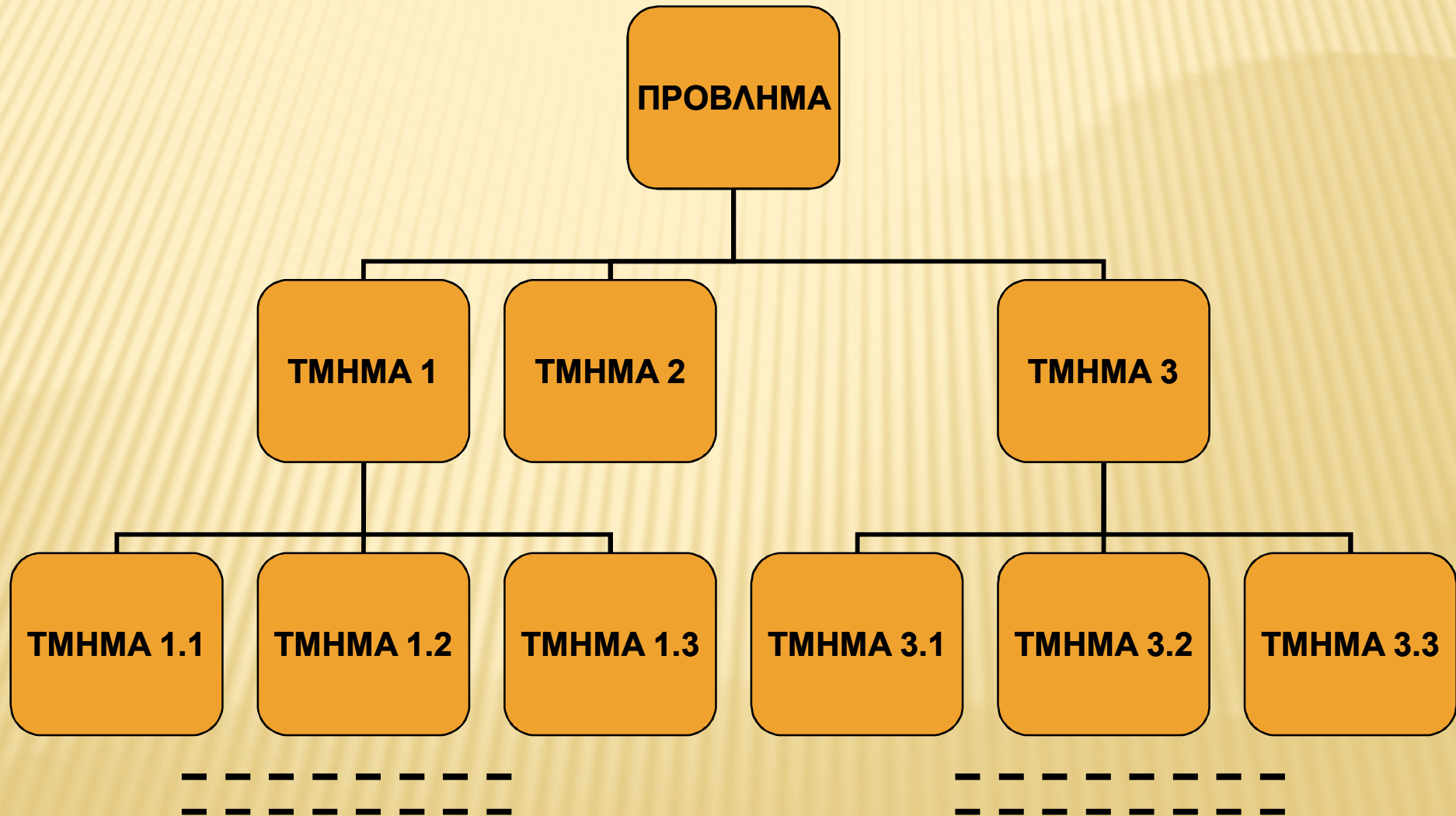
Επίλυση προβλήματος – Ανάλυση Αφαίρεση

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.

Επίλυση προβλήματος – Ανάλυση Αφαίρεση

Διαγραμματική ανάλυση



Για τη σωστή επίλυση ενός προβλήματος :

- Προσδιορίζουμε τα δεδομένα που μας παρέχονται
- Καταγράφουμε τα ζητούμενα

Δεν είναι πάντα εύκολο να διακρίνουμε τα δεδομένα. Δεν υπάρχει συγκεκριμένη μεθοδολογία.

