

Αλγόριθμοι

Βασικές έννοιες και αναπαράσταση αλγορίθμων

Διδάσκων: Τσούρος Δήμος

Προηγούμενο μάθημα

Αλγόριθμοι: τρόποι επίλυσης (υπολογιστικών) προβλημάτων

Αλγόριθμοι υπάρχουν παντού

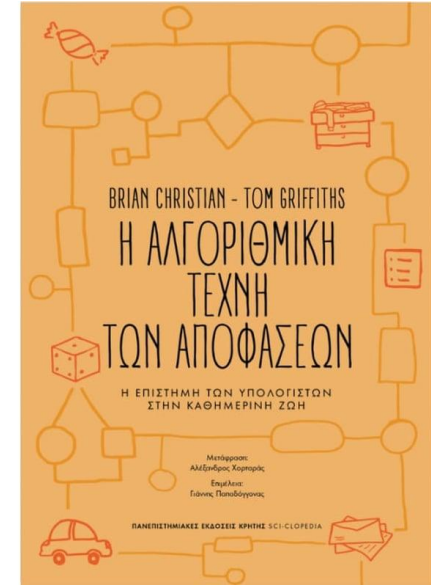
Καθημερινά παραδείγματα:

- χάρτες/πλοήγηση, εύρεση βέλτιστης διαδρομής
- αναζήτηση web,
- προτάσεις ταινιών/βίντεο/...,
- ταξινόμηση email, αναζήτηση

Οργανωτικά παραδείγματα:

- πρόγραμμα εξετάσεων,
- διαχείριση αποθήκης
- Βάρδιες σε νοσοκομεία, εργοστάσια ...

*οι αλγόριθμοι είναι τρόποι σκέψης,
όχι απλά κώδικας*



Αφαίρεση και μοντελοποίηση

- **Μοντελοποίηση προβλήματος**
 - **Πρόβλημα** (ανεπίσημη περιγραφή) → **Προδιαγραφή** (επίσημη περιγραφή)
 - **Αφαίρεση:** Κρατάμε μόνο ό,τι χρειάζεται! Αγνόησε τις μη χρήσιμες λεπτομέρειες
- **Ορίζουμε:**
 - είσοδο
 - έξοδο
 - τι θεωρείται έγκυρη είσοδος, περιορισμοί
 - Σχέση μεταξύ εισόδου και εξόδου (τι επεξεργασία χρειάζεται)
- Παράδειγμα: «Βρες το μέγιστο σε λίστα $A[1..n]$ »
 - είσοδος: $A[1..n]$,
 - έξοδος: μια τιμή,
 - επεξεργασία: η έξοδος είναι το μέγιστο της εισόδου

Η μοντελοποίηση είναι το πρώτο βήμα πριν γράψουμε αλγόριθμο ή πρόγραμμα.

Στιγμιότυπο ή περίπτωση (instance) ΥΠ

Ένα πρόβλημα περιγράφεται γενικά, π.χ.

- το πρόβλημα ανάθεσης καθηγητών σε μαθήματα
- το πρόβλημα ταξινόμησης ενός πίνακα

Στιγμιότυπο (ή περίπτωση) ενός Υπολογιστικού Προβλήματος (ΥΠ) ονομάζεται ο συνδυασμός του προβλήματος με μια συγκεκριμένη είσοδο/αρχικοποίηση των παραμέτρων του, δηλαδή τα δεδομένα που το καθορίζουν.

Για την επίλυση ενός στιγμιότυπου του προβλήματος εφαρμόζεται ένας αλγόριθμος (μέθοδος επίλυσης) που, με βάση τη συγκεκριμένη είσοδο, παράγει μια λύση ή αποφαίνεται ότι δεν υπάρχει λύση.

