

- Ορισμός

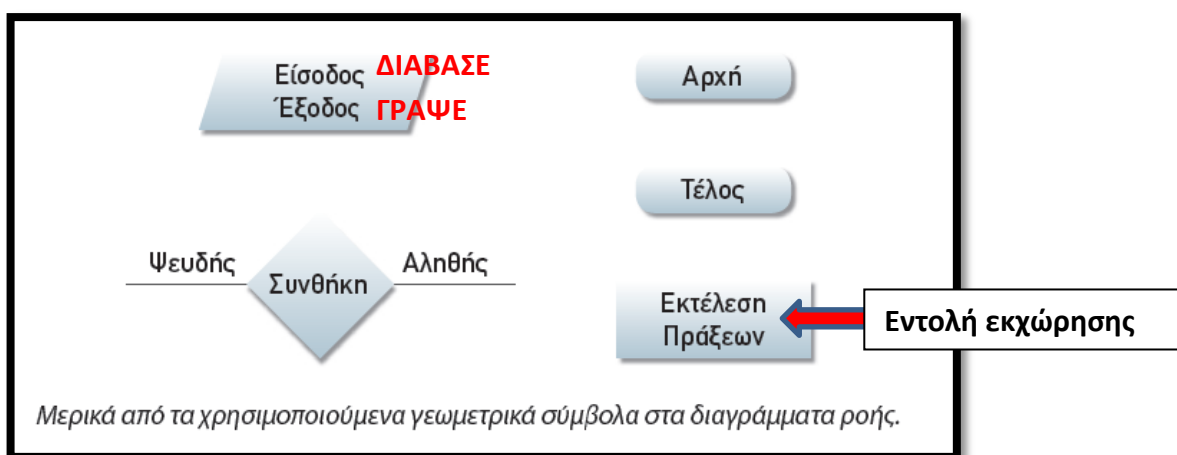
Αλγόριθμος είναι μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος.

- Κάθε αλγόριθμος απαραίτητα ικανοποιεί τα επόμενα κριτήρια:

1. **Είσοδος** (Εντολή **ΔΙΑΒΑΣΕ**)
2. **Έξοδος** (Εντολή **ΓΡΑΨΕ**)
3. **Καθοριστικότητα** (παρονομαστής $\neq 0$ και $x \geq 0$ στο $T_P(X)$)
4. **Περατότητα** (ΟΧΙ ατέρμων βρόχος στις επαναλήψεις)
5. **Αποτελεσματικότητα** (σωστός τύπος υπολογισμού)

- Περιγραφή και αναπαράσταση αλγορίθμων :

1. με **ελεύθερο κείμενο** (παραβιάζει την **αποτελεσματικότητα**)
2. με **διαγραμματικές τεχνικές** (διάγραμμα ροής)



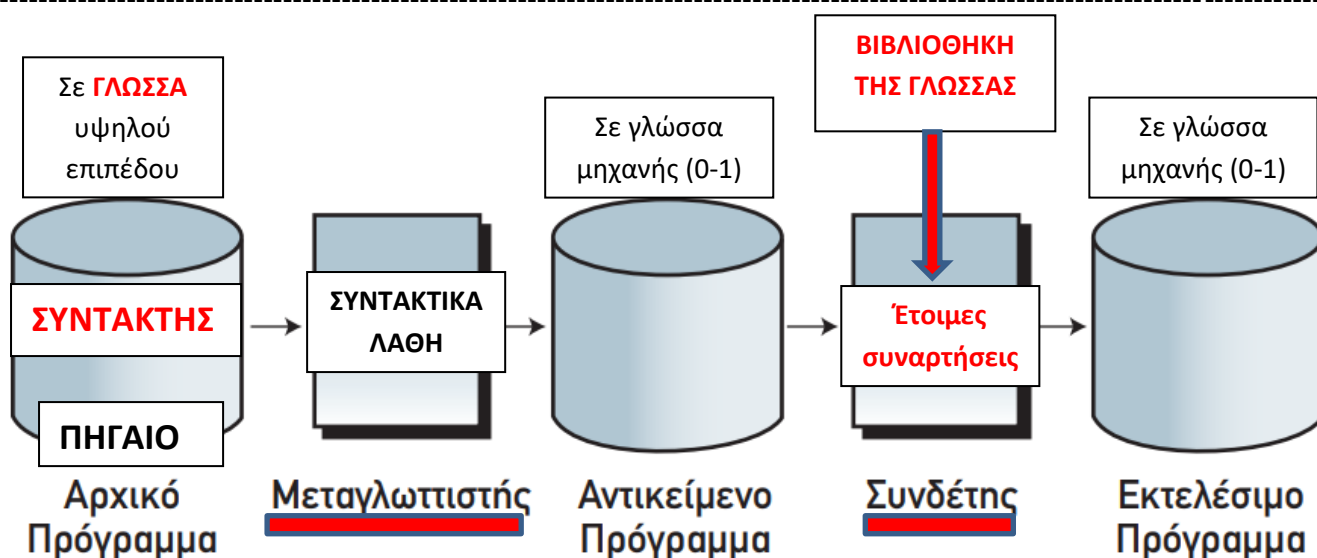
3. με **φυσική γλώσσα κατά βήματα** (παραβιάζει την **καθοριστικότητα**)
4. με **κωδικοποίηση** (σε κάποια γλώσσα προγραμματισμού)