Το ίδιο ισχύει και για τα ζητούμενα.

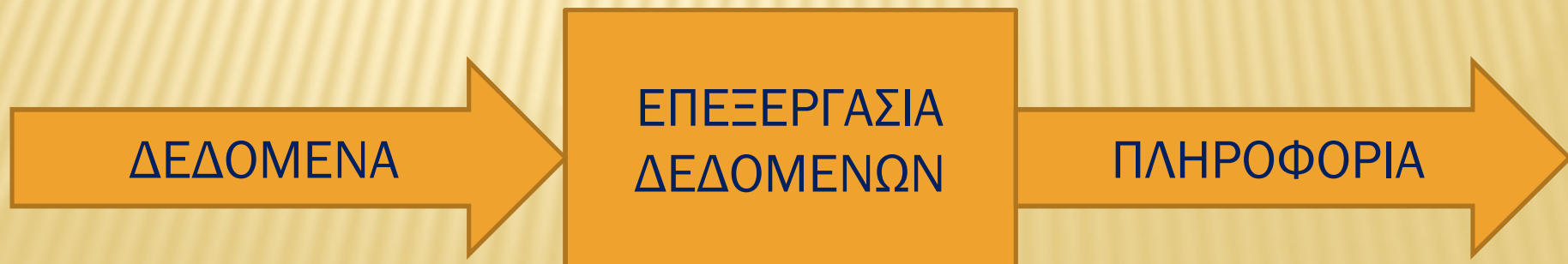
Για να βρει κάποιος τα ζητούμενα χρειάζεται να επεξεργαστεί τα δεδομένα.

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.

Ζητούμενο είναι οτιδήποτε προκύπτει ή τίθεται ως αντικείμενο έρευνας ή αναζήτησης.

Πληροφορία είναι οποιοδήποτε γνωσιακό στοιχείο προέρχεται από επεξεργασία δεδομένων.



Επίλυση προβλήματος – Σύνθεση – Κατηγοριοποίηση - Γενίκευση

Σύνθεση είναι η κατασκευή μιας νέας δομής, με την οργάνωση των επιμέρους στοιχείων του προβλήματος.

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».

Με την **γενίκευση**, μπορούν να μεταφερθούν τα αποτελέσματα σε άλλες παρεμφερείς καταστάσεις ή προβλήματα.

Αλγόριθμοι

- Η λέξη προέρχεται από έναν Πέρση μαθηματικό του 825 μ.Χ., τον Abu Jafar Mohammed ibn Musa .
- Σήμερα, το πεδίο της μελέτης των αλγορίθμων (το οποίο καλείται θεωρία αλγορίθμων) είναι ένα ιδιαίτερα ευρύ πεδίο έρευνας.
- Η έννοια του αλγορίθμου δεν συνδέεται αποκλειστικά και μόνο με προβλήματα της Πληροφορικής.
- Η αλληλουχία των ενεργειών οδηγεί στο επιθυμητό αποτέλεσμα.



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

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

1. **Αλγόριθμος** Ευκλείδης
2. **Διάβασε** x, y
3. $z \leftarrow y$
4. Όσο $z \neq 0$ **επανάλαβε**
5. $z \leftarrow x \bmod y$
6. $x \leftarrow y$
7. $y \leftarrow z$
8. **Τέλος_επανάληψης**
9. **Εμφάνισε** x
10. **Τέλος** Ευκλείδης

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.

Πίνακας 2.2: Κατηγορίες πολυπλοκότητας αλγορίθμων

Πολυπλοκότητα	Ονομασία	Παρατηρήσεις
$O(1)$	Σταθερή	Κάθε βήμα εκτελείται μια φορά ή το πολύ μερικές φορές
$O(\log n)$	Λογαριθμική	Αν δεν σημειώνεται αλλιώς, οι λογάριθμοι είναι δυαδικοί (Δυαδική Αναζήτηση)
$O(n)$	Γραμμική	Σειριακή αναζήτηση
$O(n \log n)$		Γρήγορη Ταξινόμηση
$O(n^2)$	Τετραγωνική	Ταξινόμηση με επιλογή
$O(n^3)$	Κυβική	Πολλαπλασιασμός πινάκων
$O(2^n)$	Εκθετική	Μερισμός συνόλου

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**.

