



# Περιεχόμενα

- ❑ **Δεδομένα**
  - **Μελέτη δεδομένων**
  - **Τύποι δεδομένων**
  - **Δομές δεδομένων**
- ❑ **Αλγόριθμοι**
  - **Μελέτη αλγορίθμων**
  - **Ανάλυση αλγορίθμων**
  - **Πολυπλοκότητα αλγορίθμων**

## Δεδομένα / Μελέτη δεδομένων

- ❖ Τα **δεδομένα** (data) είναι η αφαιρετική αναπαράσταση της πραγματικότητας και συνεπώς μία απλοποιημένη όψη της.
- ❖ Η συλλογή των ακατέργαστων δεδομένων και ο συσχετισμός τους δίνει ως αποτέλεσμα την **πληροφορία**.
- ❖ Η μέτρηση, η κωδικοποίηση και η μετάδοση της πληροφορίας αποτελεί αντικείμενο του κλάδου της **Θεωρία Πληροφοριών** (Information Theory), ένα ιδιαίτερο γνωστικό πεδίο της Πληροφορικής.

## Δεδομένα / Μελέτη δεδομένων

Η πληροφορική μελετά τα δεδομένα από τις ακόλουθες οπτικές:

1. Υλικού.
2. Γλωσσών προγραμματισμού.
3. Δομές Δεδομένων.
4. Ανάλυσης Δεδομένων.

## Δεδομένα / Τύποι δεδομένων

Υπάρχουν οι εξής τύποι δεδομένων:

- 1. Τύπος δεδομένων υλικού** (hardware data type) και αναφέρετε στον τύπο δεδομένων που μπορεί να υποστηρίξει το υλικό, π.χ. οι προσημασμένοι αριθμοί, κτλ.
- 2. Οι ιδεατοί (εικονικοί) τύποι δεδομένων** (virtual data type) αναφέρονται στο είδος των δεδομένων που οι μεταβλητές μίας γλώσσας προγραμματισμού υψηλού επιπέδου μπορούν να αποθηκεύσουν και να επεξεργαστούν. Π.χ. real, integer, κτλ.
- 3. Οι αφηρημένοι τύποι δεδομένων** (abstract data type), όπου βοηθούν στην τμηματοποίηση του προγράμματος και την συνεργασία πολλών προγραμματιστών.

## Δεδομένα / Δομές δεδομένων

**Δομή δεδομένων** είναι ένα σύνολο αποθηκευμένων δεδομένων που υφίσταται επεξεργασία από ένα σύνολο λειτουργιών, που καλούνται από το υπόλοιπο πρόγραμμα.

Διακρίνεται από τα εξής στοιχεία:

1. **Δομή αποθήκευσης** - δηλ. μία αναπαράσταση της δομής στο χαμηλότερο φυσικό επίπεδο
2. **Σύνολο ορισμών συναρτήσεων** – όπου κάθε συνάρτηση εκτελεί μία λειτουργία στο περιεχόμενο της δομής
3. **Ένα σύνολο αλγορίθμων** – έναν αλγόριθμο για κάθε συνάρτηση όπου κάθε αλγόριθμος περιγράφει πώς θα γίνει η επεξεργασία της δομής αποθήκευσης.

## Δεδομένα / Δομές δεδομένων

Κάθε μορφή δομής δεδομένων αποτελείται από ένα σύνολο κόμβων (nodes).

Οι βασικές **λειτουργίες** επί των δομών δεδομένων είναι οι ακόλουθες:

1. **Προσπέλαση** (access) – Συνήθως αναζητείται ένας κόμβος, ώστε να εξετασθεί ή να τροποποιηθεί το περιεχόμενό του.
2. **Εισαγωγή** (insertion) – Η προσθήκη νέων κόμβων σε υπάρχουσα δομή.
3. **Διαγραφή** (deletion) – Το αντίστροφο της εισαγωγής
4. **Αναζήτηση** (search) – Η προσπέλαση ενός ή περισσότερων κόμβων, που έχουν ως χαρακτηριστικό την ύπαρξη της ίδιας τιμής σε κάποιο ή κάποια πεδία του κόμβου. Το πεδίο του κόμβου που ανιχνεύεται ονομάζεται κλειδί (key).

## Δεδομένα / Δομές δεδομένων

- 5. Ταξινόμηση (sorting)** – Σε πολλές εφαρμογές οι κόμβοι πρέπει να ταξινομηθούν σύμφωνα με αύξουσα ή φθίνουσα σειρά ενός ή περισσότερων κλειδιών.
- 6. Αντιγραφή (copying)** – Όλοι οι κόμβοι ή μερικοί από τους κόμβους μίας δομής αντιγράφονται σε μία άλλη δομή.
- 7. Συγχώνευση (merging)** – Δύο ή περισσότερες δομές συνενώνονται σε μία ενιαία δομή.
- 8. Διαχωρισμός (separation)** – Το αντίστροφο της συγχώνευσης.