Ο αλγόριθμος είναι και αυτός **γενικός!** Μπορεί να εφαρμοστεί για την παραγωγή λύσης σε οποιοδήποτε στιγμιότυπο του προβλήματος που καλείται να λύσει

Τι είναι Αλγόριθμος

**Αλγόριθμος: Διαδικασία επίλυσης (υπολογιστικών) προβλημάτων
διατυπωμένη σε μαθηματική μορφή**

Ένας αλγόριθμος έχει συγκεκριμένες ιδιότητες – χαρακτηριστικά. Στη βιβλιογραφία υπάρχουν πολλοί ισοδύναμοι ορισμοί. Τα βασικά χαρακτηριστικά που ξεχωρίζουμε:

- Δεν περιέχει **λογικές αντιφάσεις**
- Λύνει **κάθε** στιγμιότυπο του προβλήματος
- Τελειώνει μετά από **πεπερασμένο** αριθμό πράξεων
- Δεν περιέχει εκφράσεις, οι οποίες ερμηνεύονται με **υποκειμενικό** τρόπο – Όλα τα βήματα είναι σαφή

Αλγόριθμος: Παράδειγμα

Πρόβλημα: Φτιάξε μακαρόνια με κιμά

- Δεν είναι υπολογιστικό πρόβλημα!
- Όμως η μέθοδος επίλυσής του είναι ένας αλγόριθμος.
- Είσοδος: Για πόσους
- Έξοδος: φαΐ

Μέθοδος Εκτέλεσης

ΤΙΡ Καθώς μαγειρεύεις, τσέκαρε τα βήματα που ολοκληρώνεις και ακολούθησε τη συνταγή χωρίς να χαθείς.

Ας μαγειρέψουμε...

- 1 Τοποθετούμε ένα τηγάνι σε δυνατή φωτιά και αφήνουμε να κάψει.
- 2 Ρίχνουμε 1 κ.σ. βούτυρο, σπάμε τον κιμά σε κομμάτια με τα χέρια μας, τον ρίχνουμε και αυτόν και σοτάρουμε για 2-3 λεπτά.
- 3 Γυρνάμε τα κομμάτια του κιμά από την άλλη πλευρά, σοτάρουμε για ακόμα 1 λεπτό και τον μεταφέρουμε σε ένα μπολ.
- 4 Χοντροκόβουμε το κρεμμύδι και ψιλοκόβουμε το σκόρδο.
- 5 Ρίχνουμε στο τηγάνι το υπόλοιπο βούτυρο, το κρεμμύδι και το σκόρδο, και ανακατεύουμε.
- 6 Προσθέτουμε το στικ κανέλας, τα φύλλα δάφνης, τους κόκκους μπαχάρι, τη ζάχαρη, το αλάτι και το πιπέρι, ανακατεύουμε με ένα κουτάλι και σοτάρουμε για 1-2 λεπτά.
- 7 Ρίχνουμε τον κιμά, τον σπάμε με ένα κουτάλι και σοτάρουμε για 1-2 λεπτά.
- 8 Προσθέτουμε τον πελτέ, την ντομάτα κονκασέ και το νερό, χαμηλώνουμε τη φωτιά και σιγοβράζουμε για 30 λεπτά.

Αλγόριθμος και πρόγραμμα: Παράδειγμα

Αλγόριθμος για αναζήτηση σε πίνακα

1. Πάρε τον πίνακα $A[1..n]$ και το στοιχείο-στόχο x .
2. Για κάθε θέση i από 1 μέχρι n :
 1. Σύγκρινε το $A[i]$ με το x .
 2. Αν είναι ίσα:
 1. Επιστροφή i
3. Αν τελειώσει ο πίνακας χωρίς να βρεθεί το στοιχείο:
 1. Επιστροφή -1.

Πρόγραμμα στη γλώσσα προγραμματισμού python

```
def linear_search(A, x):  
    # Για κάθε θέση του πίνακα  
    for i in range(len(A)):  
        if A[i] == x:           # αν βρέθηκε το στοιχείο  
            return i           # επιστρέφουμε τη θέση  
    return -1                   # δεν βρέθηκε
```

