

**Ανάπτυξη Εφαρμογών  
σε Προγραμματιστικό Περιβάλλον**

**Τόμος 1ος**

**Ομάδα Συγγραφής**

**ΑΘΗΝΑ ΒΑΚΑΛΗ,   
Λέκτωρ Πληροφορικής ΑΠΘ**

**ΗΛΙΑΣ ΓΙΑΝΝΟΠΟΥΛΟΣ,   
Μηχανικός Πληροφορικής**

**ΝΕΣΤΩΡ ΙΩΑΝΝΙΔΗΣ,   
Επίκουρος Καθηγητής Τμήματος**

**ΧΡΗΣΤΟΣ ΚΟΙΛΙΑΣ,   
Επίκουρος Καθηγητής Τμήματος Πληροφορικής ΤΕΙ Αθήνας**

**ΚΩΝ/ΝΟΣ ΜΑΛΑΜΑΣ, M.Sc. Πληροφορικής, Σύμβουλος Επιχειρήσεων,**

**ΙΩΑΝΝΗΣ ΜΑΝΩΛΟΠΟΥΛΟΣ, Αναπληρωτής Καθηγητής Τμήματος Πληροφορικής ΑΠΘ**

**ΠΑΝΑΓΙΩΤΗΣ ΠΟΛΙΤΗΣ,   
Δρ. Διδακτικής Πληροφορικής, Καθηγητής ΠΕ 19**

**Υπεύθυνος για το Παιδαγωγικό Ινστιτούτο  
ΓΕΩΡΓΙΟΣ ΠΑΠΑΔΟΠΟΥΛΟΣ, ΣΥΜΒΟΥΛΟΣ Π.Ι. (κατά τη συγγραφή)**

**Υπεύθυνος Μαθήματος  
ΑΔΑΜ Κ. ΑΓΓΕΛΗΣ,   
Πάρεδρος Πληροφορικής Π.Ι.**

**Επιτροπή Αξιολόγησης  
ΚΩΝ/ΝΟΣ ΓΙΑΛΟΥΡΗΣ,   
Καθηγητής ΠΕ 19   
ΕΥΑΓΓΕΛΙΑ ΓΟΥΛΗ,   
Καθηγήτρια ΠΕ 19   
ΔΗΜΗΤΡΙΟΣ ΔΕΣΠΟΤΗΣ, Αναπληρωτής Καθηγητής Πανεπιστημίου Πειραιώς   
ΚΩΝ/ΝΟΣ ΖΑΧΑΡΗΣ,   
ΠΛΗΝΕΤ Καρδίτσας   
ΑΛΕΞΑΝΔΡΟΣ ΣΙΔΕΡΙΔΗΣ, Καθηγητής Γεωπονικού Πανεπιστημίου Αθηνών**

**Εικονογράφηση   
ΑΓΓΕΛΟΣ ΑΓΙΟΣΤΡΑΤΙΤΗΣ**

**Ηλεκτρονική σελιδοποίηση   
ΑΝΝΑ ΑΡΓΥΡΟΠΟΥΛΟΥ - ΧΡΗΣΤΟΣ ΠΙΓΚΑΣ**

**Εξώφυλλο:   
ΣΠΥΡΟΣ ΣΙΑΚΑΣ - ΝΙΚΟΣ ΓΕΩΡΓΙΟΥ**

**Επιμέλεια:   
ΕΚΔΟΣΕΙΣ ΝΕΩΝ ΤΕΧΝΟΛΟΓΙΩΝ, Στουρνάρη 49Α, 106 82, Αθήνα, Τηλ. 38.45.594**

**Φορέας:**  
**ΕΛΛΗΝΙΚΗ ΕΤΑΙΡΙΑ ΕΠΙΣΤΗΜΟΝΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ (ΕΠΥ), Μαυρομιχάλη 16, Αθήνα,   
τηλ.: 3645274, e.mail:epy@epy.gr**

**Συντονιστές έργου:**

**Σπ. Μπακογιάννης, πρόεδρος Δ.σ. Βασ. Μιχαλακόπουλος, μέλος Δ.Σ.**

**Ενέργεια 1.1.α: «Προγράμματα Βιβλία»**

**Επιστημονικός Υπεύθυνος Ενέργειας**

**Θεόδωρος Γ. Εξαρχάκος,**

**Καθηγητής του Πανεπιστημίου**

**Αθηνών, Πρόεδρος του Παιδαγωγικού Ινστιτούτου**

**Έργο No 15: «Αναμόρφωση / εκ νέου σύνταξη και συγγραφή Προγραμμάτων Σπουδών και Σχολικών Βιβλίων για το Ενιαίο Λύκειο»**