## Δεδομένα / Δομές δεδομένων

Τα δεδομένα ενός κόμβου μπορεί να αντιστοιχούν σε μία ή περισσότερες μεταβλητές που αποθηκεύονται σε μία ή περισσότερες **λέξεις** (words), της κύριας μνήμης. Κάθε λέξη μπορεί να είναι από 8 ως 64 δυαδικά ψηφία. Μία υποδιαίρεση του κόμβου είναι το **πεδίο** (field). Τα πεδία μπορεί να είναι **δεδομένα** (data) ή **δείκτες** (pointer) ή **δεσμοί** (link). Ο δείκτης δείχνει την διεύθυνση κάποιου κόμβου, είτε σε απόλυτη μορφή (absolute address) είτε σχετική (relative address).

## Αλγόριθμοι / Μελέτη αλγορίθμων

- ❖ Η έννοια του αλγορίθμου είναι θεμελιώδης για την επιστήμη της πληροφορικής και η λέξη προέρχεται από έναν Πέρση μαθηματικό που έζησε το 825 π.χ..
- ❖ Κάποιοι αλγόριθμοι είναι χιλιάδες χρόνια παλιοί, όμως στα πλαίσια του μαθήματος θα επικεντρωθούμε σε αλγορίθμους των τελευταίων 3-4 δεκαετιών.

# Αλγόριθμοι / Μελέτη αλγορίθμων

## Αλγόριθμος – Ορισμός

Αλγόριθμος είναι ένα πεπερασμένο σύνολο εντολών αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, οι οποίες αν ακολουθηθούν επιτυγχάνεται ένα επιθυμητό αποτέλεσμα.

## Αλγόριθμοι / Μελέτη αλγορίθμων

Απαραίτητα χαρακτηριστικά ενός αλγορίθμου:

❖ **Είσοδος (input)** – Καμία, μία ή περισσότερες ποσότητες να δίνονται ως είσοδοι στον αλγόριθμο.

❖ **Έξοδος (output)** – Ο αλγόριθμος να δημιουργεί τουλάχιστον μία ποσότητα ως αποτέλεσμα.

❖ **Καθορισμός (definiteness)** – Κάθε εντολή να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της.

## Αλγόριθμοι / Μελέτη αλγορίθμων

❖ **Περατότητα (finiteness)** – Ο αλγόριθμος να τελειώνει μετά από πεπερασμένα βήματα εκτέλεσης των εντολών του. Μία διαδικασία που δεν τελειώνει μετά από ένα πεπερασμένο αριθμό βημάτων λέγεται απλώς υπολογιστική διαδικασία (computational procedure).

❖ **Αποτελεσματικότητα (effectiveness)** – Κάθε μεμονωμένη εντολή του αλγορίθμου να είναι απλή. Αυτό σημαίνει ότι μία εντολή δεν αρκεί να έχει ορισθεί, αλλά πρέπει να είναι και εκτελέσιμη.

## Αλγόριθμοι / Μελέτη αλγορίθμων

Η πληροφορική μελετά τους αλγορίθμους από τις ακόλουθες οπτικές:

1. Υλικού
2. Γλωσσών Προγραμματισμού
3. Θεωρητική
4. Αναλυτική

Η αναπαράσταση των αλγορίθμων γίνεται με:

1. Φυσική γλώσσα (natural language)
2. Διαγράμματα Ροής (flow chart)
3. Κωδικοποίηση (coding)

Υπάρχει μία εξάρτηση (Wirth 1976):

**Αλγόριθμοι + Δομές Δεδομένων = Προγράμματα**

## Αλγόριθμοι / Πολυπλοκότητα

Ο απλούστερος τρόπος μέτρησης της απόδοσης ενός αλγορίθμου είναι ο **εμπειρικός** (empirical) ή εκ των υστέρων (posteriori). Ο αλγόριθμος υλοποιείται και εφαρμόζεται σε ένα σύνολο δεδομένων και υπολογίζεται ο χρόνος επεξεργασίας (processing time) και η χωρητικότητα μνήμης (memory space).

### Μειονεκτήματα

- Δεν μπορεί να προβλεφθεί η συμπεριφορά του αλγορίθμου για κάποιο άλλο σύνολο δεδομένων.
- Ο χρόνος επεξεργασίας εξαρτάται από το υλικό, τη γλώσσα προγραμματισμού και το μεταφραστή, και προπάντων από τη δεινότητα του προγραμματιστή.