2.2.2 Χαρακτηριστικά αλγορίθμου

Κάθε αλγόριθμος είναι σημαντικό να έχει ορισμένα χαρακτηριστικά προκειμένου να θεωρείται πλήρης

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 σε πεπερασμένο χρόνο.

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, π.χ εντολή **Διάβασε***

Έξοδος: Κάθε αλγόριθμος χρειάζεται να δημιουργεί κάποιο αποτέλεσμα.

Το αποτέλεσμα του αλγορίθμου, η έξοδος του, είναι μία ή περισσότερες μεταβλητές ή/και σταθερές τιμές.

Η έξοδος μπορεί να επιτευχθεί με κατάλληλες εντολές

*π.χ εντολή **Εμφάνισε***

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

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

Το πεδίο της θεωρίας υπολογισμού

Χωρίζεται στους κλάδους

Θεωρία Υπολογισιμότητας

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

Θεωρία Πολυπλοκότητας

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

Για κάθε αλγόριθμο απαιτείται **ανάλυση** μέσω της οποίας :

- ❑ **τεκμηριώνεται η ορθότητά του**, δηλαδή αν ο αλγόριθμος κάνει πραγματικά τη δουλειά για την οποία έχει σχεδιαστεί και
- ❑ **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 πολυπλοκότητα ενός αλγορίθμου δίνει ένα μέτρο της χρονικής καθυστέρησης του αλγορίθμου για την επίλυση ενός προβλήματος.

Πίνακας 2.2: Κατηγορίες πολυπλοκότητας αλγορίθμων

Πολυπλοκότητα	Ονομασία	Παρατηρήσεις
$O(1)$	Σταθερή	Κάθε βήμα εκτελείται μια φορά ή το πολύ μερικές φορές
$O(\log n)$	Λογαριθμική	Αν δεν σημειώνεται αλλιώς, οι λογάριθμοι είναι δυαδικοί (Δυαδική Αναζήτηση)
$O(n)$	Γραμμική	Σειριακή αναζήτηση
$O(n \log n)$		Γρήγορη Ταξινόμηση
$O(n^2)$	Τετραγωνική	Ταξινόμηση με επιλογή
$O(n^3)$	Κυβική	Πολλαπλασιασμός πινάκων
$O(2^n)$	Εκθετική	Μερισμός συνόλου

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**.

2.2.4 Βασικοί τύποι αλγορίθμων

- **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).

2.2.5 Αναπαράσταση αλγορίθμου

Γίνεται με :

Φυσική γλώσσα όπου η αναπαράσταση γίνεται με την ομιλούμενη γλώσσα, μέσω της οποίας περιγράφονται τα βήματα επίλυσης του προβλήματος. *Παρατηρούνται όμως ασάφειες στις οδηγίες.*

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.

2.2.5 Αναπαράσταση αλγορίθμου

Γίνεται με :

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.