**Επιστημονικός Υπεύθυνος Έργου**

**Γιάννης Σαλβαράς,**

**Επίκουρος Καθηγητής του**

**Δημοκρίτειου Πανεπιστημίου Θράκης**

**Καλλιτεχνικός Υπεύθυνος Έργου**

**Σπύρος I. Παπασπύρου**

**Καθηγητής Εφαρμογών του ΤΕΙ Ηπείρου**

**Η επανέκδοση του παρόντος βιβλίου πραγματοποιήθηκε από το Ινστιτούτο Τεχνολογίας Υπολογιστών & Εκδόσεων «Διόφαντος» μέσω ψηφιακής μακέτας, η οποία δημιουργήθηκε με χρηματοδότηση από το ΕΣΠΑ / ΕΠ «Εκπαίδευση & Διά Βίου Μάθηση» / Πράξη «ΣΤΗΡΙΖΩ».**

**Οι αλλαγές που ενσωματώθηκαν στην παρούσα επανέκδοση έγιναν με βάση τις διορθώσεις του Παιδαγωγικού Ινστιτούτου**

**ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ INΣΤΙΤΟΥΤΟ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΟΛΙΤΙΚΗΣ**

**Α. ΒΑΚΑΛΗ, Η. ΓΙΑΝΝΟΠΟΥΛΟΣ, Ν. ΙΩΑΝΝΙΔΗΣ, Χ. ΚΟΙΛΙΑΣ, Κ. ΜΑΛΑΜΑΣ, Ι. ΜΑΝΩΛΟΠΟΥΛΟΣ, Π. ΠΟΛΙΤΗΣ**

**Η συγγραφή και η επιμέλεια του βιβλίου πραγματοποιήθηκε υπό την αιγίδα του Παιδαγωγικού Ινστιτούτου**

**Ανάπτυξη Εφαρμογών**

**σε Προγραμματιστικό Περιβάλλον**

**Βιβλίο Μαθητή**

**Γ' Γενικού Λυκείου**

**(Τεχνολογικής Κατεύθυνσης)**

**ΙΝΣΤΙΤΟΥΤΟ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ   
ΕΚΔΟΣΕΩΝ «ΔΙΟΦΑΝΤΟΣ»**

**ΠΡΟΣΑΡΜΟΓΗ ΤΟΥ ΒΙΒΛΙΟΥ ΓΙΑ ΜΑΘΗΤΕΣ ΜΕ ΜΕΙΩΜΕΝΗ ΟΡΑΣΗ**

**Ομάδα εργασίας για το Ινστιτούτου Εκπαιδευτικής**

**Πολιτικής**

**Προσαρμογή: Μαρία Γκλεζάκου, Εκπαιδευτικός**

**Eπιμέλεια: Αρετή Ανδριώτου, Εκπαιδευτικός**

**Επιστημονικός υπεύθυνος: Βασίλης Κουρμπέτης,**

**Σύμβουλος Α΄ του Υ.ΠΟ.ΠΑΙ.Θ**

**Υπεύθυνη του έργου: Μαρία Γελαστοπούλου,**

**M.Ed. Ειδικής Αγωγής**

**Τεχνική υποστήριξη: Κωνσταντίνος Γκυρτής,**

**Δρ. Πληροφορικής**

|  |
| --- |
| **Πρόλογος** |

**Το βιβλίο αυτό απευθύνεται στους μαθητές Γ' Τάξης Τεχνολογικής Κατεύθυνσης Ενιαίων Λυκείων, που παρακολουθούν το μάθημα "Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον" του Κύκλου Πληρο-φορικής και Υπηρεσιών.**

**Το μάθημα "Ανάπτυξη Εφαρμογών σε Προγραμματι-στικό Περιβάλλον" έχει σαν γενικό σκοπό οι μαθητές να αναπτύξουν αναλυτική και συνθετική σκέψη, να από-κτήσουν ικανότητες μεθοδολογικού χαρακτήρα και να μπορούν να επιλύουν απλά σχετικά προβλήματα.**

**Όλη η θεωρητική πλευρά του μαθήματος καλύπτεται από αυτό το βιβλίο. Περιλαμβάνει 14 κεφάλαια, που μπορούν να χωριστούν σε δύο μέρη. Το πρώτο μέρος (κεφάλαια 1-5) αναφέρεται στις ενότητες Ανάλυση Προβλήματος και Σχεδίαση αλγορίθμου, όπου η έμφαση δίνεται στην ανάπτυξη δεξιοτήτων αλγο-ριθμικής προσέγγισης των προβλημάτων. Το δεύτερο μέρος αφιερώνεται στην υλοποίηση προγραμμάτων τόσο σε περιβάλλον γλωσσών προγραμματισμού υψηλού επιπέδου όσο και σε αντικειμενοστραφές.**

