

Κεφάλαιο 2

Βασικές έννοιες αλγορίθμων

Στο κεφάλαιο αυτό θα κάνουμε μια εισαγωγή στους αλγόριθμους και θα παρουσιάσουμε διαφορετικούς τρόπους με τους οποίους μπορούμε να αναπαράστησουμε αλγόριθμους όπως: ελεύθερη γλώσσα, ψευδοκώδικα, διαγράμματα ροής κτλ.

2.1 Βασικές έννοιες

2.1.1 Ορισμοί

Όταν θέλουμε να επιλύσουμε ένα πρόβλημα μπορούμε να το κάνουμε με δύο τρόπους: είτε με τον αδόμητο τρόπο είτε με τον δομημένο. Στον αδόμητο τρόπο καταφεύγουμε όταν η επίλυση ενός προβλήματος είναι σχετικά απλή και δεν απαιτεί πολλά βήματα για να γίνει. Ωστόσο όταν η κατάσταση δυσκολεύει (όπως πχ στην εύρεση πρώτων αριθμών), τότε καταφεύγουμε στον δομημένο τρόπο και την χρήση αλγορίθμων.

Ορισμός: Αλγόριθμος είναι μια πεπερασμένη σειρά βημάτων, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που έχουν σαν στόχο τους την επίλυση ενός προβλήματος.

Κάθε αλγόριθμος πρέπει να ικανοποιεί τα ακόλουθα κριτήρια:

1. **Είσοδος.** Ο αλγόριθμος πρέπει να διαθέτει μια σειρά από γνωστά στοιχεία τα οποία θα αποκαλούμε είσοδος του αλγορίθμου. Στο παράδειγμα της απόφασης για πρώτους αριθμούς είσοδος θεωρείται ο ακέραιος αριθμός N .
2. **Έξοδος.** Ο αλγόριθμος για να έχει νόημα σαν διαδικασία πρέπει να βγάζει κάποιο αποτέλεσμα με κάποια μορφή, πχ να το εμφανίζει στην οθόνη ή σε κάποιο αρχείο. Αυτό το αποτέλεσμα θεωρείται έξοδος του αλγορίθμου. Στο παράδειγμα με τον πρώτο αριθμό έξοδος θα είναι ένα ΝΑΙ ή ΟΧΙ, για το αν ο αριθμός μπορεί να θεωρηθεί πρώτος.

3. **Καθοριστικότητα.** Τα βήματα του αλγορίθμου θα πρέπει να ορίζονται σε κάθε περίπτωση. Για παράδειγμα αν χρειαστεί σε κάποιο βήμα να γίνει κάποια διαίρεση ο αλγόριθμος θα πρέπει να λαμβάνει υπόψη του και την περίπτωση που γίνεται διαίρεση με το 0.
4. **Περατότητα.** Ο αλγόριθμος θα πρέπει να τελειώνει μετά την εκτέλεση των βημάτων του και να μην οδηγείται σε αοριστία ή ατέρμονες βρόγχους.
5. **Αποτελεσματικότητα.** Κάθε εντολή του αλγορίθμου θα πρέπει να είναι κατά το δυνατόν απλούστερη και να μην χρειάζεται περαιτέρω ανάλυση για να εκτελεστεί.

2.1.2 Μεταβλητές

Είναι οντότητες των αλγορίθμων που απαιτούν χώρο στην μνήμη και διαθέτουν τις ακόλουθες βασικές ιδιότητες:

1. **Όνομα.** Είναι ένα όνομα με το οποίο αναφερόμαστε στην θέση μνήμης, χωρίς να γνωρίζουμε την συγκεκριμένη διεύθυνση.
2. **Τύπο.** Κάθε μεταβλητή μπορεί να λαμβάνει τιμές διαφόρων ειδών, όπως ακέραιους αριθμούς, δεκαδικούς, λογικές τιμές κτλ. Για αυτό τον λόγο είναι μέλος μια μεγαλύτερης ομάδας μεταβλητών που ονομάζεται τύπος.
3. **Τιμή.** Είναι το περιεχόμενο της θέσης μνήμης όπου και βρίσκεται η μεταβλητή.

