Εισαγωγή στον Προγραμματισμό

Φροντιστήριο

Αριστείδης Ηλίας

Στόχος

- Κατανόηση της έννοιας του προβλήματος
- Εισαγωγή στους Αλγορίθμους

Προβλήματα και Ανάλυση

Στόχος

- Κατανόηση της έννοιας του προβλήματος
- Σαφήνεια στη διατύπωση
- Αναλυτική σκέψη στην προσέγγιση

- Πρόβλημα είναι μια κατάσταση η οποία χρήζει αντιμετώπισης, απαιτεί λύση, η δε λύση δεν είναι γνωστή, ούτε προφανής
- Για την αντιμετώπιση κάθε προβλήματος πρέπει προηγουμένως να έχει προηγηθεί η κατανόησή του. Αποτελεί συνάρτηση δυο παραγόντων:
  - Σωστή διατύπωση
  - Σωστή ερμηνεία από αυτόν που θα το επιλύσει
- Με τον όρο δομή προβλήματος αναφερόμαστε στα συστατικά του μέρη, στα επιμέρους τμήματα που το αποτελούν καθώς επίσης και στον τρόπο που αυτά συνδέονται μεταξύ τους.

- Ο Νίκος και η Δήμητρα είναι παντρεμένοι
  - Πρώτη ερμηνεία: Ο Νίκος και η Δήμητρα είναι παντρεμένοι μεταξύ τους.
  - Δεύτερη ερμηνεία: Ο Νίκος είναι παντρεμένος και η Δήμητρα είναι παντρεμένη, αλλά όχι μεταξύ τους.

- Διαγραμματική ανάλυση
- Η σωστή επίλυση ενός προβλήματος προϋποθέτει τον επακριβή προσδιορισμό των δεδομένων που παρέχει το πρόβλημα. Απαιτεί επίσης την λεπτομερειακή καταγραφή των ζητουμένων που αναμένονται σαν αποτελέσματα της επίλυσης του προβλήματος
- Τα στάδια αντιμετώπισης εντός προβλήματος είναι τρία
  - Κατανόηση: σωστή και πλήρης αποσαφήνιση των δεδομένων και ζητουμένων του προβλήματος
  - Ανάλυση: το αρχικό πρόβλημα διασπάται σε άλλα επί μέρους απλούστερα προβλήματα
  - Επίλυση: λύση του προβλήματος, με επίλυση των αρχικών προβλημάτων

Κατηγορίες Προβλημάτων

- Τα στάδια αντιμετώπισης εντός προβλήματος είναι τρία
  - Κατανόηση: σωστή και πλήρης αποσαφήνιση των δεδομένων και ζητουμένων του προβλήματος
  - Ανάλυση: το αρχικό πρόβλημα διασπάται σε άλλα επί μέρους απλούστερα προβλήματα
  - Επίλυση: λύση του προβλήματος, με επίλυση των αρχικών προβλημάτων
- Με κριτήριο της δυνατότητα επίλυσης ενός προβλήματος
  - Επιλύσιμα. Τα προβλήματα που η λύση τους είναι γνωστή και έχει διατυπωθεί ή η συνάφειά τους με ήδη επιλυμένα προβλήματα επιτρέπει να θεωρούμε βέβαιη την επίλυσή τους
  - Ανοικτά. Η λύση τους δεν έχει ακόμη βρεθεί αλλά δεν έχει αποδειχθεί ότι δεν λύνονται
  - Άλυτα. Έχει αποδειχθεί ότι δεν επιλύονται
- Με κριτήριο το βαθμό δόμησης ενός προβλήματος
  - Δομημένα. Η επίλυση προέρχεται από μια αυτοματοποιημένη διαδικασία
  - Ημιδομημένα. Η λύση τους επιδιώκεται στα πλαίσια ενός εύρους πιθανών λύσεων αφήνοντας στον ανθρώπινο παράγοντα περιθώρια επιλογής της
  - Αδόμητα. Οι λύσεις τους δεν μπορούν να δομηθούν λη δεν έχει διερευνηθεί η δυνατότητα δόμησής τους
- Με κριτήριο το είδος επίλυσης ενός προβλήματος
  - Απόφασης. Το πρόβλημα πρέπει πάρει μια απόφαση συνήθως σε μια ερώτηση ναι/όχι
  - Υπολογιστικά. Η λύση τους απαιτεί τη διενέργεια πολύπλοκων υπολογισμών
  - Βελτιστοποίησης. Το πρόβλημα επιζητά το βέλτιστο αποτέλεσμα για συγκεκριμένο σετ δεδομένων