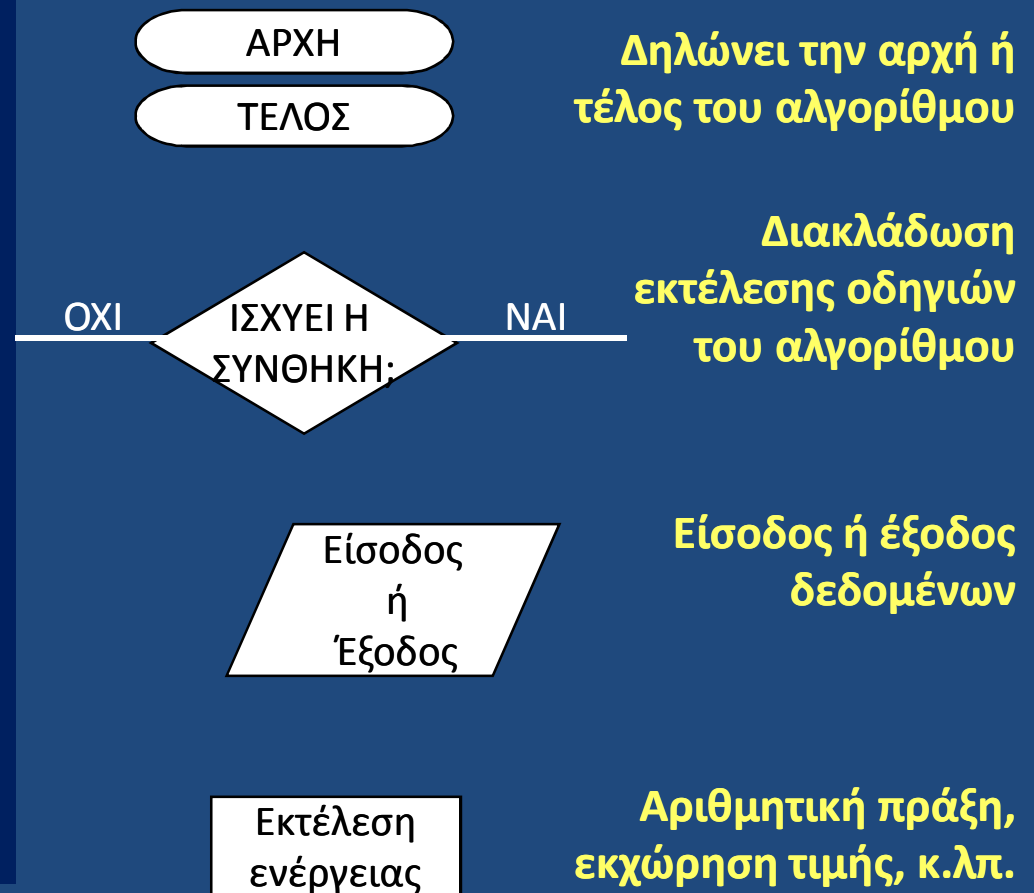
2.2.5 Αναπαράσταση αλγορίθμου

Γίνεται με :

Μεθοδολογίες διαγραμματικής αναπαράστασης αλγορίθμων

Είναι ένας γραφικός τρόπος παρουσίασης του αλγόριθμου .

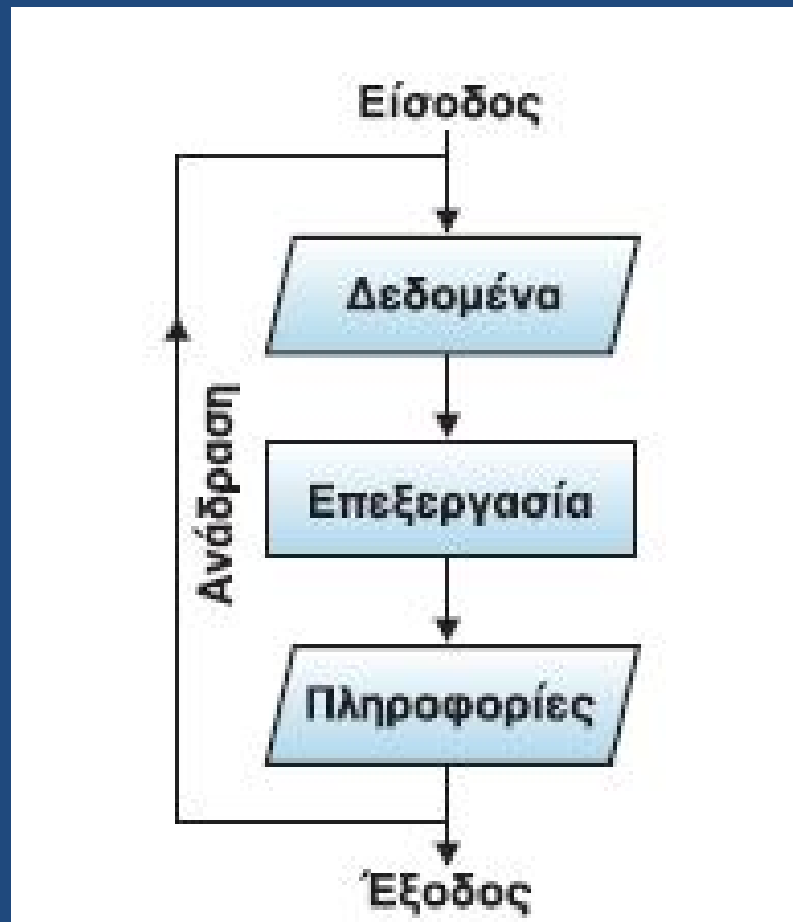
Η πιο διαδεδομένη μεθοδολογία είναι το διάγραμμα ροής.