**Τα δύο αυτά μέρη του βιβλίου δεν είναι ανεξάρτητα μεταξύ τους. Συνήθως ο σκοπός της δημιουργίας ενός αλγορίθμου είναι στη συνέχεια η κατασκευή ενός προγράμματος. Έτσι το βιβλίο αυτό δεν προορίζεται για να διαβαστεί σειριακά. Ο μαθητής θα ακολουθεί τις υποδείξεις του καθηγητή σχετικά με τη σειρά μελέτης των κεφαλαίων. Ας σημειωθεί δε ότι συχνά το ίδιο αντικείμενο μπορεί να επαναλαμβάνεται και σε άλλο σημείο του βιβλίου, αν πρόκειται για θέμα που αντιμε-τωπίζεται από αλγοριθμική σκοπιά αλλά και από την πλευρά της υλοποίησης σε υπολογιστή.**

**Το βιβλίο προσφέρει στο μαθητή όλες τις γνώσεις και πληροφορίες που είναι απαραίτητες, ώστε αυτός να κατανοήσει με ευκολία, ακρίβεια και σαφήνεια τις βασικές έννοιες αλγοριθμικής και προγραμματισμού. Η προσέγγιση των εννοιών γίνεται μέσα από πολλά παραδείγματα σε συσχέτιση με άλλα μαθήματα και γνωστικά αντικείμενα.**

**Στο βιβλίο δεν αναλύονται τεχνικές ή άλλες λεπτομέ-ρειες συγκεκριμένου λογισμικού (γλωσσών προγραμ-ματισμού). Ωστόσο δεν αποφεύγονται κάποιες αναφο-ρές σε γνωστά προγραμματιστικά περιβάλλοντα, που γίνονται για λόγους πληρότητας. Η ανάπτυξη των προ-γραμμάτων που αναφέρονται ως παραδείγματα γίνεται σε μια υποθετική γλώσσα προγραμματισμού, η οποία βέβαια ακολουθεί τις γενικές αρχές των σύγχρονων πραγματικών γλωσσών προγραμματισμού. Η υποθε-τική αυτή γλώσσα αποκαλείται ΓΛΩΣΣΑ και όπως θα γίνει αμέσως φανερό, η μετατροπή ενός προγράμματος από τη ΓΛΩΣΣΑ σε μια πραγματική γλώσσα προγραμ-ματισμού είναι απλή υπόθεση.**

**Για την υποβοήθηση της αναγνωσιμότητας, εκτός από σχήματα, πίνακες και διάφορα πλαίσια, έχουν χρησιμο-ποιηθεί και αρκετά εικονίδια τα οποία χαρακτηρίζουν το μέρος του κειμένου που συνοδεύουν. Τα εικονίδια αυτά και η σημασία τους είναι:**

**Χρήσιμη Πληροφορία**

|  |  |
| --- | --- |
|  | **Η συνθήκη είναι μια λογική έκφραση** |

**Σημείωση**

|  |  |
| --- | --- |
| **πιν.jpg** | **Η εντολή Αρχή\_επανάληψης...**  **Μέχρις\_ότου εκτελείται οπωσδήποτε μια φορά.** |

**Συμβουλή**

|  |  |
| --- | --- |
|  | **Η σχεδίαση ενός καλού τρόπου επικοινωνίας με τον χρήστη είναι το μισό μιας καλής εφαρμογής ειδικά στο χώρο των εφαρμογών πολυμέσων.** |

**Προσοχή**

|  |  |
| --- | --- |
| **τρομακτικο.jpg** | **Σε μια εντολή εκχώρησης η μεταβλητή και η έκφραση πρέπει να είναι του ίδιου τύπου.** |

**Κεφάλαιο 1**

**Ανάλυση**

**Προβλήματος**

**ΠΕΡΙΕΧΟΜΕΝΑ**

|  |  |
| --- | --- |
|  | **Η έννοια πρόβλημα** |
|  | **Κατανόηση προβλήματος** |
|  | **Δομή προβλήματος** |
|  | **Καθορισμός απαιτήσεων** |
|  | **Κατηγορίες προβλημάτων** |
|  | **Πρόβλημα και υπολογιστής** |