Προβλήματα και Υπολογιστής

- Οι υπολογιστές δρουν επικουρικά στην ανθρώπινη δραστηριότητα, δεν έχουν τις δυνατότητες του ανθρώπινου εγκεφάλου.
- Σε αυτούς αναθέτουμε την επίλυση προβλημάτων λόγω:
  - Πολυπλοκότητας υπολογισμών
  - Επαναληπτικότητα διαδικασιών
  - Ταχύτητα εκτέλεσης πράξεων
  - Μεγάλο πλήθος δεδομένων
- Ο υπολογιστής μπορεί να πραγματοποιήσει τις εξής τρεις λειτουργίες
  - Πρόσθεση
  - Σύγκριση (λογικές πράξεις)
  - Μεταφορά δεδομένων

Αλγόριθμοι και Αλγοριθμικές Δομές

Στόχος

- Να διατυπώσουμε τον ορισμό για τον αλγόριθμο και να τον συσχετίσουμε με τη λύση ενός προβλήματος.
- Να διατυπώσουμε τον ορισμό για καθεμιά από τις τρεις βασικές αλγοριθμικές δομές
  - Ακολουθίας
  - Επιλογής (δομές ελέγχου της ροής)
  - Επανάληψης (Βρόγχοι)

  και να περιγράψουμε τη χρήση τους σε αλγορίθμους.
- Να περιγράψουμε τους τρόπους αναπαράστασης των αλγορίθμων
  - Περιφραστική διατύπωση
  - Διαγράμματα για την αναπαράσταση των αλγορίθμων
  - Ψευδοκώδικας
  - Κωδικοποίηση (συγγραφή προγράμματος σε γλώσσα προγραμματισμού)
- Να γνωρίσουμε βασικούς αλγορίθμους, τον τρόπο σκέψης για την σχεδιάση τους
- Να ανακαλύψουμε τις εφαρμογές τους.
- Να ανακαλύψουμε πως μεταβαίνουμε σε γλώσσα προγραμματισμού

Αλγόριθμος

- Αλγόριθμος, είναι μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος
- Κριτήρια που πρέπει να ικανοποιεί ένας αλγόριθμος είναι:
  - Είσοδος (**Καμιά, μία ή περισσότερες τιμές δεδομένων**)
  - Έξοδο (αποτελέσματα για τον χρήστη ή άλλον αλγόριθμο)
  - Περατότητα (πεπερασμένα βήματα, που εκτελούνται σε πεπερασμένο χρόνο)
  - Καθοριστικότητα (σαφώς καθορισμένες εντολές)
  - Αποτελεσματικότητα (κάθε εντολή καλώς ορισμένη είναι και εκτελέσιμη)
- Περιγραφή & αναπαράσταση αλγορίθμων
  - Ελεύθερο κείμενο (free text): αποτελεί τον πιο ανεπεξέργαστο και αδόμητο τρόπο παρουσίασης αλγορίθμου.
  - Διαγραμματικές τεχνικές: συνιστούν ένα γραφικό τρόπο παρουσίασης του αλγορίθμου (η πιο γνωστή είναι το διάγραμμα ροής- flow chart)
  - Βηματική εκτέλεση σε φυσική γλώσσα (natural language): περιγραφή κατά βήματα.
  - Κωδικοποίηση (coding): ένα πρόγραμμα που όταν εκτελεσθεί θα δώσει τα ίδια αποτελέσματα με τον αλγόριθμο (σε ψευδογλώσσα ή γλώσσα προγραμματισμού)

- Μεταβλητές
  - Αριθμητικές: που αποθηκεύουν τιμές όπως 70, -32.5, κ.ο.κ.
  - Αλφαριθμητικές: που αποθηκεύουν τιμές, όπως «Γεώργιος», «John», κ.ο.κ.
  - Λογικές με τιμή αληθής ή ψευδής
- Αποτελούν τον φορέα για τη διατήρηση των Δεδομένων και των Αποτελεσμάτων