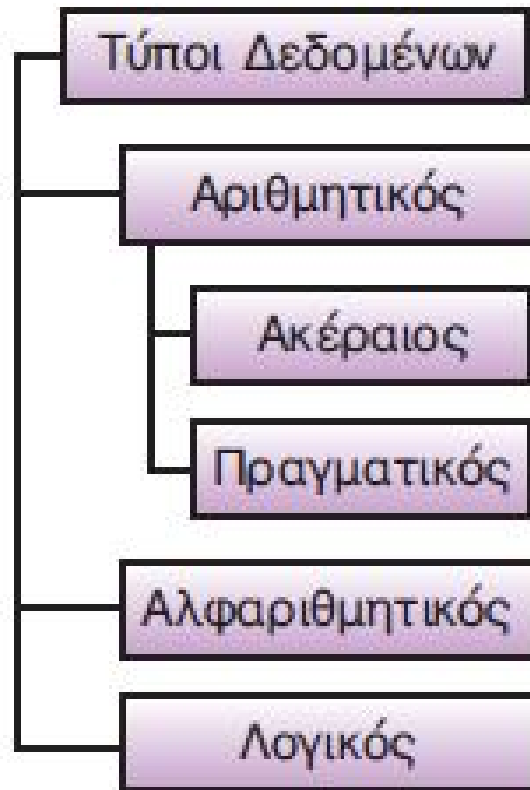
Όλα τα σχήματα ενώνονται με γραμμές που καταλήγουν σε βέλος

2.2.6 Δεδομένα και αναπαράστασή τους

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



2.2.6 Δεδομένα και αναπαράστασή τους



Εικόνα 2.12. Τύποι Δεδομένων

2.2.6 Δεδομένα και αναπαράστασή τους

- **Δεδομένα**

- Ακατέργαστα στοιχεία (π.χ. ονόματα, ποσά, ημερομηνίες, βάρος, φύλο, κλπ)
- Ως αυτόνομες τιμές, δεν έχουν κάποια αξία

- **Πληροφορίες**

- Είναι το αποτέλεσμα της επεξεργασίας των δεδομένων, ώστε να χρησιμοποιηθεί για τις κατάλληλες αποφάσεις

- **Αλγόριθμος**

- Ο τρόπος εξαγωγής της πληροφορίας (επεξεργασία) από τη χρήση των δεδομένων

2.2.6 Δεδομένα και αναπαράστασή τους

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

Οι πιο γνωστές χρησιμοποιούμενες δομές δεδομένων είναι :

- ο πίνακας
- η στοίβα
- η ουρά
- η λίστα
- το δένδρο
- και ο γράφος

2.2.6 Δεδομένα και αναπαράστασή τους

Ο **πίνακας (table)** αποτελείται από ένα σύνολο ομοειδών απλών στοιχείων, καθένα από τα οποία καθορίζεται με τη βοήθεια ενός ή περισσότερων δεικτών

Δισδιάστατος πίνακας A 3X4

12	14	20	8
17	15	11	19
7	17	19	12

Στοιχείο
[1,4]

Στοιχείο [3,1]

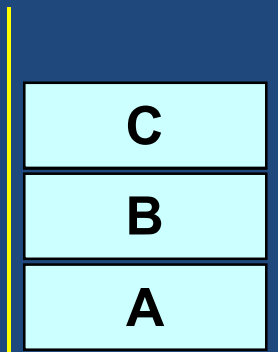
Στοιχείο [3,3]

2.2.6 Δεδομένα και αναπαράστασή τους

Η **στοίβα (stack)** είναι μια γραμμική διάταξη στοιχείων, στην οποία εισάγονται και εξάγονται στοιχεία μόνο από το ένα άκρο

ΕΙΣΕΡΧΟΜΕΝΑ
ΣΤΟΙΧΕΙΑ

ΕΞΕΡΧΟΜΕΝΑ
ΣΤΟΙΧΕΙΑ



ΣΤΟΙΒΑ ΜΕ ΤΡΙΑ ΣΤΟΙΧΕΙΑ

ΚΥΡΙΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΣΕ ΜΙΑ ΣΤΟΙΒΑ

- **ΩΘΗΣΗ (PUSH)** ενός στοιχείου στην κορυφή της στοίβας .
- **ΑΠΩΘΗΣΗ (POP)** ενός στοιχείου από την στοίβα .

κάθε στοιχείο που εισάγεται τελευταίο θα εξαχθεί πρώτο.

