Εισαγωγή στον Προγραμματισμό

Εισαγωγική ενότητα (25%) : Βασικές έννοιες και τομείς της Επιστήμης Υπολογιστών 2η ομιλία

Παναγιώτης Τζουνάκης

Φθινόπωρο 2022

Τμήμα Μαθηματικών
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Προγραμματισμός και Μαθηματικά

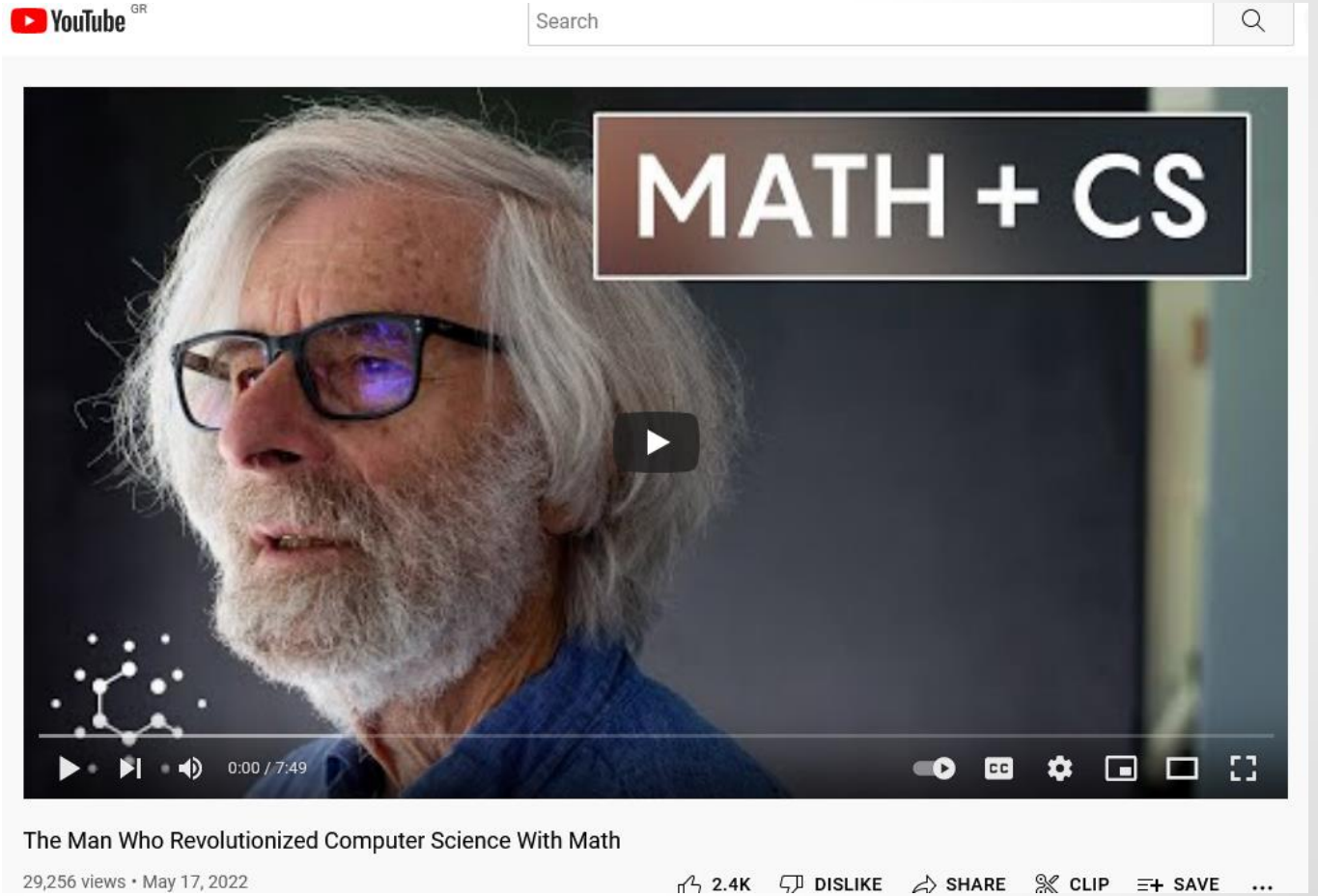
Προγραμματισμός ~~και~~ με Μαθηματικά

Ο Προγραμματισμός ~~και~~ είναι Μαθηματικά

11ε

Leslie Lamport :

- LaTeX
- Paxos, a “consensus algorithm”
- the bakery algorithm
- the Byzantine Generals Problem
- “formal verification,” with a “specification language” called TLA+ (for Temporal Logic of Actions)



https://www.quantamagazine.org/computing-expert-says-programmers-need-more-math-20220517

https://www.youtube.com/watch?v=rkZzg7Vowao

Ανάλυση Προβλημάτων

- **πρόβλημα** το [próvlima] Ο49 **:** **1.** (σύνθετο, πολύπλοκο) ερώτημα, ζήτημα, στο οποίο επιζητείται και επιχειρείται να δοθεί απάντηση με επιστημονικό τρόπο, με επιστημονική μέθοδο: *Φιλοσοφικά / ηθικά / ιστορικά / μεταφυσικά προβλήματα. Οι αρχαίοι φιλόσοφοι έθεσαν το ~ της δομής της ύλης.* || *Θεωρητικό ~,* που ανάγεται στη θεωρία, στην αφηρημένη σκέψη. **2.** (ειδικότ.) ερώτημα, ζήτημα που ορισμένα στοιχεία του είναι γνωστά, και με βάση αυτά ζητείται η εύρεση άλλων, άγνωστων με μαθηματικές ή με άλλες επιστημονικές μεθόδους: *Μαθηματικό / αλγεβρικό / γεωμετρικό ~. Διατυπώνω ένα ~. Το ~ επιδέχεται δύο διαφορετικές λύσεις. Τα δεδομένα / τα ζητούμενα ενός προβλήματος. Το ~ του τετραγωνισμού του κύκλου είναι άλυτο. Στις εξετάσεις έπεσε / μπήκε ένα πολύ δύσκολο ~.* (έκφρ.) *δήλιο* ~.* **3.** δύσκολη, περίπλοκη κατάσταση, υπόθεση, θέμα που πρέπει να αντιμετωπιστεί, που επιζητεί λύση, διευθέτηση: *Θέτω / επισημαίνω / θίγω / προσεγγίζω / συζητώ / αντιμετωπίζω / λύνω / επιλύω / οξύνω / περιπλέκω ένα ~. Ένα ~ αναδύεται / εμφανίζεται / παρουσιάζεται / προκύπτει / δημιουργείται. Γλωσσικό / δημογραφικό / κυκλοφοριακό ~. Ανθρώπινα / τεχνικά / οικονομικά / επαγγελματικά / προσωπικά προβλήματα. Το ~ του αλκοολισμού / των ναρκωτικών / του ρατσισμού / της στέγης. Η ανεργία είναι κοινωνικό ~. Απαιτείται ομοψυχία για την αντιμετώπιση των εθνικών προβλημάτων. Η αύξηση της χρήσης βίας συνιστά ένα σοβαρό ~. Τεχνητό ~. Καταβλήθηκαν προσπάθειες να παρακαμφθεί το ~. Οι εργαζόμενοι ζήτησαν από την κυβέρνηση την επίλυση των προβλημάτων τους.* (έκφρ.) *πρόβλημά σου,* για κτ. που θεωρούμε ότι αφορά μόνο το συνομιλητή μας και όχι εμάς: *Έχω μείνει χωρίς φράγκο. - Πρόβλημά σου.* **4.** δυσκολία, δυσχέρεια που δημιουργεί αρνητικές καταστάσεις, δυσλειτουργίες: *Έχει ~ συνεννόησης με τους συναδέλφους του. Στις μεγάλες πόλεις υπάρχει ~ επικοινωνίας μεταξύ των ανθρώπων. Έχει ~ να διαφωνήσει με ανωτέρους του. Το αυτοκίνητο παρουσιάζει προβλήματα στη μηχανή / στην τροφοδοσία / στα ηλεκτρολογικά. Έχει προβλήματα με τους γείτονές του. Υπάρχει ~ στο παρκάρισμα. Άτομα με προβλήματα ακοής / ομιλίας / κίνησης. Αρκετά προβλήματα έχω, μη μου δημιουργείς κι άλλα!* || *Ψυχολογικά προβλήματα,* δυσκολίες στην εύρεση ψυχικής ισορροπίας: *Η εφηβική / γεροντική ηλικία παρουσιάζει πολλά ψυχολογικά προβλήματα.* || (έκφρ.) *κανένα ~!:* α. δεν υπάρχει δυσκολία, όλα είναι εύκολα, λειτουργούν καλά. β. δεν έχω αντίρρηση: *Θα περάσεις να με πάρεις; - Κανένα ~, θα έρθω κατά τις οχτώ.* **προβληματάκι** το ΥΠΟΚΟΡ μικρό, εύκολο, ασήμαντο πρόβλημα.
- [λόγ.: 1, 2: αρχ. *πρόβλημα*· 3, 4: σημδ. γαλλ. problème & αγγλ. problem (στις νέες σημ.) < αρχ. *πρόβλημα*]

Πηγή : Λεξικό της κοινής νεοελληνικής (http://www.greek-language.gr/greekLang/modern_greek/tools/lexica/triantafyllides/)

Ανάλυση Προβλημάτων

Συνοπτικός ορισμός προβλήματος :

Με τον όρο **Πρόβλημα** προσδιορίζεται ένα ερώτημα (μια κατάσταση) που χρήζει (διανοητικής ικανότητας-σκέψης) (αντιμετώπισης), απαιτεί απάντηση (λύση), η δε απάντηση (λύση) της δεν είναι γνωστή, ούτε προφανής.