- Στα κυκλώματα του υπολογιστή τα δεδομένα αποθηκεύονται με δυαδική μορφή, δηλαδή 0 και 1. Η **ολίσθηση προς τα αριστερά** ισοδυναμεί με **πολλαπλασιασμό επί δύο**, ενώ η **ολίσθηση προς τα δεξιά** ισοδυναμεί με την **ακέραια διαίρεση διά δύο**.

- 3 Στάδια αντιμετώπισης ενός προβλήματος :

1. **Κατανόηση**, όπου απαιτείται ο **καθορισμός απαιτήσεων** δηλαδή η σωστή και πλήρης αποσαφήνιση των **δεδομένων** και των **ζητούμενων** του προβλήματος.
 2. **Ανάλυση**, όπου το αρχικό πρόβλημα **διασπάται** σε άλλα επιμέρους απλούστερα προβλήματα.
 3. **Επίλυση**, όπου υλοποιείται η λύση του προβλήματος, μέσω της λύσης των επιμέρους προβλημάτων και τη **σύνθεση** αυτών.
- Η **κατανόηση** ενός προβλήματος αποτελεί συνάρτηση δύο παραγόντων, της **σωστής διατύπωσης** εκ μέρους του δημιουργού του και της αντίστοιχα **σωστής ερμηνείας** από τη μεριά εκείνου που καλείται να το αντιμετωπίσει.
 - Μία **βασική διαφορά μεταξύ φυσικών και τεχνητών γλωσσών** είναι η **δυνατότητα εξέλιξής τους**. Οι φυσικές γλώσσες εξελίσσονται συνεχώς. Αντίθετα οι τεχνητές γλώσσες χαρακτηρίζονται από στασιμότητα, αφού κατασκευάζονται συνειδητά για ένα συγκεκριμένο σκοπό.

- Μία γλώσσα προσδιορίζεται από :
 1. Το **Αλφάβητο** (σύμβολα)
 2. Το **Λεξιλόγιο** (σύνολο λέξεων)
 3. Τη **Γραμματική** που αποτελείται από το **τυπικό** (σύνολο κανόνων λέξεων) και το **συντακτικό** (σύνολο κανόνων προτάσεων).
 4. Τη **Σημασιολογία** (το νόημα των λέξεων)
- Η **ιεραρχική σχεδίαση** ή ιεραρχικός προγραμματισμός ή η διαδικασία σχεδίασης "**από επάνω προς τα κάτω**" όπως συχνά ονομάζεται (**top-down program design**) χρησιμοποιεί τη **στρατηγική της συνεχούς διαίρεσης του προβλήματος σε υποπροβλήματα**.
- **GOTO: Το μαύρο πρόβατο του προγραμματισμού**
 Η εντολή GOTO έχει ως αποτέλεσμα την αλλαγή της ροής του προγράμματος, της διακλάδωσης σε μία άλλη εντολή του προγράμματος εκτός από την επόμενη.
 Η εντολή αυτή ήταν υπεύθυνη για τη **δυσκολία στην αρχική σχεδίαση της λύσης**, στην **παρακολούθηση** και **κατανόηση του προγράμματος** και τέλος στη **συντήρηση**. Ο δομημένος προγραμματισμός προήλθε από την ανάγκη του περιορισμού της ανεξέλεγκτης χρήσης του GOTO.
- Ο **δομημένος προγραμματισμός** στηρίζεται στη χρήση **τριών και μόνο λογικών δομών**:
 1. τη δομή της **ακολουθίας**,
 2. τη δομή της **επιλογής** και
 3. τη δομή της **επανάληψης**.
- **Πλεονεκτήματα του δομημένου προγραμματισμού:**
 1. Δημιουργία απλούστερων προγραμμάτων.
 2. Άμεση μεταφορά των αλγορίθμων σε προγράμματα.
 3. Διευκόλυνση ανάλυσης του προγράμματος σε τμήματα.
 4. Περιορισμός των λαθών κατά την ανάπτυξη του προγράμματος.
 5. Διευκόλυνση στην ανάγνωση και κατανόηση του προγράμματος από τρίτους.
 6. Ευκολότερη διόρθωση και συντήρηση.