**5 / 13**

**• Εισαγωγή**

**Το πρόβλημα αποτελεί έννοια που απαντάται σε όλες τις επιστήμες και τους κλάδους τους, αλλά παράλληλα και στην καθημερινή μας ζωή. Τόσο η αντιμετώπιση, όσο και η διατύπωση ενός προβλήματος, αποτελούν διαδικασίες που απαιτούν ιδιαίτερες αναλυτικές και συνθετικές ικανότητες, ορθολογική σκέψη, αλλά και σωστό και εμπεριστατωμένο χειρισμό της φυσικής γλώσσας. Οι δεξιότητες που αποκτούνται από τους μαθητές μέσω της ενασχόλησής τους με την ανάλυση και τον ορισμό προβλημάτων, αποτελούν εφόδια γενικής χρηστικότητας, αφού μπορούν να λογίζονται σαν γνωστικά εργαλεία χρήσιμα για κάθε δραστη-ριότητα που είτε διαπερνά όλο το φάσμα των επιστη-μών, είτε αναφέρεται σε καθημερινές καταστάσεις.**

**• Διδακτικοί στόχοι**

**Στόχοι του κεφαλαίου αυτού είναι :**

* **να μπορούν καταρχήν οι μαθητές να κατανοούν πλήρως τα προβλήματα που τους τίθενται**
* **να μπορούν στη συνέχεια να προσδιορίζουν τα συστατικά μέρη ενός προβλήματος**
* **να μπορούν να αναλύουν ένα πρόβλημα σε άλλα απλούστερα**

**6 / 14**

* **να καταστούν ικανοί να προσδιορίζουν τα δεδομένα που τους παρέχονται για την αντιμετώ-πιση του προβλήματος**
* **να μπορούν να προσδιορίζουν τα ζητούμενα αποτελέσματα και τη μορφή απόδοσής τους**
* **να είναι σε θέση να θέσουν οι ίδιοι προβλήματα διατυπώνοντάς τα με ακρίβεια και πληρότητα**

**• Προερωτήσεις**

* **θεωρείς σημαντικό το γεγονός να μιλάς και να γράφεις πολύ καλά τη φυσική γλώσσα στην προσπάθειά σου να επιλύσεις ένα τυχαίο πρό-βλημα;**
* **έχεις ακούσει για "το πρόβλημα του έτους 2000";**
* **υπάρχει νοηματική διαφορά ανάμεσα στους όρους δεδομένο και πληροφορία;**
* **όταν αναφερόμαστε σε προβλήματα, έμμεσα δηλώνουμε την ανάγκη χρήσης υπολογιστών για την αντιμετώπισή τους;**

**7 / 14**

|  |  |
| --- | --- |
| **1.1** | **Η έννοια πρόβλημα** |

**Η καθημερινή εμπειρία και πρακτική μας εμφανίζει πολλά και ποικίλα προβλήματα που μας απασχολούν είτε στον προσωπικό μας χώρο, είτε στον κοινωνικό μας χώρο γενικότερα. Οι στατιστικές και οι δημοσκο-πήσεις, που βλέπουν κατά καιρούς το φως της δημο-σιότητας, καταγράφουν τα σημαντικότερα προβλήματα που απασχολούν το κοινωνικό σύνολο. Οι περισ-σότεροι από εμάς σήμερα, σε μια πιθανή έρευνα σχε-τική με τα κυριότερα προβλήματα που απασχολούν την ελληνική κοινωνία, θα απαντούσαμε πως ανάμεσα στα βασικότερα προβλήματα βρίσκονται η ανεργία, τα ναρ-κωτικά, η ξενοφοβία. Όπως επίσης σε μια παρόμοια έρευνα που θα επιχειρούσε να εντοπίσει τα κυριότερα παγκόσμια προβλήματα που απασχολούν την ελληνική κοινωνία, θα απαντούσαμε πως σαν κύρια προβλήματα θεωρούμε τον πόλεμο και τη μόλυνση του περιβάλλοντος.**

**ΟΡΙΣΜΟΣ**

**Με τον όρο Πρόβλημα εννοείται μια κατάσταση η οποία χρήζει αντιμετώπισης, απαιτεί λύση, η δε λύση της δεν είναι γνωστή, ούτε προφανής.**

**Άλλου είδους απαντήσεις, που θα βρισκόντουσαν σε άλλο φάσμα, άλλο εύρος, θα δίναμε αν η ερώτηση αφορούσε στα προσωπικά μας προβλήματα. Πολλοί από μας πιθανό να ανέφεραν επαγγελματικά προβλή-ματα με τον εργοδότη τους, προσωπικά προβλήματα στη σχέση τους ή προβλήματα που σχετίζονται με απλά θέματα της καθημερινής ζωής.**

**Η ύπαρξη προβλημάτων δεν αποτελεί χαρακτηριστικό γνώρισμα της εποχής μας. Μια απλή περιήγηση ανά τις σελίδες της ιστορίας, αρκεί για να μας επιβεβαιώσει ότι σε κάθε εποχή αναφέρονται προβλήματα διαφορετικής υφής και εμβέλειας.**