Δεδομένα (γνωστά) →

→ Ζητούμενα (άγνωστα)

Κατάσταση (κακή) →

οδικός χάρτης ;;;;

→ κατάσταση (καλή)

Κατηγορίες Προβλημάτων

Ως προς δυνατότητα επίλυσης
Επιλύσιμα
Ανοικτά
Μη επιλύσιμα

Ως προς δόμηση λύσεων
Δομημένα
Ημιδομημένα
Αδόμητα

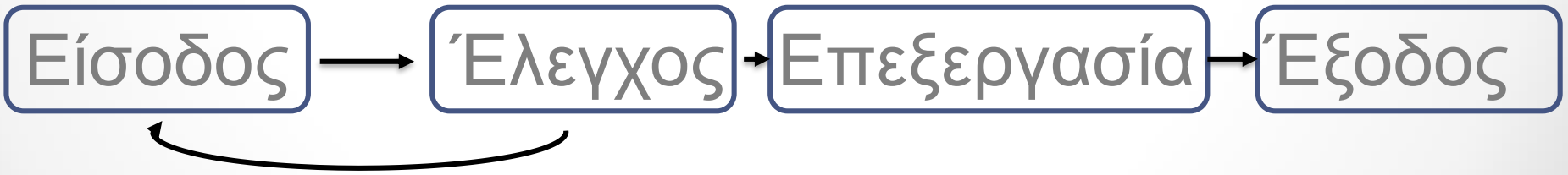
Ως προς είδος ζητούμενης επίλυσης
Απόφασης (ΝΑΙ/ΟΧΙ)
Υπολογιστικά (τιμή ικανοποιητικής απάντησης)
Βελτιστοποίησης (min/max result)

Διαδικασία επίλυσης προβλημάτων

- Κατανόηση (σαφήνεια, πληρότητα, συνέπεια έκφρασης)
- Απαιτήσεις : καθορισμός δεδομένων και ζητουμένων
- Ανάλυση - αφαίρεση
- Σύνθεση
- Κατηγοριοποίηση
- Γενίκευση

Διαδικασία επίλυσης προβλημάτων

- Καταχώριση δεδομένων
- Έλεγχος δεδομένων
- Επεξεργασία δεδομένων
- Εξαγωγή αποτελεσμάτων



Νοοτροπία επίλυσης προβλημάτων

- **CLARITY** : Εκφράζω με **σαφήνεια** το πρόβλημα και την ζητούμενη λύση
- **RELEVANCE** : Ξεδιαλέγω τα **σχετικά – συναφή** όντα και ασχολούμαι μόνο με αυτά
- **RESULT** : Εστιάζω στο **αποτέλεσμα** (από το οποίο θα κριθώ στο τέλος…)

Επίλυση προβλημάτων με Η/Υ

Όταν:

- Υπάρχει μεγάλη πολυπλοκότητα των υπολογισμών (μπερδεύει τον άνθρωπο)
- Υπάρχει επαναληπτικότητα των διαδικασιών (ο άνθρωπος βαριέται, κουράζεται και κάνει λάθη)
- Απαιτείται ταχύτητα εκτέλεσης των πράξεων (βιαζόμαστε – εφαρμογές πραγματικού χρόνου)
- υπάρχει μεγάλο πλήθος των δεδομένων (big data(sets) – κλίμακα που υπερβαίνει τις ανθρώπινες δυνατότητες)

Πως λύνει προβλήματα ο Η/Υ

Βασικές λειτουργίες (κλασσικού) υπολογιστή :

- Πρόσθεση
- Σύγκριση
- Μεταφορά

Οι παραπάνω δράσεις αποτελούν τους δομικούς λίθους με τους οποίους διεκπεραιώνεται το σύνολο των (σύνθετων) εργασιών του υπολογιστή.

Αλγόριθμοι

Ορισμοί:

- Λεξικό της κοινής νεοελληνικής:

  αλγόριθμος ο [alγόriθmos] Ο20α : (μαθημ.) σύνολο κανόνων που εφαρμόζονται για την επίλυση ενός προβλήματος.

  κανόνας 1 ο [kanónas] Ο2 : Ι1. ό,τι ρυθμίζει υποχρεωτικά σχέσεις ή τρόπους ενέργειας. …
- Βιβλία Λυκείου: «Αλγόριθμος είναι μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος.»
- https://en.wikipedia.org/wiki/Algorithm : In mathematics and computer science, an algorithm … is a **finite** sequence of **well-defined**, computer-implementable **instructions**, typically to solve a class of problems or to perform a computation … Algorithms are always unambiguous and are used as specifications for performing calculations, data processing, automated reasoning, and other tasks.

Αλγόριθμοι
χαρακτηριστικά (κριτήρια)

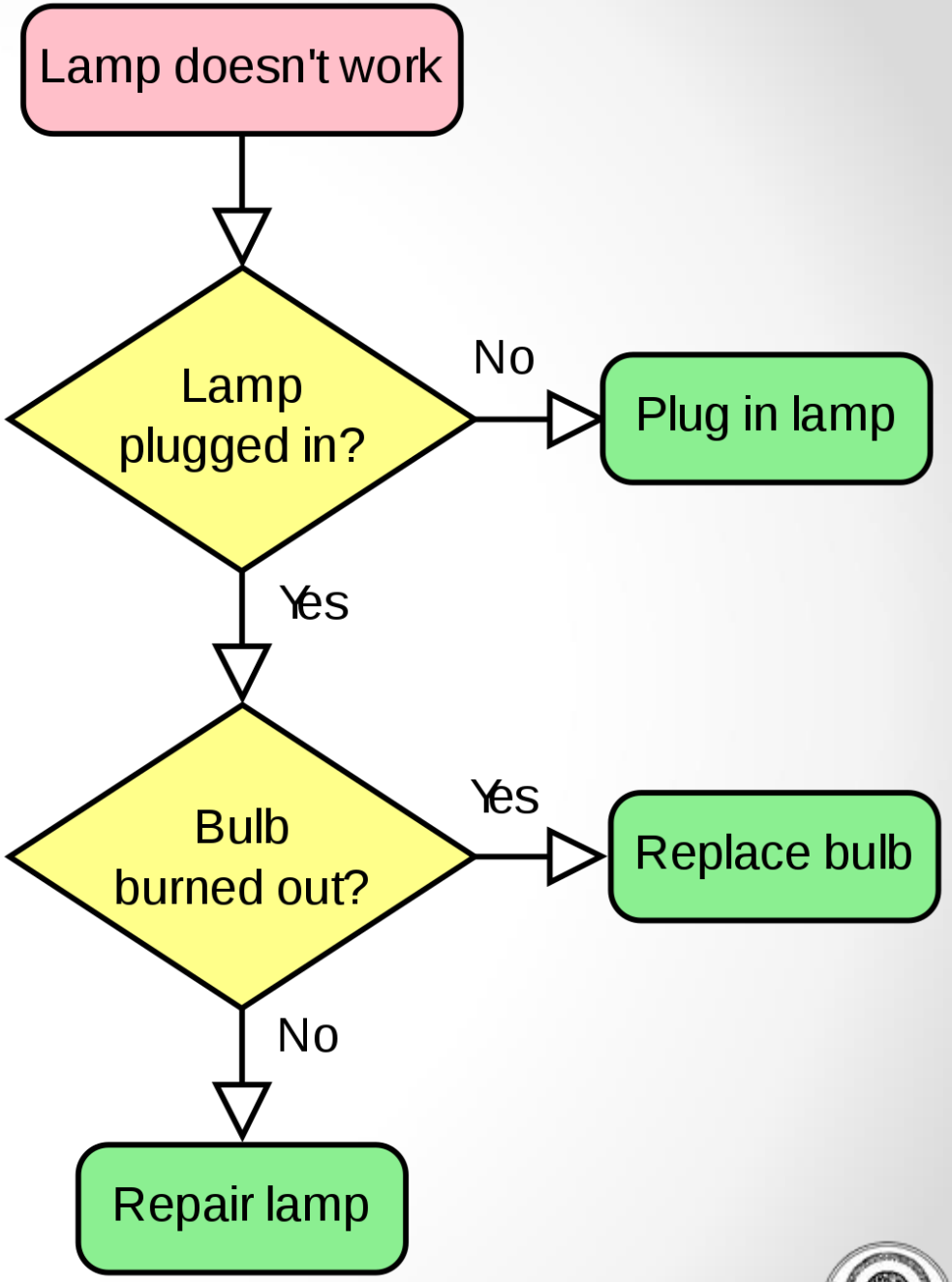
- Είσοδος (προαιρετικά) 0 ή περισσότερων τιμών
- Έξοδος (υποχρεωτικά) τουλάχιστον μιας τιμής
- Σαφώς καθορισμένος τρόπος εκτέλεσης όλων των εντολών – βημάτων
- Αποτελεσματικότητα : για την εκτέλεση κάθε (βήματος) εντολής απαιτείται πεπερασμένος χρόνος και πόροι
- Πεπερασμένος αριθμός βημάτων μέχρι την παραγωγή εξόδου