2.1.3 Ανάγνωση

Αν σε έναν αλγόριθμο δεν αλλάζουν οι τιμές των μεταβλητών τότε τις ονομάζουμε **σταθερές**. Οι σταθερές μπορούν να είναι είναι χρήσιμες σε διάφορες περιπτώσεις, όπως για παράδειγμα ο συντελεστής ΦΠΑ, το πλήθος των μαθημάτων ανά εξάμηνο φοίτησης κτλ. Στις περισσότερες των περιπτώσεων όμως θα πρέπει αυτές οι τιμές να αλλάζουν, προκειμένου ο αλγόριθμος να μπορεί να καλύψει όσο το δυνατόν περισσότερες περιπτώσεις. Για παράδειγμα αν έχουμε ένα πρόγραμμα το οποίο εμφανίζει τον μέσο όρο της βαθμολογίας ενός μαθητή, αυτό το πρόγραμμα θα πρέπει να διαβάζει τις αντίστοιχες βαθμολογίες προκειμένου να προσαρμοστεί στον εκάστοτε μαθητή. Επομένως αυτό που απαιτείται είναι να εκτελούμε κάθε φορά την πράξη της ανάγνωσης προκειμένου να γίνει το πρόγραμμα πιο χρήσιμο.

2.1.4 Εμφάνιση

Για να μπορέσουμε να αντιληφθούμε την απόδοση ενός αλγορίθμου χρειάζεται να εμφανίζουμε στον χρήστη το αποτέλεσμα των εκφράσεων που υπολογίζουμε μέσα στον αλγόριθμο. Αυτό πραγματοποιείται με την χρήση των εντολών εμφάνισης. Αυτές οι εντολές μπορούν είτε να εμφανίζουν το αποτέλεσμα στην καθιερωμένη έξοδο (οθόνη) είτε σε κάποιο αρχείο στον σκληρό δίσκο του υπολογιστή.

2.2 Παρουσίαση αλγορίθμων

Για να μπορέσουμε να εμφανίσουμε οπτικά έναν αλγόριθμο μπορούμε να καταφύγουμε σε διάφορα εργαλεία, όπως αυτά είναι: η ελεύθερη γλώσσα, ο ψευδοκώδικας και τα διαγράμματα ροής.

2.2.1 Ελεύθερη γλώσσα

Είναι ο συνήθης τρόπος με τον οποίο παρουσιάζονται τα προβλήματα σε βιβλία προγραμματισμού. Ωστόσο αυτός ο τρόπος εμπεριέχει τον κίνδυνο της ασάφειας και του αποπροσανατολισμού του προγραμματιστή. Χρησιμοποιείται συνήθως σαν ο πρώτος τρόπος να παρουσιάσει κάποιος ένα πρόβλημα και στην συνέχεια χρησιμοποιείται κάποια άλλη τεχνική, όπως διάγραμμα ροής ή ψευδοκώδικας για την καλύτερη αναπαράσταση του προβλήματος. Για παράδειγμα έστω το εξής πρόβλημα: *να γραφεί αλγόριθμος ο οποίος θα διαβάζει αριθμούς από το πληκτρολόγιο μέχρι να λάβει αρνητικό αριθμό και στο τέλος θα εμφανίζει το άθροισμα των αριθμών που διάβασε.* Σε ελεύθερη γλώσσα θα γράφαμε: θέσε Σ σε 0, διάβασε N και αν ο αριθμός είναι αρνητικός σταμάτα, αλλιώς: πρόσθεσε τον N στο Σ και διάβασε τον επόμενο N μέχρι αυτός να γίνει αρνητικός. Στο τέλος ο αλγόριθμος θα εμφανίσει τον Σ .