**8 / 15**

* **Ο Όμηρος στην Ιλιάδα περιγράφει το πρόβλημα που αντιμετώπιζαν οι Έλληνες πολιορκητές της Τροίας, μέχρι ο Οδυσσέας να επινοήσει το Δούρειο Ίππο.**
* **Το πρόβλημα μέτρησης του χρόνου, που αποτελούσε ταυτόχρονα ανθρώπινη ανάγκη, ήρθε να αντιμετωπίσει η εμφάνιση της κλεψύδρας και του εκκρεμούς.**
* **Προβλήματα κοινωνικής αδικίας και εκμετάλ-λευσης ήταν αυτά που οδήγησαν στην εμφά-νιση του Robin Wood στα δάση της επαρχίας του Nottingham.**
* **Το πρόβλημα με το ψύχος που αντιμετώπισαν τα στρατεύματα του Ναπολέοντα στην εκστρα-τεία του στη Ρωσία, είχε σαν αποτέλεσμα την ανακοπή της προέλασης και την οπισθοχώ-ρηση του.**

**9 / 15**

**Ένα από τα σημαντικότερα προβλήματα στο χώρο των υπολογιστών είναι αυτό που αναφέρεται σαν πρόβλη-μα του έτους 2000 (millennium bug). Το πρόβλημα εντοπίζεται στο ότι οι υπολογιστές μετρούν την ημε-ρομηνία μόνο με δύο στοιχεία για κάθε ένα από τα τρία συνθετικά της. Για παράδειγμα, η ημερομηνία 15 Απριλίου 1999 συμβολίζεται με τον κωδικό 150499.**

**Από τις πρώτες περιόδους λειτουργίας των υπολο-γιστών, τότε που γινόταν κάθε δυνατή προσπάθεια να εξοικονομηθεί πολύτιμος αποθηκευτικός χώρος, καθιερώθηκε η καταγραφή της ημερομηνίας με τον παραπάνω τρόπο. Οπότε η πρώτη μέρα του 21ου αιώνα θα συμβολίζεται με τον κωδικό 010100, πράγμα που θα επιφέρει μεγάλη αναστάτωση και σύγχυση στους υπολογισμούς που θα πραγματοποιούν οι υπολογιστές.**

**Η αυγή του 2000 απειλεί να "τρελάνει" τους υπολο-γιστές. Οι αυτόματες μηχανές συναλλαγών (ATM) των τραπεζών μπορεί να μην δίνουν λεφτά ή να δίνουν απίστευτα ποσά που να μην ανταποκρίνονται στις πραγματικές καταθέσεις των πελατών. Τα μηχανο-γραφημένα λογιστήρια των επιχειρήσεων μπορεί να αποδίδουν τρελούς πίνακες οικονομικών στοιχείων που καμία σχέση να μην έχουν με την πραγματικότητα. Ένα ευρύ φάσμα κοινωνικών υπηρεσιών - ασφάλιση, υγειoνομική περίθαλψη, παροχή ενέργειας, μεταφορές, κλπ - παρέχονται μέσα από χρήση πολύπλοκων υπολογιστικών συστημάτων, που απειλούνται από το πρόβλημα του έτους 2000. Το ζήτημα όσο και αν φαί-νεται απλό, στην πραγματικότητα είναι πολύπλοκο και κυρίως μπορεί να έχει πλευρές που δεν μπορούν να προβλεφθούν.**

**10 / 16**

**Οι επιπτώσεις του προβλήματος του έτους 2000 μπορεί να είναι πολύ μεγάλες. Το πρόβλημα πρέπει να αντιμε-τωπιστεί με καθαρά τεχνικούς τρόπους, δεν παύει όμως να έχει και λειτουργικές, οικονομικές και νομικές προεκτάσεις.**