Αλγόριθμοι

Προσεγγίσεις μελέτης

- Υλικό/αρχιτεκτονική : ταχύτητα εκτέλεσης
- Γλώσσα προγραμματισμού : επίπεδο αφαίρεσης, φιλοσοφία-οικογένεια (https://en.wikipedia.org/wiki/List_of_programming_languages_by_type , https://en.wikipedia.org/wiki/Category:Programming_languages)
- Θεωρητική : εύρεση αλγορίθμου για επίλυση προβλήματος
- Αναλυτική : υπολογιστικοί πόροι

Αλγόριθμοι (διάγραμμα ροής)

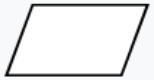


https://en.wikipedia.org/wiki/Flowchart#/media/File:LampFlowchart.svg

Αλγόριθμοι (διάγραμμα ροής)

Common symbols [edit]

The American National Standards Institute (ANSI) set standards for flowcharts and their symbols in the 1960s.[14] The International Organization for Standardization (ISO) adopted the ANSI symbols in 1970.[15] The current standard, ISO 5807, was revised in 1985.[16] Generally, flowcharts flow from top to bottom and left to right.[17]

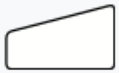
ANSI/ISO Shape	Name	Description
	Flowline (Arrowhead)[15]	Shows the process's order of operation. A line coming from one symbol and pointing at another.[14] Arrowheads are added if the flow is not the standard top-to-bottom, left-to right.[15]
	Terminal[14]	Indicates the beginning and ending of a program or sub-process. Represented as a stadium,[14] oval or rounded (fillet) rectangle. They usually contain the word "Start" or "End", or another phrase signaling the start or end of a process, such as "submit inquiry" or "receive product".
	Process[15]	Represents a set of operations that changes value, form, or location of data. Represented as a rectangle.[15]
	Decision[15]	Shows a conditional operation that determines which one of the two paths the program will take.[14] The operation is commonly a yes/no question or true/false test. Represented as a diamond (rhombus).[15]
	Input/Output[15]	Indicates the process of inputting and outputting data,[15] as in entering data or displaying results. Represented as a rhomboid.[14]
	Annotation[14] (Comment)[15]	Indicating additional information about a step in the program. Represented as an open rectangle with a dashed or solid line connecting it to the corresponding symbol in the flowchart.[15]
	Predefined Process[14]	Shows named process which is defined elsewhere. Represented as a rectangle with double-struck vertical edges.[14]
	On-page Connector[14]	Pairs of labeled connectors replace long or confusing lines on a flowchart page. Represented by a small circle with a letter inside.[14][18]
	Off-page Connector[14]	A labeled connector for use when the target is on another page. Represented as a home plate-shaped pentagon.[14][18]

https://en.wikipedia.org/wiki/Flowchart

Αλγόριθμοι (διάγραμμα ροής)

Other symbols [edit]

The ANSI/ISO standards include symbols beyond the basic shapes. Some are:[17][18]

Shape	Name	Description
	Data File or Database	Data represented by a cylinder (disk drive).
	Document	Single documents represented a rectangle with a wavy base.
		Multiple documents represented stacked rectangle with a wavy base.
	Manual operation	Represented by a trapezoid with the longest parallel side at the top, to represent an operation or adjustment to process that can only be made manually.
	Manual input	Represented by quadrilateral, with the top irregularly sloping up from left to right, like the side view of a keyboard.
	Preparation or Initialization	Represented by an elongated hexagon, originally used for steps like setting a switch or initializing a routine.

https://en.wikipedia.org/wiki/Flowchart

Αλγόριθμοι

Άλλοι τρόποι περιγραφής αλγορίθμων:

Φυσική γλώσσα : Διαβάστε δυο αριθμούς στις μεταβλητές a και b.
Αντιγράψτε την τιμή της μεταβλητής a σε μια νέα μεταβλητή x.
Αντιγράψτε την τιμή της μεταβλητής b στη θέση της μεταβλητής a.
Αντιγράψτε την τιμή της μεταβλητής x στη θέση της μεταβλητής b.
Τυπώστε το περιεχόμενο των μεταβλητών a και b.

Ψευδογλώσσα :

```
read a,b
x <- a
a <- b
b <- x
write a, b
```

Γλώσσα προγραμματισμού: (αναλυτική συζήτηση προσεχώς!)

Αλγόριθμοι (δομή)

Γλωσσικά αντικείμενα (δεδομένα, πράξεις, εντολές):

Σταθερές

- Αριθμητικές, π.χ. 3,14
- Αλφαριθμητικές, π.χ. «Μήτσος»
- Λογικές, π.χ. «αληθής», true

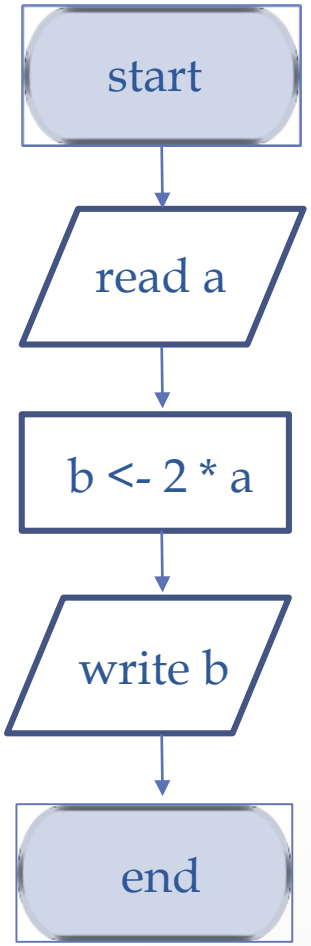
- Μεταβλητές π.χ. x, y, z, n_1
- Τελεστές π.χ. '+', '-', (λογικό)AND
- Εκφράσεις π.χ. (2+α)*4, n^2
- Εντολή εκχώρησης (assignment)

  μεταβλητή <- έκφραση

Αλγόριθμοι (συνιστώσες)

Ακολουθία εντολών
(σειριακά βήματα)

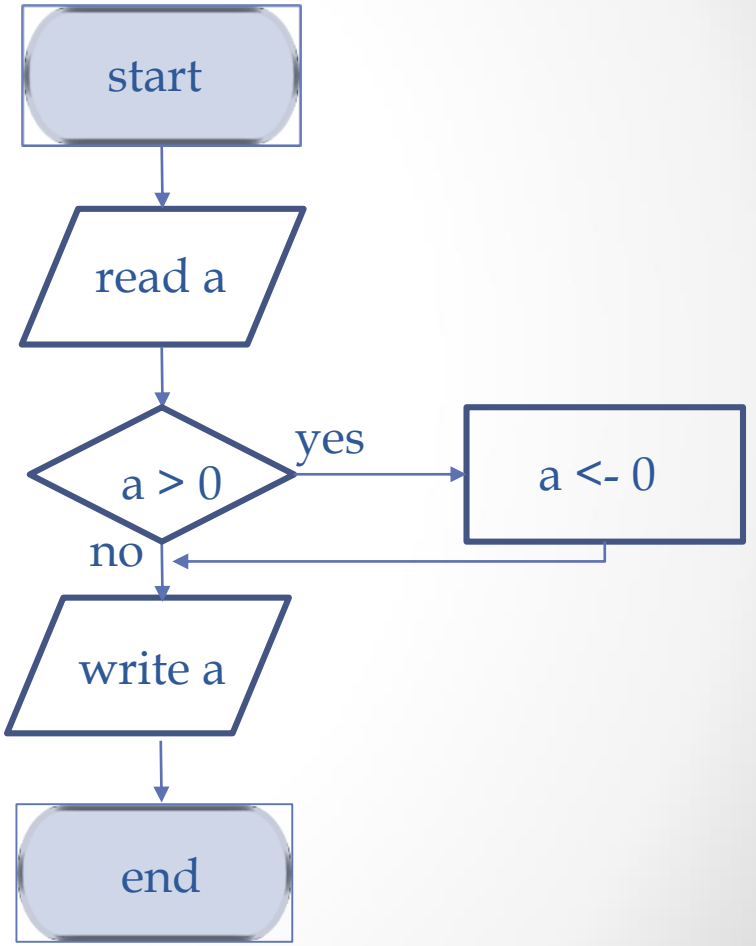
```
start
read a
b <- 2 * a
print b
end
```



Αλγόριθμοι (συνιστώσες)

Ακολουθία εντολών
(επιλογή)