Βασικές έννοιες/ενέργειες αλγορίθμων

Δεδομένα, μεταβλητές, τιμές

- Δεδομένα: οι πληροφορίες που επεξεργάζεται/δέχεται ένας αλγόριθμος
 - Π.χ. ο πίνακας στην ταξινόμηση
- Τιμή: η τιμή των δεδομένων (ή μεταβλητών) (π.χ. 7, "N", Αληθές)
 - Μπορεί να έχει διαφορετικούς τύπους: ακέραιος, πραγματικός αριθμός, χαρακτήρας/κείμενο, πίνακες κλπ.
- Μεταβλητή: ένα «κουτάκι» που κρατά μια τιμή και μπορεί να αλλάζει
 - Παράδειγμα: “ηλικία” $\leftarrow$ 20, μετά “ηλικία” $\leftarrow$ 21 (η τιμή αλλάζει)"

Εκφράσεις και τελεστές

- Αριθμητικοί: $+$, $-$, $\times$, $\div$, mod
 - Μόνο μεταξύ αριθμών
- Συγκρίσεις: $=$, $\neq$, $<$, $\leq$, $>$, $\geq$
 - Κυρίως μεταξύ αριθμών
 - Μπορούμε να συγκρίνουμε και κείμενο αν χρειάζεται (κυρίως με $=$, $\neq$)
- Λογικοί: και, ή, όχι
 - Παράδειγμα: Αν $(x > 10)$ και $(y \leq 5)$ τότε ...

Σκελετός αλγορίθμου/ψευδοκώδικα

- Είσοδος: τι μας δίνεται (π.χ. λίστα αριθμών)
- Επεξεργασία: βήματα που εκτελούμε (π.χ. έλεγχος κάθε στοιχείου)
- Έξοδος: τι επιστρέφουμε/εμφανίζουμε (π.χ. μέγιστη τιμή)
- Σκελετός ψευδοκώδικα/αλγορίθμου:
 - Είσοδος: ...
 - Επεξεργασία: ...
 - Έξοδος: ...

Βασικές δομές ελέγχου: Απλή ακολουθία

- εκτέλεση εντολών με τη σειρά
- Παράδειγμα:
 - διάβασε x (Διάβασε είσοδο από τον χρήστη και εκχώρησε την τιμή στην μεταβλητή x)
 - $y \leftarrow x + 1$ (εκχώρησε την τιμή « $x+1$ » στην μεταβλητή y)
 - εμφάνισε y (εμφάνισε την τιμή της μεταβλητής y)
- Δεν υπάρχει διακλάδωση

Βασικές δομές ελέγχου: Επιλογή (Αν/Αλλιώς) (1)

- Επιλογή: εκτέλεση ανάλογα με μια συνθήκη
- Παράδειγμα:
 - Αν $x \geq 18$ τότε
 - εμφάνισε "ενήλικος"
 - αλλιώς
 - εμφάνισε "ανήλικος"
- Μπορούν να υπάρχουν παραπάνω από μια εντολή σε κάθε τμήμα του Αν... τότε... αλλιώς....
 - Αν $x \geq 18$ τότε
 - εμφάνισε "ενήλικος"
 - $y = y + 1$
 - Αλλιώς ...

Βασικές δομές ελέγχου: Επιλογή (Αν/Αλλιώς) (2)

- Μπορούν να υπάρχουν πολλαπλές συνθήκες συνδυασμένες με και/ή
 - Αν $x \geq 18$ και $x < 24$ τότε ...
 - Αν $x \geq 18$ ή $y \geq 10$ τότε
- Επίσης μπορούν να υπάρχουν πολλαπλές περιπτώσεις, χρησιμοποιώντας το «Αλλιώς_αν».
 - Αν $x \geq 18$ τότε ...
 - Αλλιώς αν $x < 16$ τότε ...
 - Αλλιώς ...
- Το τελικό αλλιώς χρησιμεύει ως γενική περίπτωση:
 - Τι κάνουμε σε περίπτωση που καμία προηγούμενη συνθήκη δεν είναι αληθής.