* **Σοβαρότατα προβλήματα επιδημιών, όπως η πανούκλα, η χολέρα και η λύσσα, αφάνιζαν καθημερινά χιλιάδες ανθρώπους τον περασμένο αιώνα μέχρις ότου επιστήμονες, όπως ο Pasteur και ο Fleming, να ανακαλύψουν τα κατάλληλα εμβόλια.**
* **Το πρόβλημα της μεταφοράς της ηλεκτρικής ενέργειας από τον τόπο παραγωγής στα σημεία κατανάλωσης πονοκεφάλιασε πολύ τους υπεύ-θυνους περασμένων εποχών μέχρι να εμφανιστούν οι μετασχηματιστές οι οποίοι έδωσαν λύση στο πρόβλημα.**
* **Το ενεργειακό πρόβλημα από την άποψη των αποθεμάτων που απασχόλησε έντονα, αλλά και εξακολουθεί να απασχολεί την παγκόσμια κοινό-τητα, οδήγησε στην υιοθέτηση ήπιων μορφών ενέργειας, όπως είναι η ηλιακή ενέργεια, η αιολική ενέργεια και η βιομάζα.**
* **Το πρόβλημα της τρύπας του όζοντος, και κατ' επέκταση το πρόβλημα της προστασίας του φυσικού περιβάλλοντος, αντιμετωπίστηκε σε πρώτο βαθμό με τον περιορισμό εκπομπής χλωροφθορανθράκων από τις βιομηχανικές μονάδες που αποτελούν την κύρια αιτία του προβλήματος.**
* **Φυσικά φαινόμενα, όπως εκρήξεις ηφαιστείων, παλιρροιακά κύματα, σεισμοί και τυφώνες, αποτελούν σημαντικά προβλήματα ακόμα και στην εποχή μας, με αποτέλεσμα οι πληθυσμοί των περιοχών που πλήττονται να μετρούν ανθρώπινα θύματα, να υπόκεινται οικονομική καταστροφή και να αναγκάζονται πολλές φορές σε μετακίνηση.**

**11 / 15-16**

* **Ο υποσιτισμός ενός πολύ μεγάλου μέρους του πληθυσμού της αφρικανικής κύρια ηπείρου, οι καθημερινοί θάνατοι πολλών ανθρώπων, ειδικά μικρών παιδιών, αποτελεί ένα από τα σοβαρότερα προβλήματα της ανθρωπότητας σήμερα, χωρίς να έχει μπορέσει να αντιμετωπιστεί επαρκώς από τις ανθρωπιστικές οργανώσεις και τους διεθνείς οργανισμούς.**
* **Η αργή ταχύτητα μετάδοσης των δεδομένων σε σχέση με τις απαιτήσεις της σύγχρονης τεχνο-λογίας, αποτελεί ένα πρόβλημα που αντιμετω-πίζεται σε ικανοποιητικό βαθμό από τη τεχνολογία των οπτικών ινών.**
* **Η ενοποίηση των τεσσάρων πεδίων δυνάμεων, του βαρυτικού, του ηλεκτρομαγνητικού, του ασθενούς πυρηνικού και του ισχυρού πυρηνικού, αποτελεί ένα πρόβλημα της σύγχρονης φυσικής που, προς το παρόν, δεν έχει επιλυθεί.**

|  |  |
| --- | --- |
| **1.2** | **Κατανόηση προβλήματος** |

**Η οποιαδήποτε προσπάθεια αντιμετώπισης ενός προβλήματος είναι καταδικασμένη σε αποτυχία αν προηγουμένως δεν έχει γίνει απόλυτα κατανοητό το πρόβλημα που τίθεται. 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.**

**12 / 16-17**

**Η μορφή με την οποία παρουσιάζεται ένα πρόβλημα μπορεί να είναι οποιαδήποτε αρκεί να μπορεί να γίνει αντιληπτή από μία από τις πέντε ανθρώπινες αισθή-σεις. Το πρόβλημα της ρύπανσης της ατμόσφαιρας της πρωτεύουσας μπορεί να το αντιληφθεί ο καθένας κοιτάζοντας τον αττικό ουρανό ή αναπνέοντας με δυσκολία ανηφορίζοντας κάποιο καλοκαιρινό μεσημέρι την οδό Ακαδημίας. Τα προβλήματα και τα δεινά που ταλαιπωρούν και σκοτώνουν χιλιάδες συνανθρώπους μας εξ αιτίας των πολεμικών συγκρούσεων στα διάφορα μέρη του κόσμου, μας γίνονται γνωστά είτε διαβάζοντας εφημερίδες, είτε ακούγοντας το ραδιό-φωνο, είτε βλέποντας τηλεόραση.**

**Τα προβλήματα που μπορεί να κληθούμε να αντιμε-τωπίσουμε κατά τη διάρκεια της ζωής μας μπορούν να αναφέρονται σε οποιοδήποτε τομέα, μπορεί να αφο-ρούν στα μαθηματικά, στη φυσική, στη λογική, στην καθημερινή ζωή ή οτιδήποτε άλλο θα μπορούσε κάποιος να σκεφτεί. Μπορεί να απαιτούνται γνώσεις συγκεκριμένων επιστημών ή μπορεί οι βιωματικές μας καταστάσεις και εμπειρίες να επαρκούν για την αντιμε-τώπισή τους. Μπορεί να είναι πολύπλοκα ή σχετικά απλά, πρωτόγνωρα ή συνηθισμένα. Σε κάθε όμως περίπτωση θα πρέπει να γίνουν απολύτως κατανοητά πριν γίνει κάθε προσπάθεια αντιμετώπισής τους.**