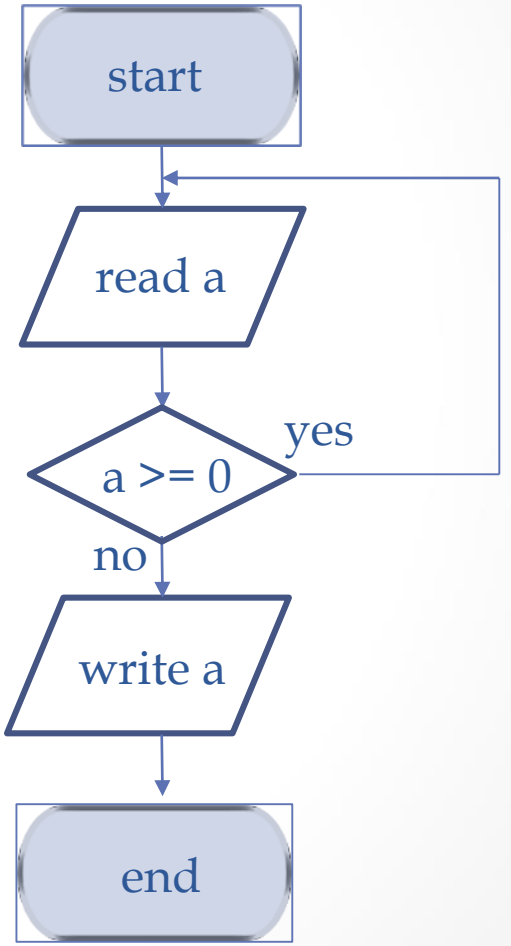
```
start
read a
if a > 0 then a <- 0
write a
end
```



Αλγόριθμοι (συνιστώσες)

Ακολουθία εντολών
(επανάληψη)

```
start
repeat
read a
until a < 0
write a
end
```



Αλγόριθμοι (σχεδίαση)

- Αναγκαιότητα ανάλυσης των προβλημάτων και σχεδίασης των κατάλληλων αλγορίθμων
- Ακολουθία βημάτων για την ανάλυση των αλγορίθμων
- Σύγχρονες τεχνικές σχεδίασης αλγορίθμων
- Κυριότερες προσεγγίσεις επίλυσης και ανάλυσης προβλημάτων

Αλγόριθμοι (ανάλυση)

- Καταγραφή των διαθέσιμων πληροφοριών για το πρόβλημα (αρχική κατάσταση, δεδομένα, εκτίμηση μεγέθους προβλήματος)
- Αντίληψη τυχόν ιδιαιτεροτήτων (προαπαιτούμενες συνθήκες, παράπλευρες επιπτώσεις/παρενέργειες)
- Πρόταση μεθόδου επίλυση και μορφοποίηση της (π.χ. ψευδογλώσσα, διάγραμμα ροής, κ.α.)
- Επιλογή γλώσσας προγραμματισμού και υπολογιστικής πλατφόρμας, υλοποίηση και αποτελέσματα

Αλγόριθμοι (μέθοδοι σχεδίασης)

Χαρακτηριστικά μεθόδων

- Τρόπος αντιμετώπισης/χειρισμού δεδομένων
- Ακολουθία εντολών
- αποδοτικότητα

Ενδεικτικές μέθοδοι

- Διαίρει και βασίλευε
- Δυναμικός προγραμματισμός
- Άπληστοι (Greedy) αλγόριθμοι

Αλγόριθμοι (Διαίρει και βασίλευε)

- Ζητείται να λυθεί συγκεκριμένη περίπτωση (στιγμιότυπο) ενός προβλήματος.
- Η περίπτωση αυτή του προβλήματος μπορεί να διαιρεθεί σε υπο-περιπτώσεις με τα ίδια χαρακτηριστικά.
- Λύνουμε ανεξάρτητα κάθε μια υποπερίπτωση.
- Συνδυάζουμε όλες τις επιμέρους λύσεις που κατασκευάσαμε για τις υπο-περιπτώσεις, συνθέτοντας έτσι τη συνολική λύση του προβλήματος.

Παράδειγμα: Δυαδική αναζήτηση σε ηλεκτρονικό τηλεφωνικό κατάλογο (σημ: οι εγγραφές του τηλ. Καταλόγου είναι πάντα ταξινομημένες αλφαβητικά).

Αλγόριθμοι (Δυναμικός προγραμματισμός)

Ας ακολουθήσουμε τώρα την αντίστροφη προσέγγιση:

- Λύνουμε πρώτα την μικρότερη/ειδικότερη υποπερίπτωση του προβλήματος.
- Σταδιακά υπολογίζουμε τις λύσεις όλο και μεγαλύτερων/γενικότερων υποπεριπτώσεων του προβλήματος.
- Τελικά καταλήγουμε στη σύνθεση της συνολικής λύσης.

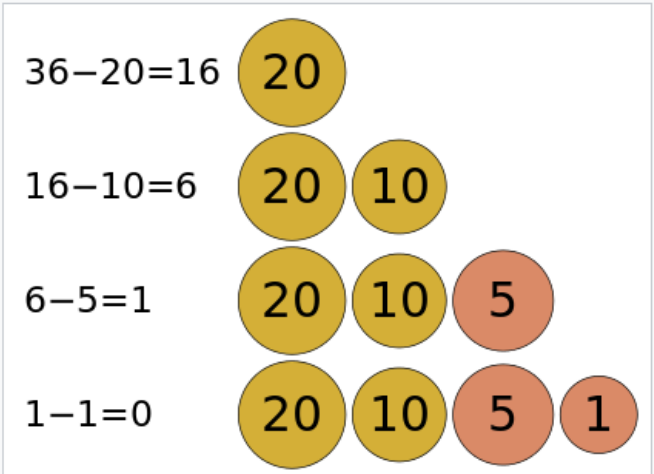
Παράδειγμα: Υπολογισμός δύναμης π.χ. 2^7 με τέσσερις μόνο πολλαπλασιασμούς (2^4*2^2*2) αντί για έξι ($2*2*2*2*2*2*2$)

Αλγόριθμοι (άπληστοι αλγόριθμοι)
ή και μυωπικοί αλγόριθμοι

Επιλέγει τοπικά βέλτιστη λύση σε κάθε βήμα (σημ: Δεν βρίσκει πάντοτε την βέλτιστη λύση).
Χαρακτηριστικά:

- Ένα σύνολο με υποψήφιες λύσεις
- Λειτουργία επιλογής του καλύτερου υποψηφίου
- Λειτουργία ελέγχου σκοπιμότητας για απόφαση επιλεξιμότητας υποψηφίου
- Λειτουργία αποτίμησης που αποδίδει μια τιμή σε μια λύση ή μερική λύση
- Λειτουργία επίλυσης που μας δείχνει αν έχουμε φτάσει σε μια συνολική λύση

βλ. και https://en.wikipedia.org/wiki/Greedy_algorithm



Greedy algorithms determine minimum number of coins to give while making change. These are the steps a human would take to emulate a greedy algorithm to represent 36 cents using only coins with values {1, 5, 10, 20}. The coin of the highest value, less than the remaining change owed, is the local optimum. (In general the change-making problem requires dynamic programming to find an optimal solution; however, most currency systems, including the Euro and US Dollar, are special cases where the greedy strategy does find an optimal solution.)

Αλγόριθμοι (επίδοση)

Αξιολόγηση ως προς α) χρόνο εκτέλεσης β) χρησιμοποιούμενη μνήμη

- Χειρότερη περίπτωση
- Μέγεθος εισόδου

Εξαρτώνται από είδος Η/Υ, γλώσσα προγραμματισμού, δομή προγράμματος και δεδομένων. Ο χρόνος επίσης από καθυστέρηση αποθηκευτικού συστήματος και συστήματος εισόδου-εξόδου, άλλους χρήστες/προγράμματα που μοιράζονται τον Η/Υ.

Δύο αλγόριθμοι είναι συγκρίσιμοι όταν:

- Γράφτηκαν στην ίδια γλώσσα προγραμματισμού
- Μεταφράστηκαν από τον ίδιο μεταγλωττιστή (με τις ίδιες παραμέτρους)
- Τρέχουν σε μηχανήματα με ίδιο υλικό, λειτουργικό σύστημα και υπολογιστικό περιβάλλον
- Δέχονται ακριβώς το ίδιο σύνολο δεδομένων ως είσοδο κατά τη σύγκριση

Αλγόριθμοι (ορθότητα)

Λογικά λάθη

Πλημμελής έλεγχος εισόδου

Σημ: συντακτικά λάθη διορθώνονται προκειμένου ο compiler να μπορέσει να μεταφράσει το πρόγραμμα που υλοποιεί τον αλγόριθμο σε εκτελέσιμη μορφή.

Απόδειξη της ορθότητας προϋποθέτει ότι:

- ο αλγόριθμός τερματίζει (**μη προφανές**!)
- κάθε εκτέλεση που τερματίζει δίνει αποδεκτά αποτελέσματα.

Η ίδια η διαδικασία (specification & verification) ενός αλγορίθμου είναι πολύπλοκη και κοστίζει πολύ σε υπολογιστικούς πόρους.