## Αλγόριθμοι / Πολυπλοκότητα

❖ Ο δεύτερος τρόπος εκτίμησης της απόδοσης είναι ο **θεωρητικός** (theoretical) ή αλλιώς εκ των προτέρων (a priori).

❖ Για το σκοπό αυτό εισάγεται μία μεταβλητή  $n$  η οποία εκφράζει το μέγεθος του προβλήματος, ώστε η μέτρηση της αποδοτικότητας του αλγορίθμου να ισχύει για οποιοδήποτε σύνολο δεδομένων και ανεξάρτητα από υποκείμενους παράγοντες.

❖ Ο χρόνος επεξεργασίας και ο απαιτούμενος χώρος εκτιμώνται από μία συνάρτηση  $f(n)$ , που εκφράζει τη χρονική πολυπλοκότητα ή την πολυπλοκότητα χώρου. Σε πολλές περιπτώσεις όμως δεν μας ενδιαφέρουν οι ακριβείς τιμές αλλά η γενική συμπεριφορά του αλγορίθμου, δηλ. η τάξη του αλγορίθμου. Για αυτό τον σκοπό εισάγουμε τον συμβολισμό  $O$ .



## Αλγόριθμοι / Πολυπλοκότητα

### Ορισμός O

Αν η πολυπλοκότητα ενός αλγορίθμου είναι  $f(n)$ , τότε λέγεται ότι είναι τάξης  $O(g(n))$  αν υπάρχουν δύο θετικοί ακέραιοι  $c$  και  $n_0$ , έτσι ώστε για κάθε  $n \geq n_0$  να ισχύει:

$$|f(n)| \leq c |g(n)|$$

Ο συμβολισμός  $O$  δίνει ένα **άνω** φράγμα για την πολυπλοκότητα του αλγορίθμου.

## Αλγόριθμοι / Πολυπλοκότητα

### Ορισμός Ω

Αν η πολυπλοκότητα ενός αλγορίθμου είναι  $f(n)$ , τότε λέγεται ότι είναι τάξης  $\Omega(g(n))$  αν υπάρχουν δύο θετικοί ακέραιοι  $c$  και  $n_0$ , έτσι ώστε για κάθε  $n \geq n_0$  να ισχύει:

$$|f(n)| \geq c|g(n)|$$

Ο συμβολισμός  $\Omega$  δίνει ένα **κάτω** φράγμα για την πολυπλοκότητα του αλγορίθμου.

## Αλγόριθμοι / Πολυπλοκότητα

### Ορισμός Θ

Αν η πολυπλοκότητα ενός αλγορίθμου είναι  $f(n)$ , τότε λέγεται ότι είναι τάξης  $\Theta(g(n))$  αν υπάρχουν δύο ακέραιοι  $c_1, c_2$  και  $n_0$ , έτσι ώστε για κάθε  $n \geq n_0$  να ισχύει:

$$c_1 |g(n)| \leq |f(n)| \leq c_2 |g(n)|$$

## Αλγόριθμοι / Πολυπλοκότητα

### Χρήσιμοι κανόνες – O

1. Αν η  $f(x)$  είναι άθροισμα όρων, τότε κρατάμε εκείνον τον όρο που μεγαλώνει πιο γρήγορα.
2. Αν η  $f(x)$  είναι γινόμενο όρων, τότε αφαιρούμε όλες τις σταθερές.

### Παράδειγμα:

$$f(x) = 6x^4 - 2x^3 + 5, \quad \rightarrow 1^{\text{ος}} \text{ κανόνας} \rightarrow 6x^4 \rightarrow 2^{\text{ος}} \text{ κανόνας} \rightarrow x^4, \text{ άρα } g(x) = x^4$$

Με εφαρμογή του ορισμού έχουμε:

$$|6x^4 - 2x^3 + 5| \leq 6x^4 + 2x^3 + 5 \leq 6x^4 + 2x^4 + 5x^4 = 13x^4 = 13|x^4|$$

για  $n_0=1$ , και  $c=13$ .

## Αλγόριθμοι / Πολυπλοκότητα

Άσκηση 1 : Να υπολογιστεί η τάξη πολυπλοκότητας (αναφορικά με την πράξη του πολλαπλασιασμού) των παρακάτω αλγορίθμων.

$$Q_p(r) = \sum_{k=0}^p (-1)^{p+k} \frac{(p+k+1)!}{(p-k)!k!(k+1)!} r^k$$

και

$$Q_p(r) = (-1)^p \cdot T_{p0} + \sum_{k=1}^p (-1)^{p+k} \cdot T_{pk} \cdot r^k$$

$$T_{p0} = p+1$$

$$T_{pk} = \frac{(p+k+1)(p-k+1)}{k(k+1)} \cdot T_{p(k-1)}$$

Ποιος από του παραπάνω αλγορίθμους είναι πιο γρήγορος ;