**Η κατανόηση ενός προβλήματος εξαρτάται σε μεγάλο βαθμό από την διατύπωσή του. Οποιοδήποτε μέσο μπορεί να χρησιμοποιηθεί για να αποδοθεί η διατύ-πωση ενός προβλήματος. Συνηθέστερο από όλα είναι ο λόγος, είτε ο προφορικός, είτε ο γραπτός.**

**13 / 17**

**Σαφήνεια διατύπωσης**

**Ο λόγος σαν μέσο επικοινωνίας και συνεννόησης πρέπει να χαρακτηρίζεται από σαφήνεια. Άστοχη χρήση ορολογίας, λανθασμένη σύνταξη, είναι δύο στοιχεία που μπορούν να προκαλέσουν παρερμηνείες και παραπλανήσεις. Η ικανότητα εκφοράς σωστού προφορικού και γραπτού λόγου αποτελεί μεγάλο προτέρημα για κάθε άτομο. Η παρερμηνεία είναι δυνατή ακόμα και σε περιπτώσεις όπου όλοι οι λεξικολογικοί και συντακτικοί κανόνες κρατούνται με ευλάβεια.**

**ΠΑΡΑΔΕΙΓΜΑ 1**

**Ένας πολυάσχολος επιχειρηματίας απευθύνεται στη σύζυγο του και της ζητά να φροντίσει για την αγορά αναμνηστικών δώρων για μερικούς παιδικούς του φίλους, που πρόκειται να συναντήσει μετά από πάρα πολλά χρόνια. Η σύζυγος του ζητάει να της δώσει κάποια χαρακτηριστικά γνωρίσματα των φίλων του, έτσι ώστε να γίνει πιο εύκολη η επιλογή των δώρων.**

**Τελικά, αυτά που πληροφορείται η σύζυγος από τον επιχειρηματία σχετικά με τους φίλους του είναι πως :**

**Ο Γιάννης και η Μαρία είναι παντρεμένοι. Ο Χρήστος είναι αθλητικός τύπος. Η Ελένη είναι προϊσταμένη σε τράπεζα.**

**Πράγματι, η σύζυγος φρόντισε και αγόρασε τα δώρα για τους φίλους του επιχειρηματία και τα έστειλε στο γρα-φείο του. Η παραλαβή των δώρων των φίλων του, έκρυβε για τον επιχειρηματία μια έκπληξη. Τα δώρα που παρέλαβε ήταν τρία, ενώ ο ίδιος περίμενε τέσσερα. Επικοινώνησε αμέσως με τη σύζυγο του, η οποία όμως τον διαβεβαίωσε, ότι έκανε τις επιλογές της ακριβώς σύμφωνα με τις πληροφορίες που της είχε ο ίδιος δώσει σχετικά με τους φίλους του. Το ένα από τα τρία δώρα προοριζόταν για το ζευγάρι των φίλων του. Στην πραγματικότητα όμως ζευγάρι φίλων δεν υπήρχε. Αμέσως έγινε κατανοητή η αιτία της παραπλάνησης.**

**14 / 17-18**

**15 / 18**

**Το πρόβλημα της αγοράς των δώρων αντιμετωπίστηκε από τη σύζυγο σύμφωνα με τις πληροφορίες που είχε πάρει. Η παραπλάνησή της οφείλεται στον τρόπο που ερμήνευσε την πρόταση Ο Γιάννης και η Μαρία είναι παντρεμένοι. Η πρόταση επιδέχεται δύο διαφορετικές, και ταυτόχρονα σωστές, ερμηνείες.**

**Το γεγονός αυτό - της αποδεκτής διπλής ερμηνείας - οφείλεται στο ότι η διατύπωση της αφήνει περιθώρια για κάτι τέτοιο.**

**Πρώτη ερμηνεία: Ο Γιάννης και η Μαρία είναι παντρε-μένοι μεταξύ τους.**

**Δεύτερη ερμηνεία : Ο Γιάννης είναι παντρεμένος και η Μαρία είναι παντρεμένη.**

**Οι δύο διαφορετικές αυτές ερμηνείες οφείλονται στον ασαφή συνδετικό ρόλο που διαδραματίζει στη συγκε-κριμένη πρόταση ο λογικός τελεστής ΚΑΙ. Δεν είναι σαφές από τη διατύπωση, αν ο τελεστής συνδέει δύο υποκείμενα μιας κύριας πρότασης (πρώτη ερμηνεία) ή αν συνδέει δύο υπονοούμενες κύριες προτάσεις (δεύτερη ερμηνεία).**