Αλγόριθμοι (πολυπλοκότητα)

Αφορά

1. στον απαιτούμενο χρόνος επεξεργασίας (processing time)
2. στην αντίστοιχη χωρητικότητα μνήμης (memory space)

ΟΡΙΣΜΟΣ : Η πολυπλοκότητα f(n) ενός αλγορίθμου είναι τάξης O(g(n)), αν υπάρχουν δύο θετικοί ακέραιοι c και n0, έτσι ώστε για κάθε $n \geq n0$ να ισχύει:

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

Αλγόριθμοι (πολυπλοκότητα)

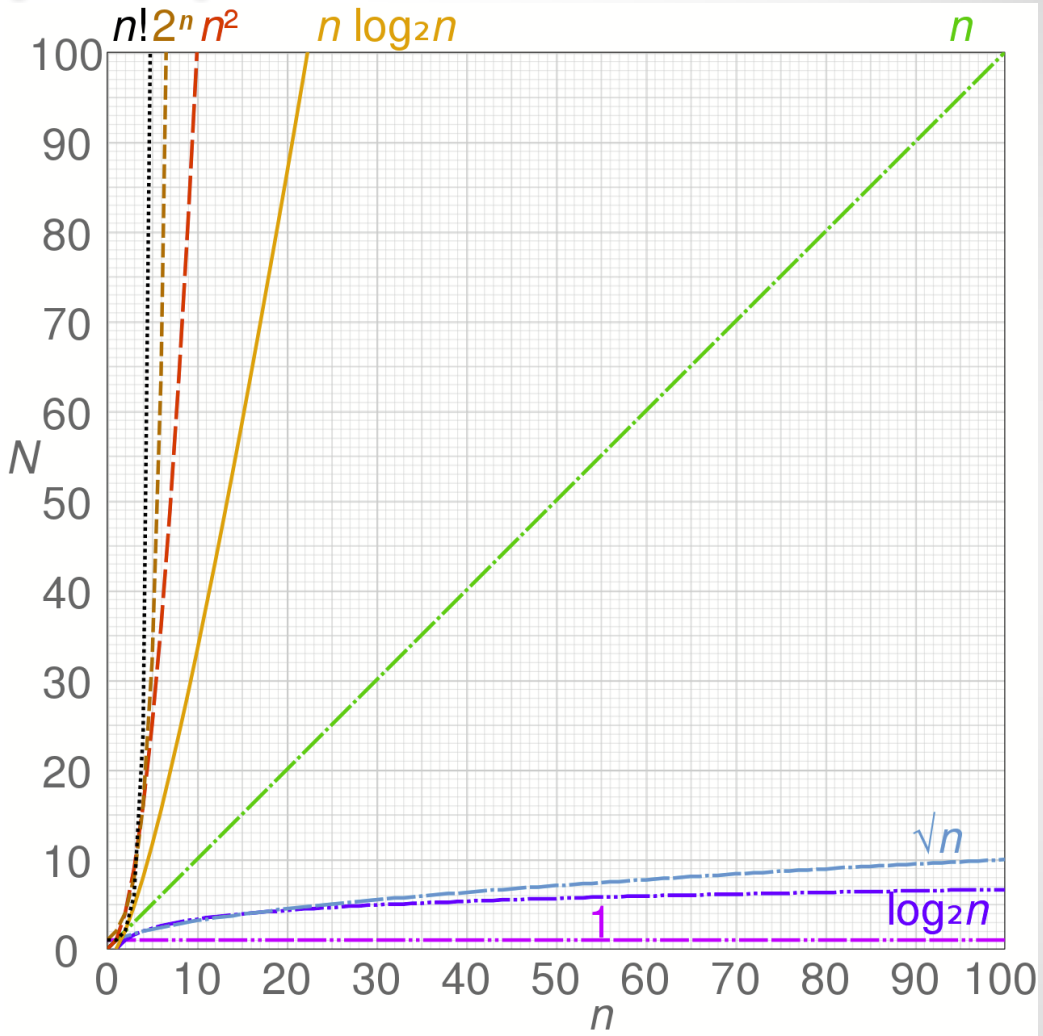
https://en.wikipedia.org/wiki/Time_complexity#Table_of_common_time_complexities

The following table summarizes some classes of commonly encountered time complexities. In the table, poly(x) = $x^{O(1)}$, i.e., polynomial in x.

Name	Complexity class	Running time ($T(n)$)	Examples of running times	Example algorithms
constant time		$O(1)$	10	Finding the median value in a sorted array of numbers Calculating $(-1)^n$
inverse Ackermann time		$O(\alpha(n))$		Amortized time per operation using a disjoint set
iterated logarithmic time		$O(\log^* n)$		Distributed coloring of cycles
log-logarithmic		$O(\log \log n)$		Amortized time per operation using a bounded priority queue[2]
logarithmic time	DLOGTIME	$O(\log n)$	$\log n$, $\log(n^2)$	Binary search
polylogarithmic time		poly(log n)	$(\log n)^2$	
fractional power		$O(n^c)$ where $0 < c < 1$	$n^{1/2}$, $n^{2/3}$	Searching in a kd-tree
linear time		$O(n)$	n, $2n + 5$	Finding the smallest or largest item in an unsorted array, Kadane's algorithm
"n log-star n" time		$O(n \log^* n)$		Seidel's polygon triangulation algorithm.
linearithmic time		$O(n \log n)$	$n \log n$, $\log n!$	Fastest possible comparison sort; Fast Fourier transform.
quasilinear time		n poly(log n)		
quadratic time		$O(n^2)$	n^2	Bubble sort; Insertion sort; Direct convolution
cubic time		$O(n^3)$	n^3	Naive multiplication of two $n \times n$ matrices. Calculating partial correlation.
polynomial time	P	$2^{O(\log n)}$ = poly(n)	$n^2 + n$, n^{10}	Karmarkar's algorithm for linear programming; AKS primality test[3][4]
quasi-polynomial time	QP	$2^{\text{poly}(\log n)}$	$n^{\log \log n}$, $n^{\log n}$	Best-known O($\log^2 n$)-approximation algorithm for the directed Steiner tree problem.
sub-exponential time (first definition)	SUBEXP	$O(2^{n^\varepsilon})$ for all $\varepsilon > 0$	$O(2^{\log n^{\log \log n}})$	Contains BPP unless EXPTIME (see below) equals MA.[5]
sub-exponential time (second definition)		$2^{o(n)}$	$2^{n^{1/3}}$	Best-known algorithm for integer factorization; formerly-best algorithm for graph isomorphism
exponential time (with linear exponent)	E	$2^{O(n)}$	1.1^n, 10^n	Solving the traveling salesman problem using dynamic programming
exponential time	EXPTIME	$2^{\text{poly}(n)}$	2^n, 2^{n^2}	Solving matrix chain multiplication via brute-force search
factorial time		$O(n!)$	$n!$	Solving the traveling salesman problem via brute-force search
double exponential time	2-EXPTIME	$2^{2^{\text{poly}(n)}}$	2^{2^n}	Deciding the truth of a given statement in Presburger arithmetic

Αλγόριθμοι

https://en.wikipedia.org/wiki/Time_complexity#/media/File:Comparison_computational_complexity.svg



Graphs of functions commonly used in the analysis of algorithms, showing the number of operations N versus input size n for each function

Δεδομένα

Προσεγγίσεις μελέτης

- Υλικό/αρχιτεκτονική : μορφές/αναπαραστάσεις (κωδικοποίηση)
- Γλώσσα προγραμματισμού : τύποι δεδομένων (ακέραιοι, δεκαδικοί, χαρακτήρες, κ.ο.κ.)
- Δομές δεδομένων : σύνθετοι τύποι δεδομένων (οι οποίοι χτίζονται από απλούς τύπους) και οι αντίστοιχες (σύνθετες) λειτουργίες οι οποίες χτίζονται από τις βασικές πράξεις και εντολές της γλώσσας (π.χ. εγγραφή (record) με επιμέρους πεδία (fields) – μιγαδικός αριθμός με 2 πεδία (δεκαδικοί)
- Ανάλυση δεδομένων : μέθοδοι τήρησης και συσχέτισης δεδομένων (π.χ. Βάσεις Δεδομένων/Γνώσης)

Δεδομένα : κωδικοποίηση

Κώδικας ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

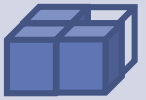
Δεδομένα

Λειτουργίες σε δομές δεδομένων

- Προσπέλαση (access)
- Εισαγωγή (insertion)
- Διαγραφή (deletion)
- Αναζήτηση (search)
- Ταξινόμηση (sorting)
- Αντιγραφή (copy)
- Συγχώνευση (merge)
- Διαχωρισμός (splitting)

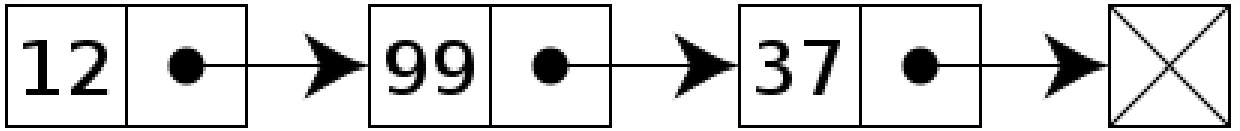
- Εφαρμόζονται συνολικά στη δομή δεδομένων ή κάποιο δομικό στοιχείο (building block/node) της
- Δεν έχουν πάντα νόημα όλες οι λειτουργίες σε όλες τις δ.δ.