Αυτή η μέθοδος επεξεργασίας λέγεται **LIFO (Last in First Out)**

2.2.6 Δεδομένα και αναπαράστασή τους

Η **ουρά (queue)** αποτελεί μια γραμμική διάταξη στοιχείων, στην οποία εισάγονται νέα στοιχεία από ένα άκρο και εξάγονται υπάρχοντα στοιχεία από το άλλο άκρο



ΚΥΡΙΕΣ ΛΕΙΤΟΥΡΓΙΕΣ ΣΕ ΜΙΑ ΟΥΡΑ

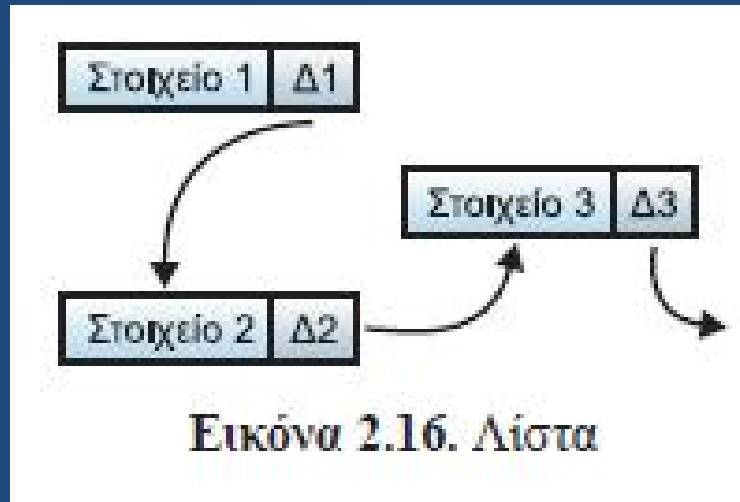
ΕΙΣΑΓΩΓΗ (ENQUEUE) ενός στοιχείου στο πίσω άκρο της ουράς .

ΕΞΑΓΩΓΗ (DEQUEUE) ενός στοιχείου από το εμπρός άκρο της ουράς .

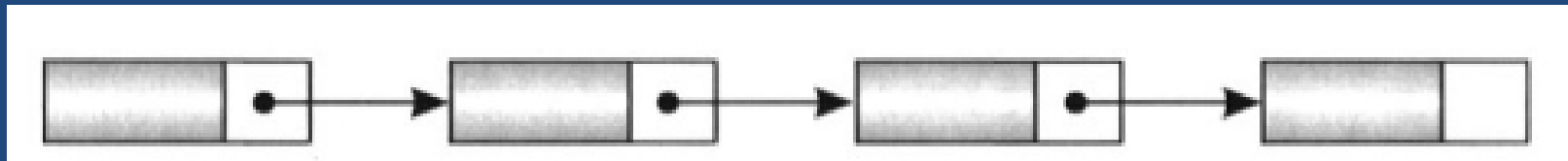
το πρώτο στοιχείο που εισάγεται είναι και το πρώτο στοιχείο που μπορεί να εξαχθεί , **(FIFO) first-in-first-Out.**

2.2.6 Δεδομένα και αναπαράστασή τους

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



Ανεξάρτητα από τη θέση που καταλαμβάνει στη μνήμη ένα δεδομένο, συσχετίζεται με το επόμενο του με τη βοήθεια κάποιου **δείκτη (pointer)**.