2.2.2 Διαγράμματα ροής

Είναι ένας διαγραμματικός τρόπος παρουσίασης του αλγορίθμου, όπου διάφορα μπλοκ συνδέονται μεταξύ τους με βέλη που δείχνουν την πορεία εκτέλεσης του αλγορίθμου. Τα σημαντικότερα στοιχεία των διαγραμμάτων ροής παρουσιάζονται στο σχήμα 2.1. Το σύμβολο της ΑΡΧΗΣ μπαίνει στην εκκίνηση ενός διαγράμματος ροής και το σύμβολο του ΤΕΛΟΥΣ φυσικά όταν τελειώνει ο αλγόριθμος. Κάθε άλλο σύμβολο θα μπει ανάμεσα και όλα τα σύμβολα συνδέονται μεταξύ τους με βέλη των οποίων οι ακμές δείχνουν την πορεία εκτέλεσης του αλγορίθμου. Το σχήμα ΕΙΣΟΔΟΣ/ΕΞΟΔΟΣ χρησιμοποιείται για την είσοδο ή για την εμφάνιση μεταβλητής. Ένα ολοκληρωμένο παράδειγμα είναι αυτό του προβλήματος της άθροισης αριθμών που είδαμε στην προηγούμενη υποενότητα. Το παράδειγμα αυτό παρουσιάζεται στο σχήμα 2.2.

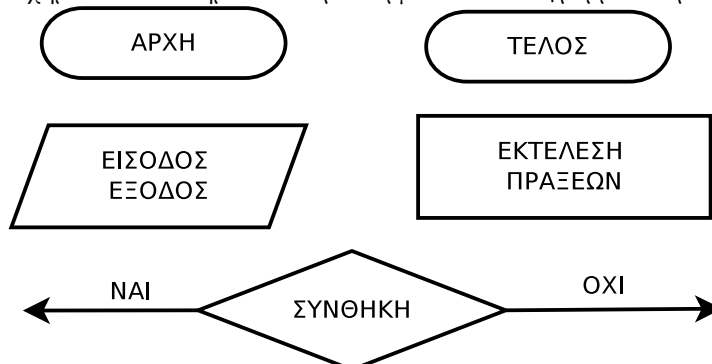
2.2.3 Ψευδοκώδικας

Ο τελευταίος τρόπος εμφάνισης αλγορίθμων και από τους πιο κοινά χρησιμοποιούμενους είναι ο ψευδοκώδικας. Είναι πιο κοντά σε μια τυπική γλώσσα προγραμματισμού από άλλους τρόπους αναπαράστασης, καθώς διαθέτει: απλές εντολές, επιλογείς, ανακυκλώσεις, συναρτήσεις κτλ.

2.2.3.1 Απλές πράξεις

Ένας αλγόριθμος με την χρήση ψευδοκώδικα οριοθετείται πάντα στο μπλοκ
Αλγόριθμος ΟΝΟΜΑ

Σχήμα 2.1: Τα σημαντικότερα σύμβολα των διαγραμμάτων ροής



Αλγόριθμος 2.1 Υπολογισμός αθροίσματος

Αλγόριθμος Άθροισμα **Διάβασε** X **Διάβασε** Y $Z \leftarrow X + Y$ **Τύπωσε** Z**Τέλος** Άθροισμα

Τέλος ΟΝΟΜΑ

Όπου ΟΝΟΜΑ ένα όνομα για τον αλγόριθμο της επιλογής μας, πχ ΜΙΣΘΟΔΟΣΙΑ. Για να διαβάσουμε μεταβλητές προφανώς υπάρχει η εντολή

Διάβασε ΟΝΟΜΑ_ΜΕΤΑΒΛΗΤΗΣ

και αν θέλουμε να εμφανίσουμε μια μεταβλητή υπάρχει το αντίστοιχο

Τύπωσε ΟΝΟΜΑ_ΜΕΤΑΒΛΗΤΗΣ

Ένα απλό παράδειγμα με την χρήση ψευδοκώδικα παρουσιάζεται στον αλγόριθμο 2.1.