Δομές δεδομένων (Πίνακας)

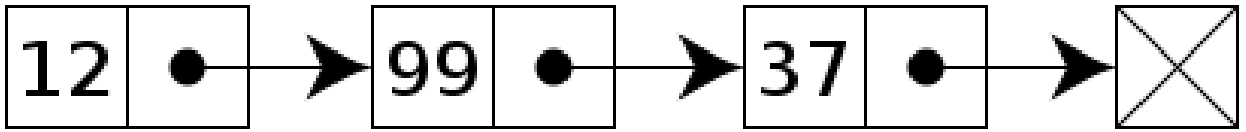
Μαθηματικό αντικείμενο	Δομή δεδομένων	Παράδειγμα
Βαθμωτή ποσότητα (scalar)	Μεταβλητή βασικού τύπου π.χ. Real x = 5,1	
Διάνυσμα σε n-διάστατο πραγματικό χώρο $\mathbb{R}^n$ (1, 5, 3)	Μονοδιάστατος πίνακας array[i] of real ο δείκτης i ∈ [1, 2, .., n] array[1]=1, array[2]=5, array[3]=3	n=3 
Πίνακας (matrix) n × m με στοιχεία πραγματικούς (1 0) (2 3)	Δυσδιάστατος πίνακας array[i][j] of real ο δείκτης i ∈ [1, 2, .., n] ο δείκτης j ∈ [1, 2, .., m] array[1][1]=1, array[1][2]=0, array[2][1]=2, array[2][2]=3	 n = m = 2
Τανυστής (tensor) … (νωρίς για πρωτοετείς…)	Πολυδιάστατος πίνακας array[i_1][i_2]…[i_n] of real ο δείκτης $i_1 \in$ [1, 2, .., k_1] ο δείκτης $i_2 \in$ [1, 2, .., k_2] … ο δείκτης $i_n \in$ [1, 2, .., k_n]	 n = 3, $k_1 = k_2 = k_3 = 2$

Δομές δεδομένων (λίστα)

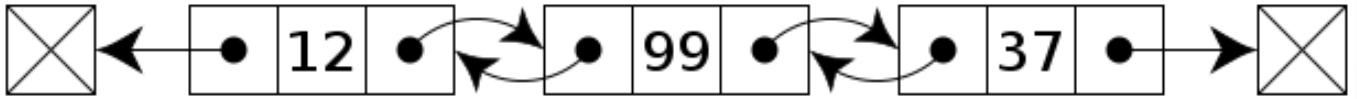
https://en.wikipedia.org/wiki/Linked_list



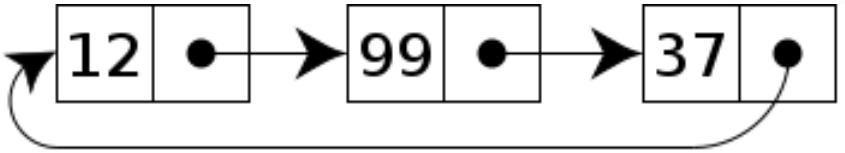
A linked list whose nodes contain two fields: an integer value and a link to the next node. The last node is linked to a terminator used to signify the end of the list.



A singly linked list whose nodes contain two fields: an integer value and a link to the next node

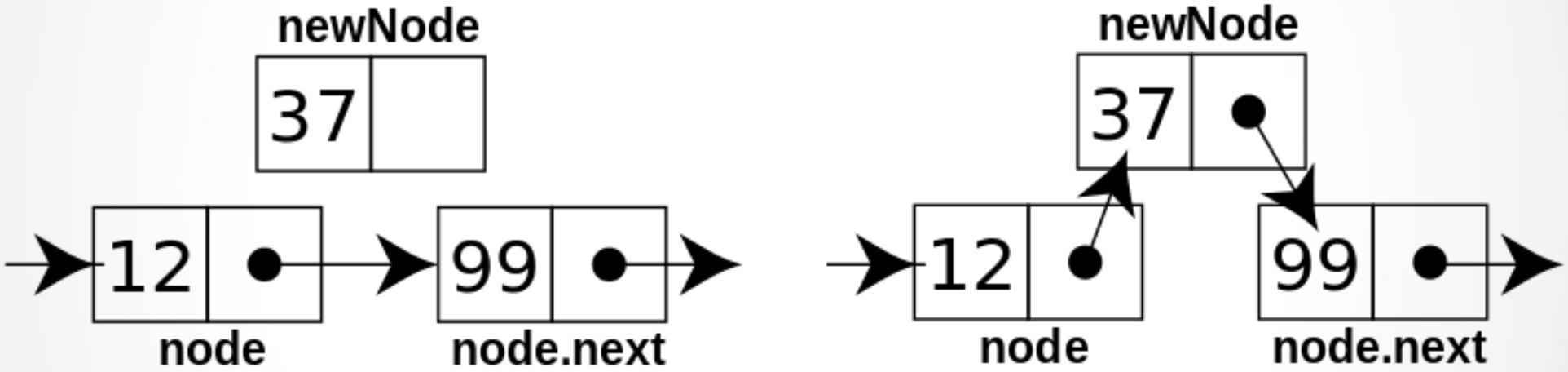


A doubly linked list whose nodes contain three fields: an integer value, the link forward to the next node, and the link backward to the previous node



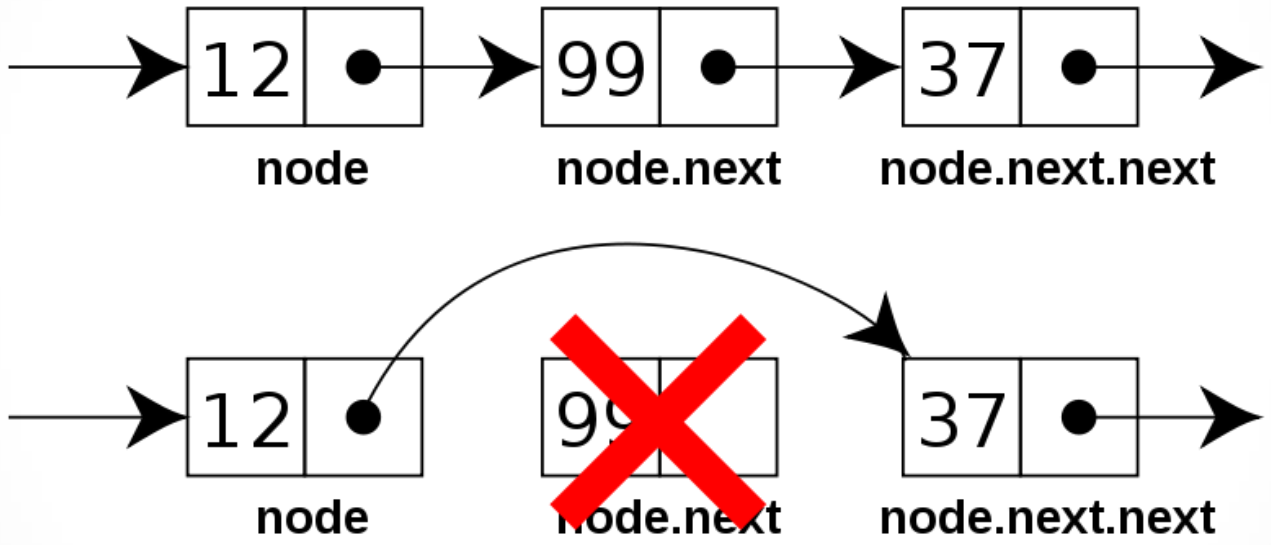
A circular linked list

Δομές δεδομένων (λίστα)



Εισαγωγή στοιχείου σε λίστα

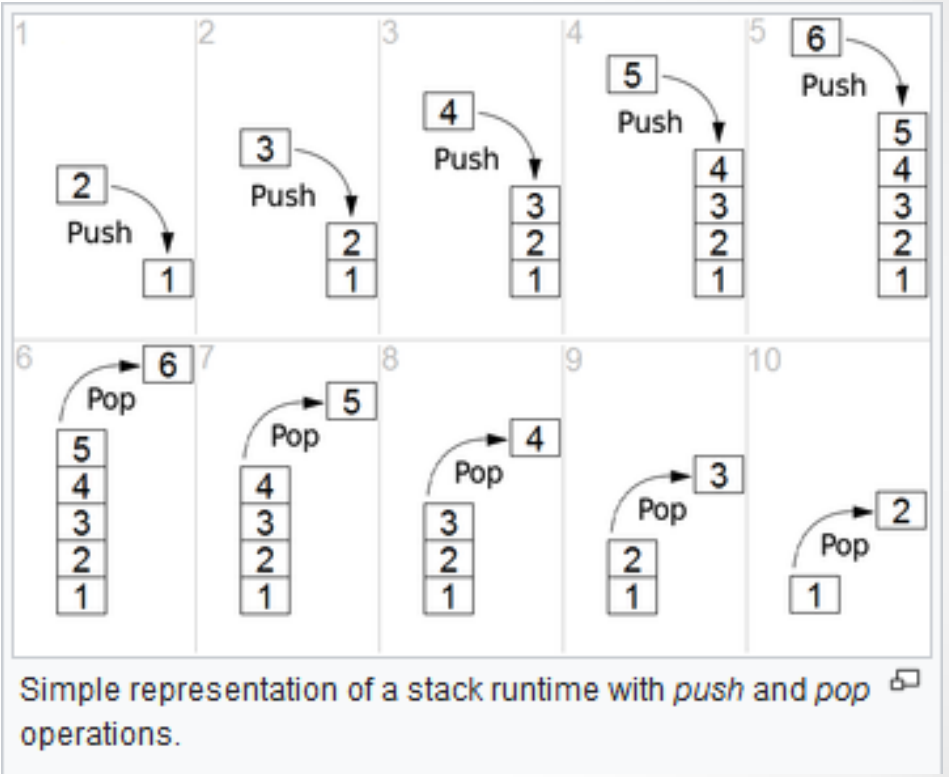
Δομές δεδομένων (λίστα)