Σχ. 6.10. Μεταγλώττιση και σύνδεση προγράμματος

- Για την αρχική σύνταξη των προγραμμάτων και τη διόρθωσή τους χρησιμοποιείται ένας μικρός **επεξεργαστής κειμένου**, ο **συντάκτης**.
- Υπάρχουν δύο μεγάλες κατηγορίες ειδικών μεταφραστικών προγραμμάτων οι μεταγλωττιστές και οι διερμηνευτές
Ο μεταγλωττιστής δέχεται στην είσοδο ένα πρόγραμμα γραμμένο σε μια γλώσσα υψηλού επιπέδου και παράγει ένα ισοδύναμο πρόγραμμα σε γλώσσα μηχανής.
Αντίθετα ο διερμηνευτής διαβάζει μία προς μία τις εντολές του αρχικού προγράμματος και για καθεμία εκτελεί αμέσως μια ισοδύναμη ακολουθία εντολών μηχανής.
- Ο μεταγλωττιστής ή ο διερμηνευτής ανιχνεύει τα συντακτικά λάθη και εμφανίζει κατάλληλα διαγνωστικά μηνύματα.
Η χρήση διερμηνευτή έχει το πλεονέκτημα της άμεσης εκτέλεσης και συνεπώς και της άμεσης διόρθωσης.
Η εκτέλεση του προγράμματος με χρήση διερμηνευτή καθίσταται **σημαντικά πιο αργή** από ότι με χρήση μεταγλωττιστή.
- Το αρχικό πρόγραμμα (σε μια γλώσσα υψηλού επιπέδου) λέγεται **πηγαίο** πρόγραμμα, ενώ το πρόγραμμα που παράγεται από το μεταγλωττιστή λέγεται **αντικείμενο** πρόγραμμα (σε γλώσσα μηχανής).
- Το **αντικείμενο** πρόγραμμα συνήθως χρειάζεται να συνδεθεί με άλλα τμήματα που βρίσκονται στις **βιβλιοθήκες της γλώσσας**. Το πρόγραμμα **συνδέτης** – φορτωτής επιτρέπει αυτή τη σύνδεση και παράγει το **εκτελέσιμο** πρόγραμμα.
- Η «**Διαίρει και Βασίλευε**» είναι μια μέθοδος σχεδίασης αλγορίθμων στην οποία ένα πρόβλημα υποδιαιρείται σε μικρότερα υποπροβλήματα με την ίδια τυποποίηση. Η σταδιακή επίλυση των μικρότερων υποπροβλημάτων, συνθέτει τη συνολική λύση του αρχικού προβλήματος. Ονομάζεται σχεδίαση «**από πάνω προς τα κάτω**» (**top-down**).
- Ο **μέγιστος αριθμός των συγκρίσεων** (επαναλήψεων) που απαιτούνται για την εύρεση ενός στοιχείου σε ένα σύνολο «**η**» **ταξινομημένων στοιχείων**, συμπεριλαμβανομένης και της περίπτωσης μη ύπαρξης του στοιχείου, δίνεται από το ακέραιο μέρος του $\lceil \log_2(n)+1 \rceil$ (με στρογγυλοποίηση προς τα κάτω).

« η » ταξινομημένων στοιχείων	<u>ΣΕΙΡΙΑΚΗ ΑΝΑΖΗΤΗΣΗ</u> μέγιστος αριθμός συγκρίσεων	<u>ΔΥΑΔΙΚΗ ΑΝΑΖΗΤΗΣΗ</u> μέγιστος αριθμός συγκρίσεων
10	10	4 (10<16=2 ⁴)
20	20	5 (20<32=2 ⁵)
30	30	5 (30<32=2 ⁵)
50	50	6 (50<64=2 ⁶)
70	70	7 (70<128=2 ⁷)
100	100	7 (100<128=2 ⁷)
200	200	8 (200<256=2 ⁸)
1000	1000	10 (1000<1024=2 ¹⁰)
2000	2000	11 (2000<2048=2 ¹¹)