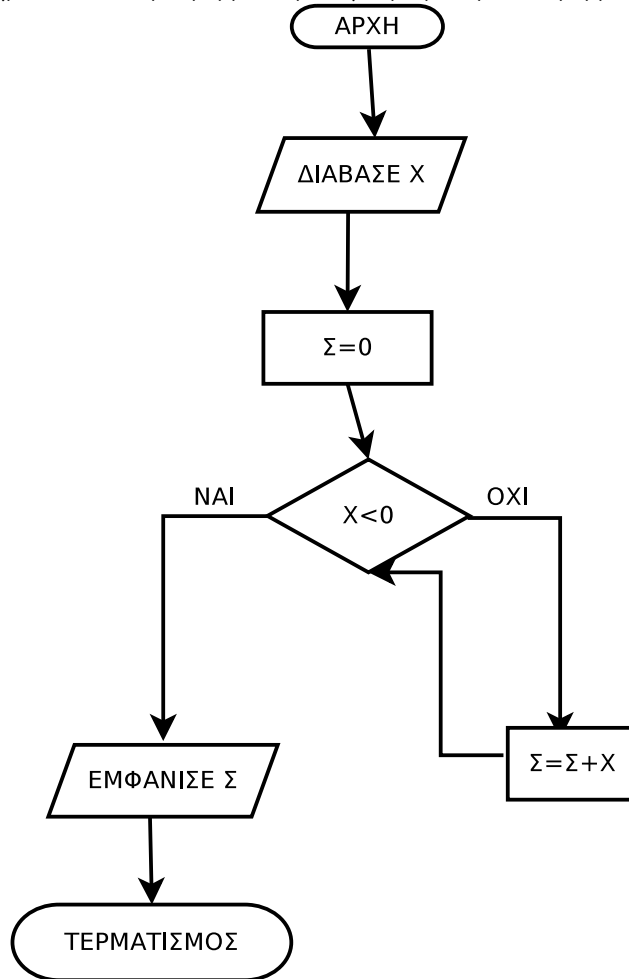
2.2.3.2 Επιλογή

Με την χρήση της επιλογής μπορούμε να ελέγξουμε την τιμή μιας συνθήκης και ανάλογα με την ισχύ της συνθήκης να εκτελεστούν μια σειρά από εντολές. Το γενικό σχήμα της εντολής παρουσιάζονται στον αλγόριθμο 2.2. Σε αυτό το σχήμα αν η συνθήκη στο Αν ισχύει, τότε εκτελείται το μπλοκ εντολών Εντολές-1, διαφορετικά εκτελείται το Μπλοκ εντολών Εντολές-2. Ένα παράδειγμα χρήσης της επιλογής έχουμε στον αλγόριθμο 2.3, όπου δεδομένης μιας ηλικίας ενός ατόμου αποφασίζουμε αν είναι ενήλικος ή ανήλικος.

2.2.3.3 Δομές ανακύκλωσης

Με τον όρο ανακύκλωση εννοούμε την επαναληπτική εκτέλεση εντολών όσο ισχύει ή μέχρι να ισχύσει μια συνθήκη. Η πρώτη δομή επανάληψης που θα παρουσιάσουμε

Σχήμα 2.2: Το πρόγραμμα άθροισης αριθμών με διάγραμμα ροής.