- Επιλογή ονομάτων μεταβλητών
  - Τα ονόματα των μεταβλητών μπορούν να περιλαμβάνουν πεζά ή κεφαλαία γράμματα, αριθμούς και το χαρακτήρα κάτω παύλα (_) και άλλα σύμβολα
  - Μπορεί να είναι case sensitive (διακρίνει και διαφοροποιεί τα πεζά από τα κεφαλαία, γιατί έχουν διαφορετική ASCII, ανάλογα με τη γλώσσα και το υπολογιστικό σύστημα που υλοποιείται ο αλγόριθμος)
  - Το όνομα πρέπει να ξεκινά από χαρακτήρα και όχι αριθμό χωρίς κενά. Αν επιθυμούμε την ύπαρξη δυο λέξεων τότε χρησιμοποιείται ή κάτω παύλα π.χ. Μέγιστη_Τιμή
  - Είναι προτιμότερο να είναι περιγραφικά και περιγραστικά ονόματα για να μπορούμε να τα θυμόμαστε (με μέτρο στο πλήθος των χαρακτήρων)
  - Δεν επιτρέπεται να χρησιμοποιηθεί ως όνομα μεταβλητής κάποια από τις δεσμευμένες λέξεις της γλώσσας προγραμματισμού ή ακόμη και της ψευδογλώσσας που χρησιμοποιείται για την απεικόνιση του αλγορίθμου
  - Ουσιαστικά, χρησιμοποιούμε μια μεταβλητή για να αποθηκεύσουμε σε αυτήν κάποια τιμή (είσοδο δεδομένων από το χρήστη ή το αποτέλεσμα κάποιας έκφρασης). Όταν χρησιμοποιούμε την μεταβλητή επικαλούμαστε την τιμή που αυτή περιέχει
  - Αποδεκτά ονόματα μεταβλητών: τιμή1, μέσος_όρος, ΜέγιστηΤιμή, A10 (και σε μεταγραφή με λατινικούς χαρακτήρες για τη C)
  - Μη αποδεκτά ονόματα μεταβλητών: 12a, μέση τιμή
- Σταθερές, πρόκειται για προκαθορισμένες τιμές που παραμένουν αμετάβλητες κατά την εκτέλεση του αλγορίθμου. Και αυτές διακρίνονται σε αριθμητικές, αλφαριθμητικές και λογικές

- Χρησιμοποιείτε όσες μεταβλητές νομίζετε και σας προκύπτει ότι χρειάζεστε!!!

- Άρα προσθέτετε μια κάθε φορά που προκύπτει η ανάγκη, χωρίς φόβο και πάθος και

- Αργότερα και με την αύξηση της εμπειρίας σας (ελπίζουμε σε αυτό το μάθημα και σε αυτό το εξάμηνο!!!) στον προγραμματισμό θα εντοπίζετε πόσες και ποιες πραγματικά χρειάζεστε

- Σημαντικά σημεία για τους αλγόριθμους που θα αναπτύσσετε:
  - Οι εντολές εκτελούνται σειριακά όπως παρατίθενται στον αλγόριθμο/ πρόγραμμα (τουλάχιστον για την ώρα, επειδή αναφερόμαστε σε διαδικασιακό προγραμματισμό)
  - Πρέπει να προβλέπεται κάποιας μορφής εισόδων/ εξόδων
    - Απόδοση τιμών εντός του αλγορίθμου (προγράμματος αργότερα)
    - Πληκτρολόγιο
    - Είσοδος από αρχείο
    - Είσοδος από άλλο τμήμα του προγράμματος
    - Είσοδος από άλλο πρόγραμμα
  - Δεν μπορεί να χρησιμοποιείται ίδιο όνομα για δύο διαφορετικές μεταβλητή
    - Εάν είναι case sensitive μπορεί να γίνει, αλλά θα έχουμε πρόβλημα εμείς
  - Δεσμευμένες λέξεις (εντολές, αλγοριθμικές δομές, κ.λπ.), αυτές οι λέξεις καθώς και όλες όσες είναι έντονα γραμμένες καλούνται δεσμευμένες λέξεις

- Πρόκειται για γνωστά σύμβολα πράξεων (αναλυτικά και στη C θα μιλήσουμε για όλους τους τελεστές σε μερικές εβδομάδες, εδώ εισαγάγουμε όσους χρειαζόμαστε για την αλγοριθμική σκέψη που θα δουλέψουμε σε αυτήν τη φάση)
  - Αριθμητικοί (+, -, *, /, ** για την ύψωση σε δύναμη π.χ. το 8^2 συμβολίζεται με 8**2 (στη C υπάρχουν μερικές διαφοροποιήσεις στον κλασικό τρόπο χρήσης των αριθμητικών- και όχι μόνο τελεστών- αλλά αυτές θα τις αντιμετωπίσουμε αργότερα)
  - Υπάρχουν και οι τελεστές div και mod
    - Ο τελεστής div (διαφορετικές προσεγγίσεις στη C, από / μέχρι div(), div_t(), κ.λπ.) επιστρέφει το **ακέραιο πηλίκο** της διαίρεσης δυο αριθμών
    - Το mod (% στη C) το **ακέραιο υπόλοιπο**