Βασικές δομές ελέγχου: Επανάληψη (Όσο)

- Επανάληψη «Όσο»: επανάλαβε βήματα μέχρι μια συνθήκη να γίνει Ψευδής
 - Όσο <συνθήκη> επανάλαβε
 - <εντολές>
 - Τέλος_επανάληψης (προαιρετικό)
- Η επανάληψη είναι βασική δομή ελέγχου.
 - Όποτε θέλουμε να χρησιμοποιήσουμε μια ακολουθία εντολών πολλές φορές
- Και εδώ μπορούν να συνδυαστούν πολλαπλές συνθήκες με και/ή
- Και όσες εντολές χρειαζόμαστε μέσα στην επανάληψη
- Δεν υπάρχουν όμως διακλαδώσεις με «Αλλιώς ...»
 - Όταν η συνθήκη είναι ψευδής, ο αλγόριθμος συνεχίζει με τις επόμενες εντολές

Βασικές δομές ελέγχου: Επανάληψη (Όσο) (2)

- Παράδειγμα: Βρες άθροισμα των αριθμών $1..n$
 - Είσοδος: n
 - $sum \leftarrow 0$,
 - $i \leftarrow 1$
 - Όσο $i \leq n$:
 - $sum \leftarrow sum + i$,
 - $i \leftarrow i + 1$
 - Έξοδος: sum

Βασικές δομές ελέγχου: Επανάληψη (Για) (1)

- Επανάληψη «Για»
 - για περιπτώσεις όπου **ξέρουμε εκ των προτέρων** πόσες φορές θα γίνει η επανάληψη.
- Γενική μορφή:
 - Για $\langle i \rangle$ από $\langle a \rangle$ μέχρι $\langle b \rangle$ επανάλαβε
 - $\langle \text{εντολές} \rangle$
 - Τέλος_επανάληψης (προαιρετικό)
- $\langle i \rangle$: μεταβλητή, παίρνει διαδοχικά όλες τις τιμές από a έως b .
- $\langle a \rangle$, $\langle b \rangle$: μπορούν να είναι είτε τιμές είτε μεταβλητές
 - Επομένως το «ξέρουμε εκ των προτέρων» μπορεί να μην είναι ακριβές
 - Αλλά να ξέρουμε πως εξαρτάται από μια ή περισσότερες μεταβλητές

Βασικές δομές ελέγχου: Επανάληψη (Για) (2)

- Παράδειγμα: Βρες άθροισμα των αριθμών $1..n$
 - Είσοδος: n
 - $sum \leftarrow 0$
 - Για i από 1 μέχρι n επανάλαβε
 - $sum \leftarrow sum + i$
 - Έξοδος: sum

Βασικές δομές ελέγχου: Επανάληψη (Για) (3)

- $<i>$: μεταβλητή, παίρνει διαδοχικά όλες τις τιμές από a έως b .
 - αυξάνεται αυτόματα κατά 1 σε κάθε βήμα.
 - μπορεί να χρησιμοποιηθεί και με βήμα διαφορετικό από 1:
 - Για i από 1 μέχρι 10 με_βήμα 2 επανάλαβε:
 - ...
- Παράδειγμα: Βρες άθροισμα των άρτιων αριθμών $2..n$
 - Είσοδος: n
 - $sum \leftarrow 0$
 - Για i από 2 μέχρι n με βήμα 2 επανάλαβε
 - $sum \leftarrow sum + i$
 - Έξοδος: sum

Συναρτήσεις/Διαδικασίες ως «κουτιά»

- Ιδέα:
 - ομαδοποιούμε βήματα και τους δίνουμε όνομα
- Οφέλη:
 - επαναχρησιμοποίηση,
 - σαφήνεια,
 - Εύκολος έλεγχος και εντοπισμός λαθών
- Συνάρτηση <ΌνομαΣυνάρτησης>(<παράμετροι>) επιστρέφει <αποτέλεσμα>
 - <εντολές / βήματα επεξεργασίας>
 - Επιστροφή <αποτέλεσμα>
- Τέλος_Συνάρτησης (προαιρετικό)