Αλγόριθμος 2.2 Το γενικό σχήμα της επιλογής

Αν συνθήκη **Τότε**

 Εντολές-1

Αλλιώς

 Εντολές-2

Τέλος Αν

Αλγόριθμος 2.3 Πρόγραμμα για την απόφαση ενηλικίωσης

Αλγόριθμος Ενηλικίωση **Διάβασε** Ηλικία **Αν** Ηλικία \geq 18 **Τότε** **Τύπωσε** 'Είσαι Ενήλικος' **Αλλιώς** **Τύπωσε** 'Είσαι Ανήλικος' **Τέλος Αν****Τέλος** Ενηλικίωση

Αλγόριθμος 2.4 Το γενικό σχήμα της εντολής ΟΣΟ

Όσο συνθήκη **Επανάλαβε**

Εντολές

Τέλος_Επανάληψης

είναι η δομή Όσο που εμφανίζεται στον αλγόριθμο 2.4. Σε αυτό το σχήμα όσο η συνθήκη είναι αληθής εκτελούνται οι Εντολές. Προφανώς και πρέπει να υπάρχει κάποιος τρόπος να αλλάζει η συνθήκη προκειμένου η ανακύκλωση να μην εκτελείται για πάντα (ατέρμονας βρόγχος). Ένα παράδειγμα της χρήσεως αυτού του σχήματος έχουμε στον αλγόριθμο 2.5 όπου παρουσιάζεται το πρόβλημα της άθροισης αριθμών αυτή την φορά με την χρήση της ψευδογλώσσας. Μια δεύτερη μορφή ανακύκλωσης παρουσιάζεται στον αλγόριθμο 2.6 όπου οι εντολές επαναλαμβάνονται μέχρι να ισχύσει μια συνθήκη. Το τελευταίο σχήμα επανάληψης που θα εξετάσουμε είναι αυτό της εντολής ΓΙΑ όπου οι εντολές πραγματοποιούνται για ένα συγκεκριμένο αριθμό φορών που καθορίζεται από έναν μετρητή. Το γενικό σχήμα της εντολής ΓΙΑ παρουσιάζεται στον αλγόριθμο 2.7. Ένα παράδειγμα χρήσης της εντολής ΓΙΑ έχουμε στον αλγόριθμο 2.8 όπου με την χρήση της εντολής αθροίζουμε όλους τους αριθμούς από το 10 μέχρι και το 20.

Αλγόριθμος 2.5 Άθροιση αριθμών με την χρήση ψευδογλώσσας

Αλγόριθμος Άθροιση **Διάβασε** X $\Sigma \leftarrow 0$ **Όσο** X > 0 **Επανάλαβε** $\Sigma \leftarrow \Sigma + X$ **Διάβασε** X **Τέλος_Επανάληψης** **Εμφάνισε** Σ **Τέλος** Άθροιση

Αλγόριθμος 2.6 Γενικό σχήμα της εντολής Μέχρι**Επανάλαβε**

Εντολές

Μέχρις_Ότου Συνθήκη

Αλγόριθμος 2.7 Το γενικό σχήμα της εντολής ΓΙΑ**ΓΙΑ** μετρητής από ΤΙΜΗ1 μέχρι ΤΙΜΗ2

Εντολές

Τέλος_Επανάληψης

2.3 Ασκήσεις

1. Παρουσιάστε στην τάξη μια σειρά από προβλήματα και δώστε τις εισόδους και εξόδους που διαθέτει ο καθένας από αυτούς.
2. Παρουσιάστε στην τάξη μια σειρά από πρόβλημα όπου απαιτείται η χρήση σταθερών.
3. Παρουσιάστε στην τάξη ένα πρόβλημα της αρεσκείας σας με την χρήση διαγράμματος ροής.
4. Παρουσιάστε στην τάξη το προηγούμενο πρόβλημα με την χρήση ψευδογλώσσας.

Αλγόριθμος 2.8 Άθροιση αριθμών από το 10 μέχρι και το 20 με την εντολή ΓΙΑ**Αλγόριθμος ΑΘΡΟΙΣΗ**

sum←0

Για i από 10 μέχρι 20

sum←sum+i

Τέλος_Επανάληψης **Εμφάνισε** sum**Τέλος ΑΘΡΟΙΣΗ**