      π.χ. 13 div 3 = 4

      13 mod 3 = 1

      Ο τελεστής mod χρησιμοποιείται ευρέως σε ελέγχους, όπως στις περιπτώσεις όπου πρέπει να διερευνηθεί αν κάποιος αριθμός είναι άρτιος ή περιττός, δηλαδή αν είναι πολλαπλάσιο του 2 ή όχι

      4 mod 2 = 0 ενώ 13 mod 2 = 1

      Παρατηρήσεις για τον τελεστή mod
      - Σε μια πράξη **a mod b** αναμένουμε αποτέλεσμα από 0 μέχρι b-1
      - Αν το a δεν διαιρείται πλήρως με το b τότε παράγει κάποια μη μηδενική ακέραια τιμή
      - Αν το a διαιρείται πλήρως με το b τότε το υπόλοιπο γίνεται 0 (μηδέν)
      - Αν το a είναι κάποιος αριθμός και το b είναι 0, τότε λαμβάνουμε ένα σφάλμα χρόνου μεταγλώττισης (compile-time error)

- Σε περιπτώσεις αρνητικών **a** ή/και **b**, αν και ανάλογα με τη γλώσσα μπορεί να έχουμε διαφοροποιήσεις, θεωρητικά (έτσι υλοποιείται και στη C) αν:
  - ✓ a>0 και b<0 επιστρέφεται τιμή που αντιστοιχεί στο a%(|b|),
  - ✓ a<0 και b>0 επιστρέφεται –((|a|)%b),
  - ✓ a<0 και b<0, με a<b επιστρέφεται –((|a|)%b),
  - ✓ a<0 και b<0, με a>b επιστρέφεται a.

- Συγκριτικοί: >, <, >= , <=, = και <>
- Λογικοί: **ΚΑΙ (AND, &&)**, **Η (OR, ||)**, **ΟΧΙ (NOT, !)** οι οποίοι συνδέουν κυρίως συνθήκες ελέγχου και το αποτέλεσμα είναι πάντα μια λογική τιμή **αληθής** (**true, 1, κ.λπ.**) ή **ψευδής (false, 0, κ.λπ.)**

• Περιπτώσεις Χρήσης DIV και MOD:

- Η απομόνωση των ψηφίων ενός αριθμού απαιτεί συνεχείς διαιρέσεις του αριθμού με τους κατάλληλους αριθμούς (συνήθως 10, 100, 1000, κ.λπ.). Συνήθως ξεκινώντας από τους μεγαλύτερους αριθμούς (διαιρέτες).
- Όταν επιθυμούμε να στρογγυλοποιήσουμε έναν αριθμό.
- Στο έλεγχο εάν ένας αριθμός είναι πολλαπλάσιος ή υποδιαίρεση κάποιου άλλου, οπότε το υπόλοιπο μιας ακέραιας διαίρεσης των δύο αριθμών θα πρέπει να είναι μηδέν. Τέτοιες περιπτώσεις είναι και ο χωρισμός σε συγκεκριμένες κατηγορίες συσκευασιών.
- Στον έλεγχο άρτιων και περιττών.

- Ιεραρχία Τελεστών
  1. δύναμη
  2. πολλαπλασιασμός, διαίρεση, div, mod
  3. πρόσθεση και αφαίρεση
  4. συνθήκες (τελεστές σύγκρισης)
  5. λογικές πράξεις

- ΠΡΟΣΟΧΗ:!!! Η παρένθεση αλλάζει την ιεραρχία αυτή.