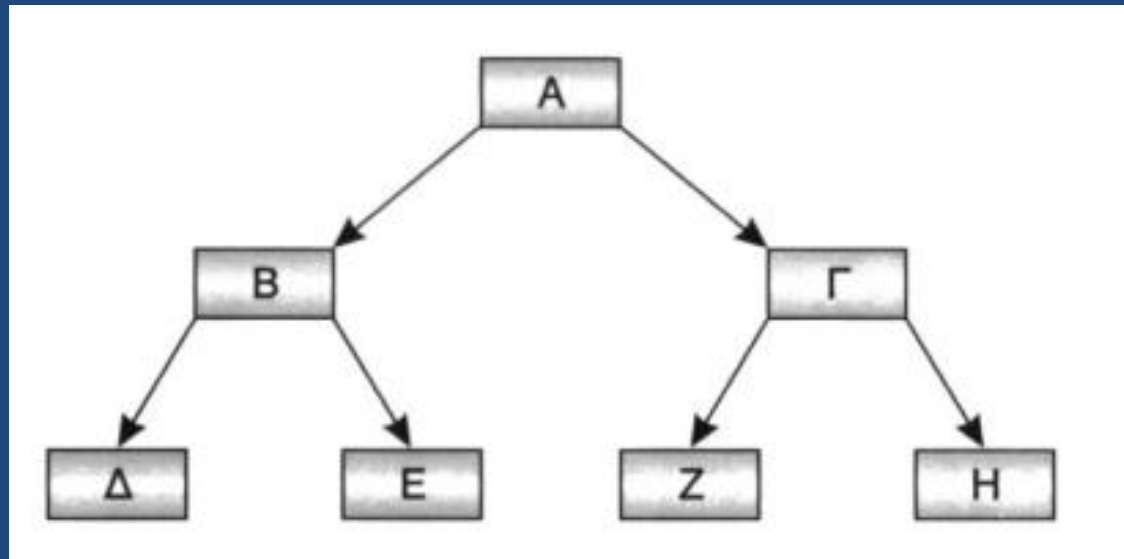
Μία λίστα με τέσσερις κόμβους

2.2.6 Δεδομένα και αναπαράστασή τους

Το **δένδρο (tree)** είναι **μη γραμμική δομή** που αποτελείται από ένα σύνολο κόμβων, οι οποίοι συνδέονται με ακμές .

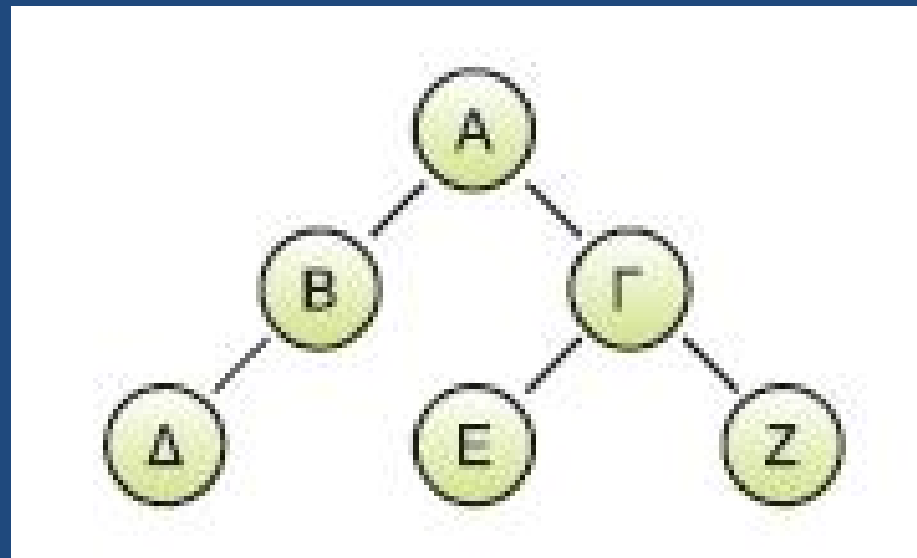
Υπάρχει μόνο ένας κόμβος **(A)** , από τον οποίο μόνο ξεκινούν ακμές, που ονομάζεται **ρίζα (root)**.

Σε όλους τους άλλους κόμβους καταλήγει μία ακμή και ξεκινούν καμία, μία ή περισσότερες. Οι κόμβοι στους οποίους καταλήγουν μόνο ακμές, ονομάζονται **φύλλα**.



2.2.6 Δεδομένα και αναπαράστασή τους

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



2.2.6 Δεδομένα και αναπαράστασή τους

Οι δομές δεδομένων χωρίζονται σε **στατικές και δυναμικές**

ΣΤΑΤΙΚΕΣ ΔΟΜΕΣ	ΔΥΝΑΜΙΚΕΣ ΔΟΜΕΣ
Έχουν σταθερό μέγεθος	δεν έχουν σταθερό μέγεθος
Κρατούν συγκεκριμένο πλήθος στοιχείων	Το πλήθος των στοιχείων τους μεγαλώνει η μικραίνει

2.2.6 Δεδομένα και αναπαράστασή τους

Οι δομές δεδομένων χωρίζονται σε **γραμμικές και μη γραμμικές**

Στις **γραμμικές δομές** δεδομένων τα δεδομένα είναι γραμμικά διατεταγμένα, δηλαδή κάποιο στοιχείο είναι **πρώτο** και κάποιο **τελευταίο**, ενώ για οποιοδήποτε υπάρχει ένα **προηγούμενο** και ένα **επόμενο** στοιχείο.

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**.

2.2.6 Δεδομένα και αναπαράστασή τους

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



Εικόνα 2.19. Δομή αρχείου