Διαγραφή στοιχείου από λίστα

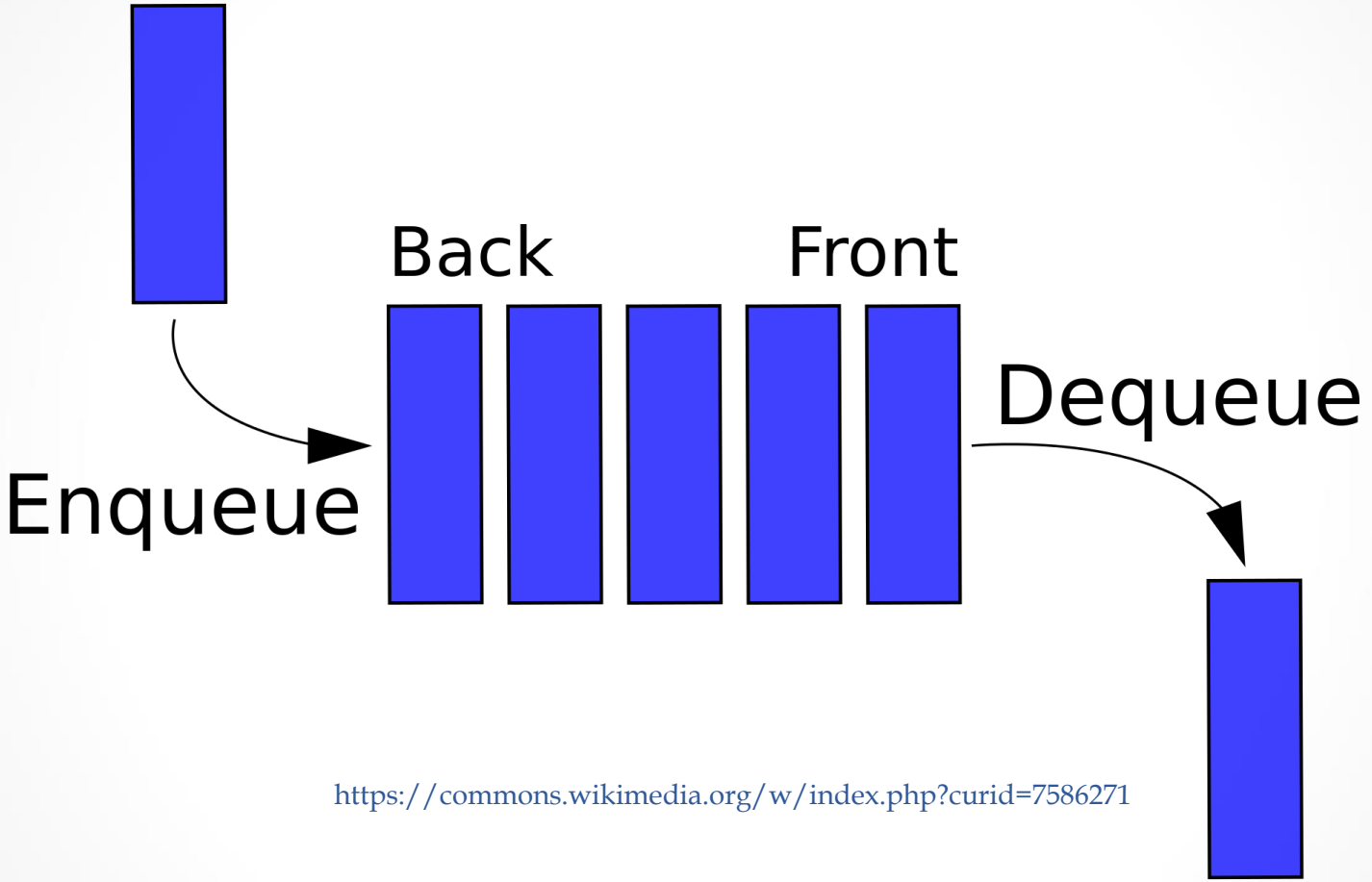
Δομές δεδομένων (στοίβα)

https://en.wikipedia.org/wiki/Stack_(abstract_data_type)



Last In – First Out (LIFO)

Δομές δεδομένων (ουρά)



https://commons.wikimedia.org/w/index.php?curid=7586271

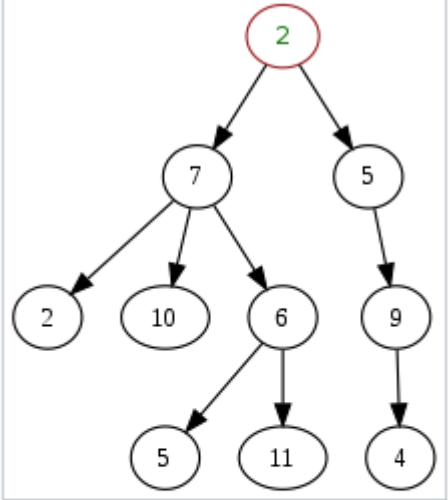
Representation of a FIFO (first in, first out) queue

Δομές δεδομένων (δένδρο)

https://en.wikipedia.org/wiki/Tree_(data_structure)

Συνήθεις πράξεις:

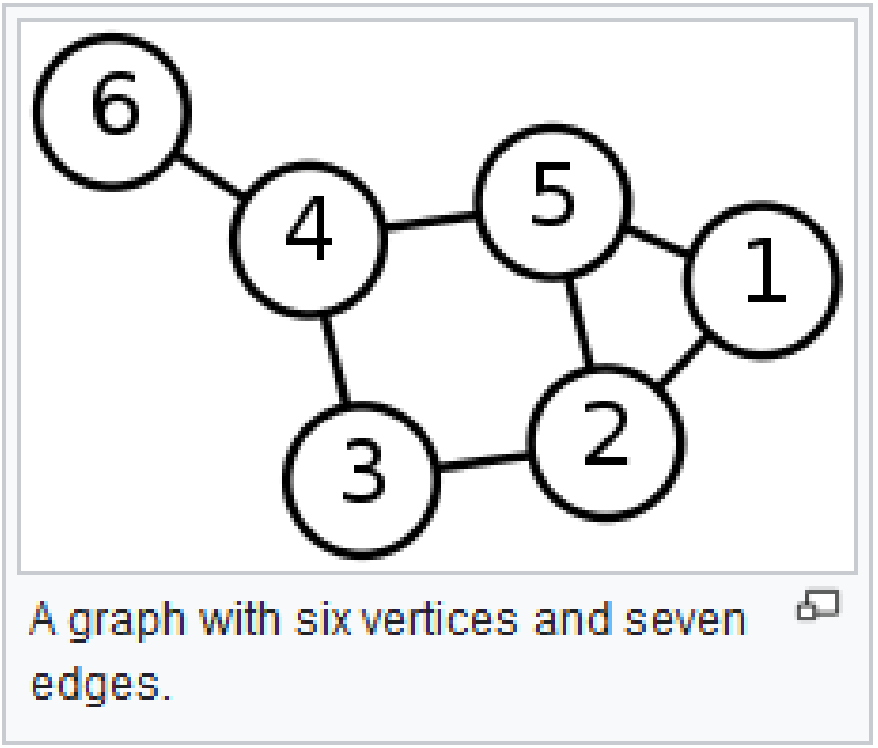
- Απαρίθμηση όλων των κόμβων
- Απαρίθμηση των κόμβων ενός υποδέντρου
- Αναζήτηση στοιχείου
- Εισαγωγή νέου στοιχείου σε συγκεκριμένη θέση στο δέντρο
- Μπόλιασμα: εισαγωγή ενός ολόκληρου υποδένδρου
- Αναζήτηση της ρίζας ενός κόμβου
- Αναζήτηση του κοντινότερου κοινού προγόνου δύο κόμβων



A generic, and so non-binary, unsorted, some labels duplicated, arbitrary diagram of a tree. In this diagram, the node labeled 7 has three children, labeled 2, 10 and 6, and one parent, labeled 2. The root node, at the top, has no parent.