- Σε ότι αφορά την ιεραρχία των κατηγοριών των πράξεων, αυτή είναι:
  - Πρώτα εκτελούνται οι αριθμητικοί τελεστές
  - Έπειτα οι συγκριτικοί και τέλος
  - Οι Λογικοί

Εκφράσεις

- Συνδυασμοί μεταβλητών ή σταθερών και τελεστών
- Το αποτέλεσμα μιας έκφρασης αποδίδεται σε μια μεταβλητή με εκχώρηση (ή ανάθεση) τιμής

π.χ. τιμή ← α + β,

- όπου το αποτέλεσμα του αθροίσματος των τιμών των μεταβλητών α και β θα εκχωρηθεί (ανατεθεί) στη μεταβλητή τιμή (το βελάκι δείχνει την ενέργεια)
- Η τελική τιμή μια εκχώρησης εξαρτάται από την ιεραρχία των πράξεων και τις παρενθέσεις
- Αποδεκτές εκφράσεις

τιμή ← 5

τιμή ← «αρκετά»

τιμή ← α * β

τιμή ← τιμή + 3

- Μη αποδεκτές εκφράσεις

τιμή ← 5 + «χ»

α + τιμή ← 6

Να διαβασθούν δύο αριθμοί, να υπολογισθεί και να εκτυπωθεί το άθροισμά τους

```
Αλγόριθμος Άθροισμα
   Διάβασε a
   Διάβασε b
   c ← a + b
   Εκτύπωσε c
Τέλος Άθροισμα
```

Ένας αθλητής πέτυχε στη σφαίρα τις επιδόσεις a, b, c. Να αναπτύξετε αλγόριθμο ο οποίος:

α) θα διαβάζει τις τιμές των επιδόσεων a, b, c.

β) θα υπολογίζει και θα εμφανίζει την μέση τιμή των παραπάνω τιμών.

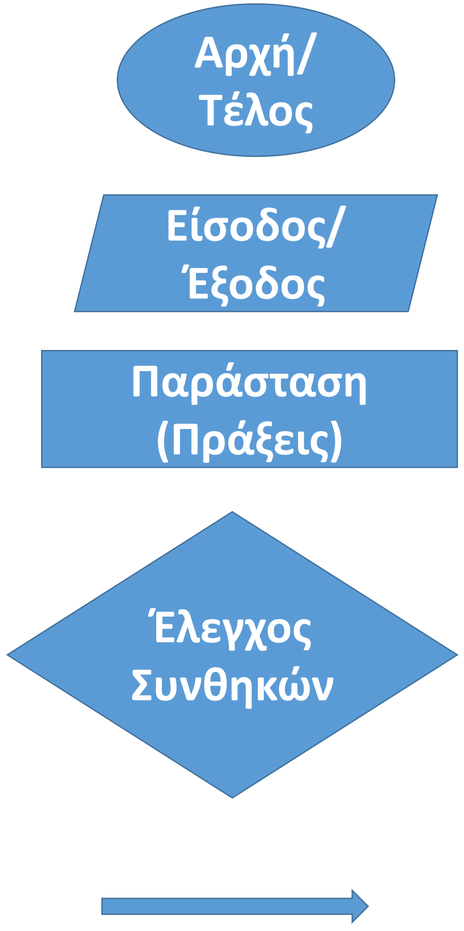
```
Αλγόριθμος Επίδοση
    Διάβασε a, b, c
    Average ←(a + b + c)/3
    Εκτύπωσε Average
Τέλος Επίδοση
```

- Είναι αναγκαίο (τουλάχιστον για τώρα) να σχεδιάζετε έναν αλγόριθμο στο «χαρτί», να εντοπίζετε τις αναμενόμενες τιμές των αποτελεσμάτων κι έπειτα να τον υλοποιείτε
- Υλοποιείτε σε κάποια γλώσσα
- Να επαληθεύετε τα αποτελέσματα και να τα συγκρίνετε με τα αναμενόμενα, π.χ. με πίνακες τιμών
  - πίνακας με τόσες στήλες, όσες και οι μεταβλητές που υπάρχουν και θέλουμε να παρακολουθήσουμε
  - εκτελούμε τις εντολές του αλγορίθμου μια προς μια και όταν μια μεταβλητή αλλάζει τιμή αλλάζουμε γραμμή στον πίνακα και τοποθετούμε τη νέα τιμή

χ ← 2
Α ← 5 * χ
Β ← 2 * χ + 8
Γ ← (Α + Β) mod 2 + 10 div 13
Δ ← 5 * Γ – 4 mod 3
Υ ← 2 * x + Α / 2
Εμφάνισε Α, Β, Γ, Δ

Α	Β	Γ	Δ	Υ	χ
					2
10					2
10	12				2
10	12	0			2
10	12	0	-1		2
10	12	0	-1	9	2

- Σχηματική αναπαράσταση των αλγορίθμων



Ευχαριστώ

Ερωτήσεις?