Συναρτήσεις/Διαδικασίες

- Παράδειγμα:
 - Συνάρτηση Αθροισμα($A[1..n]$), επιστρέφει τιμή sum
 - $sum \leftarrow 0$
 - Για i από 1 μέχρι n επανάλαβε
 - $sum \leftarrow sum + A[i]$
 - Τέλος_επανάληψης
 - Επιστροφή sum
 - Τέλος_Συνάρτησης
- Χρήση συνάρτησης:
 - Είσοδος ΤιμέςΠροϊόντων (πίνακας $A[1..n]$), διαθέσιμο budget (μεταβλητή budget)
 - ΣυνολικόΚόστος $\leftarrow$ Αθροισμα(ΤιμέςΠροϊόντων)
 - Αν ΣυνολικόΚόστος $>$ budget τότε
 - Εμφάνισε «Μπορούμε να τα αγοράσουμε»
 - Αλλιώς
 - Εμφάνισε «Δεν φτάνει το budget για όλα»

Βασικές έννοιες/ενέργειες αλγορίθμων

- Είσοδος/έξοδος
 - Δεδομένα, μεταβλητές, τιμές
- Εκφράσεις και τελεστές
 - Εκχώρηση τιμής
 - Διάβασμα τιμών/Εμφάνιση τιμών
 - Συναρτήσεις

Ροή:

- Απλή ακολουθία
- Συνθήκη (Αν ... Αλλιώς αν ... Αλλιώς)
- Επαναλήψεις
 - Όσο ...
 - Για ...

Συνηθισμένα λάθη και πώς τα αποφεύγουμε

- Off-by-one: λάθος όρια
 - π.χ. i από 1 μέχρι n ή μέχρι $n-1$;
- Μη αρχικοποιημένες μεταβλητές
 - π.χ. `sum` χωρίς `sum  $\leftarrow$  0`
- Σύγκριση vs ανάθεση
 - `=` αντί `$\leftarrow$`
- Μη ξεκάθαρη στοίχιση σε συνθήκη/επανάληψη
- Μη ξεκάθαρη είσοδος/έξοδος

Χειροκίνητη εκτέλεση αλγορίθμων

Σε πολλές περιπτώσεις, ιδιαίτερα κατά την διάρκεια σχεδιασμού αλγορίθμων, δοκιμάζουμε χειροκίνητη εκτέλεση για να επιβεβαιώσουμε την ορθή λειτουργία τους.

- Και σε ασκήσεις που θα βλέπουμε στο μάθημα
- (ή στις εξετάσεις!)

- Επιλέγουμε συγκεκριμένη είσοδο
- Ακολουθούμε τα βήματα του αλγορίθμου όπως περιγράφονται
- Βλέπουμε πώς αλλάζουν τιμές οι μεταβλητές βήμα-βήμα
- Άσκηση: Βρες το μέγιστο αριθμό σε έναν πίνακα $A[1..n]$
 - Σχεδιασμός απλού αλγορίθμου
 - Χειροκίνητη εκτέλεση για δοσμένο πίνακα

Χειροκίνητη εκτέλεση αλγορίθμων

Αλγόριθμος για εύρεση μέγιστου αριθμού σε πίνακα

1. Είσοδος: πίνακας A με n στοιχεία
2. $\max \leftarrow A[1]$
3. Για i από 2 μέχρι n
 1. Αν $A[i] > \max$ τότε
 1. $\max \leftarrow A[i]$
 2. Τέλος_αν
4. Τέλος_Για
5. Επιστροφή $\max$ // μέγιστη τιμή και θέση της

Εκτέλεση για:

$A = [0, 5, 8, 23, 2, 11]$

$A = [1, -15, 2, 5, 8, 11, 13]$

Χειροκίνητη εκτέλεση αλγορίθμων

Αλγόριθμος για αναζήτηση στοιχείου σε πίνακα

1. Είσοδος: πίνακας A με n στοιχεία, ζητούμενο x
2. Για i από 1 μέχρι n
 1. Αν $A[i] = x$ τότε
 1. Επιστροφή i // θέση που βρέθηκε
 2. Τέλος_αν
3. Τέλος_Για
4. Επιστροφή -1 // δεν βρέθηκε

Εκτέλεση για:

A = [0, 5, 8, 2, 11, 23], x=8 και x=12