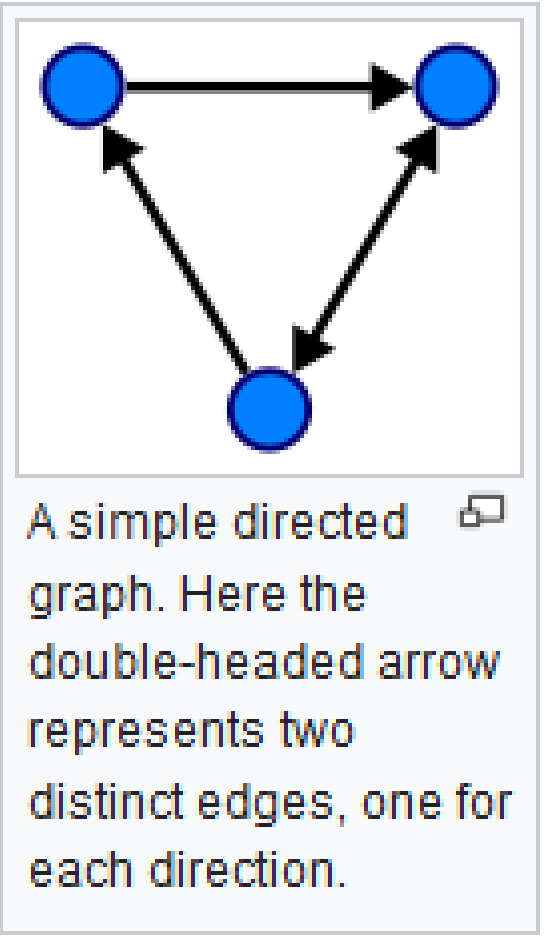
Δομές δεδομένων (γράφος)

https://en.wikipedia.org/wiki/Graph_(abstract_data_type)



A graph with six vertices and seven edges.

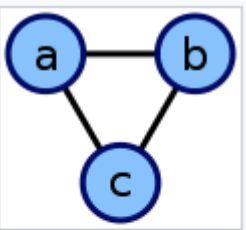
Μη κατευθυνόμενος γράφος



A simple directed graph. Here the double-headed arrow represents two distinct edges, one for each direction.

Δομές δεδομένων (γράφος)

Υλοποίηση με λίστα γειτόνων (adjacency list https://en.wikipedia.org/wiki/Adjacency_list)

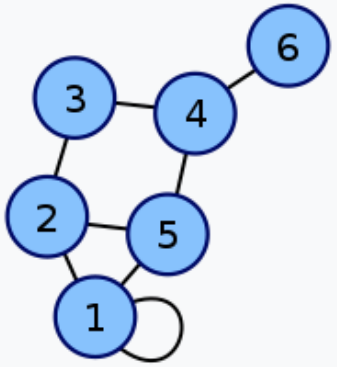
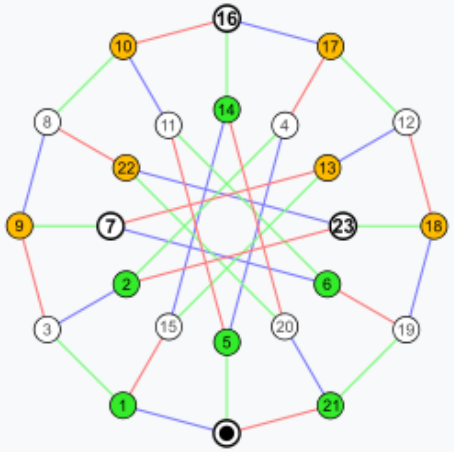
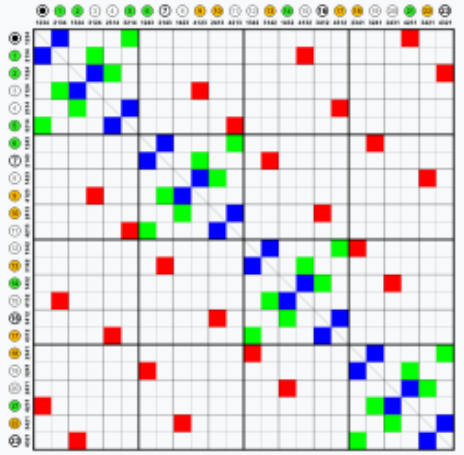


This undirected cyclic graph can be described by the three unordered lists {b, c}, {a, c}, {a, b}.

The graph pictured above has this adjacency list representation:		
a	adjacent to	b,c
b	adjacent to	a,c
c	adjacent to	a,b

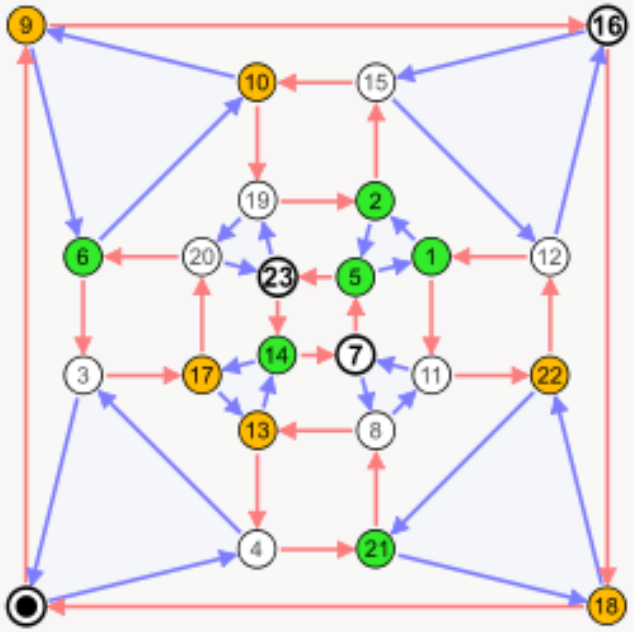
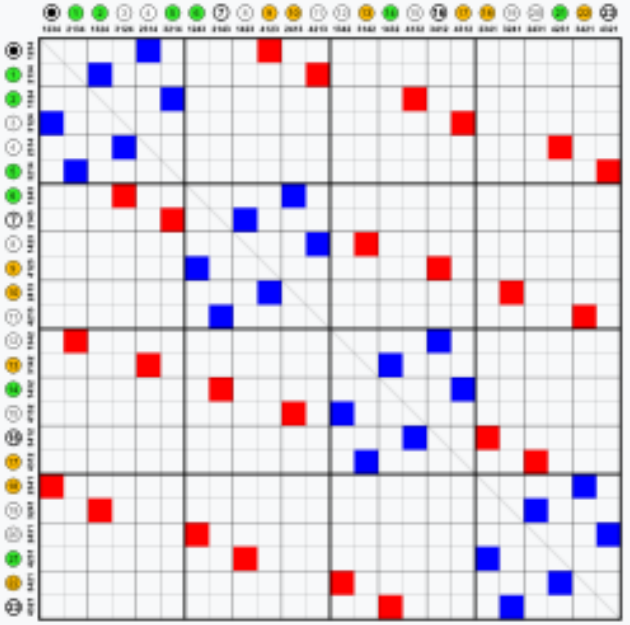
Δομές δεδομένων (γράφος)

Υλοποίηση με πίνακα γειτνίασης (adjacency matrix https://en.wikipedia.org/wiki/Adjacency_matrix)

Labeled graph	Adjacency matrix
	$\begin{pmatrix} 2 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$ Coordinates are 1–6.
 Nauru graph	 Coordinates are 0–23. White fields are zeros, colored fields are ones.

Δομές δεδομένων (γράφος)

πίνακας γειτνίασης για κατευθυνόμενο γράφο

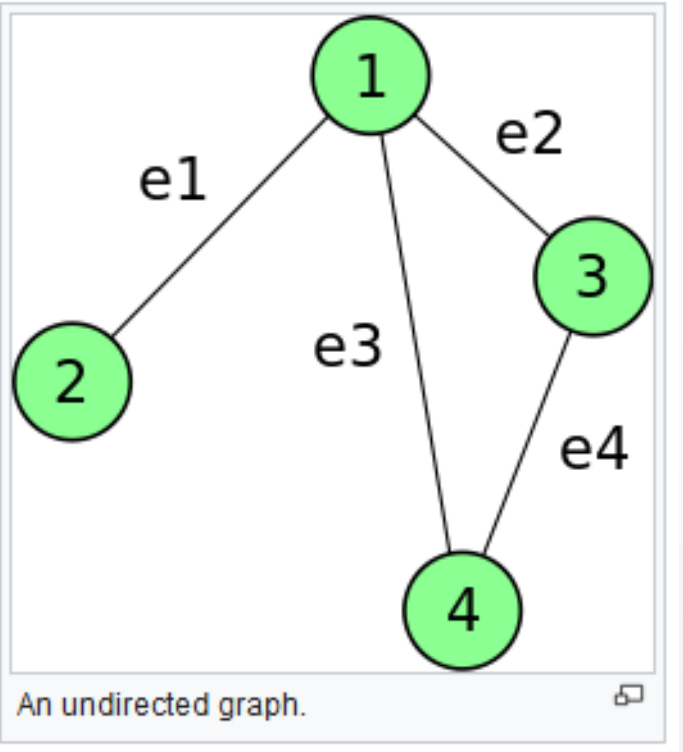
Labeled graph	Adjacency matrix
	
Directed Cayley graph of S_4	Coordinates are 0–23. As the graph is directed, the matrix is not necessarily symmetric.

Δομές δεδομένων (γράφος)

Υλοποίηση με πίνακα προσπτώσεων (Incidence matrix https://en.wikipedia.org/wiki/Incidence_matrix)

	e_1	e_2	e_3	e_4
1	1	1	1	0
2	1	0	0	0
3	0	1	0	1
4	0	0	1	1

$$= \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$



An undirected graph.

Ερωτήσεις & Απαντήσεις