**Γίνεται λοιπόν αντιληπτό το ειδικό βάρος που έχει η σωστή διατύπωση στη σωστή κατανόηση ενός προ-βλήματος.**

**Σημαντικός ακόμα παράγοντας στη σωστή αντιμετώ-πιση ενός προβλήματος είναι η αποσαφήνιση του χώρου στον οποίο αναφέρεται. Η πληροφορία αυτή παρέχεται επίσης από την εκφώνηση του προβλή-ματος. Τα δεδομένα του προβλήματος είναι αυτά που θα μας παρέχουν αυτήν την πληροφορία.**

**ΟΡΙΣΜΟΙ**

**Με τον όρο δεδομένο δηλώνεται οποιοδήποτε στοιχείο μπορεί να γίνει αντιληπτό από έναν τουλάχιστον παρατηρητή με μία από τις πέντε αισθήσεις του.**

**Με τον όρο πληροφορία αναφέρεται οποιοδήποτε γνωσιακό στοιχείο προέρχεται από επεξεργασία δεδομένων.**

**Ο όρος επεξεργασία δεδομένων δηλώνει εκείνη τη διαδικασία κατά την οποία ένας "μηχανισμός" δέχεται δεδομένα, τα επεξεργάζεται συμφωνά με έναν προκαθορισμένο τρόπο και αποδίδει πληροφορίες.**

**Επί χιλιετίες ο "μηχανισμός" επεξεργασίας των δεδομένων ήταν και εξακολουθεί να είναι ο ανθρώ-πινος εγκέφαλος. Στις μέρες μας, ένας άλλος "μηχανισμός" επεξεργασίας δεδομένων είναι ο υπολογιστής.**

|  |  |
| --- | --- |
| **1.3** | **Δομή προβλήματος** |

**Η κατανόηση του προβλήματος είναι βασική προϋπόθεση για να γίνει στη συνέχεια δυνατή η σωστή αποτύπωση της δομής του. Η καταγραφή της δομής ενός προβλήματος σημαίνει αυτόματα ότι έχει αρχίσει η διαδικασία ανάλυσης του προβλήματος σε άλλα απλούστερα. Με τη σειρά τους τα νέα προβλήματα μπορούν να αναλυθούν σε άλλα, ακόμη πιο απλά. Η διαδικασία αυτή της ανάλυσης μπορεί να συνεχιστεί μέχρις ότου τα επιμέρους προβλήματα που προέκυψαν θεωρηθούν αρκετά απλά και η αντιμετώπισή τους χαρακτηριστεί ως δυνατή.**

**16 / 18-19**

**/ 000**

**ΟΡΙΣΜΟΣ**

**Με τον όρο δομή ενός προβλήματος αναφερόμαστε στα συστατικά του μέρη, στα επιμέρους τμήματα που το αποτελούν καθώς επίσης και στον τρόπο που αυτά τα μέρη συνδέονται μεταξύ τους.**

**Η δυσκολία αντιμετώπισης των προβλημάτων ελαττώνεται όσο περισσότερο προχωράει η ανάλυση τους σε απλούστερα προβλήματα. Ο κατακερματισμός ενός προβλήματος σε άλλα απλούστερα είναι μια από τις διαδικασίες που ενεργοποιούν και αμβλύνουν τόσο τη σκέψη, αλλά κυρίως την αναλυτική ικανότητα .του ατόμου.**

**ΠΑΡΑΔΕΙΓΜΑ 2**

**Ας υποθέσουμε ότι τίθεται σαν πρόβλημα το θέμα αντιμετώπισης των ναρκωτικών.**

**Η αντιμετώπιση του προβλήματος θα γίνει απλούστερη αν μπορέσουμε να αναλύσουμε το πρόβλημα σε άλλα απλούστερα Το αρχικό πρόβλημα είναι "Αντιμετώπιση ναρκωτικών". Αυτό θα μπορούσε να αναλυθεί καταρχήν σε τρία υποθέματα, σε τρία επιμέρους προβλήματα :**

**17 / 19-20**

**(1) Πρόληψη  
(2) Θεραπεία  
(3) Επανένταξη**

**Τα τρία αυτά επιμέρους προβλήματα πιθανό να μην είναι ιδιαίτερα λεπτομερή έτσι ώστε να επιτρέπουν την εύκολη αντιμετώπισή τους. Πρέπει λοιπόν κάθε ένα από αυτά να αναλυθεί σε ακόμα απλούστερα.**

**Έτσι λοιπόν το επιμέρους πρόβλημα (1) Πρόληψη, μπορεί να αναλυθεί σε :**

**(1.1) Σωστή ενημέρωση των πολιτών σχετικά με το θέμα  
(1.2) Υποβοήθηση προς την κατεύθυνση ανάπτυξης ενδιαφερόντων, οραμάτων και στόχων εκ μέρους των εφήβων   
(1.3) Υποστήριξη ομάδων αυξημένης θεωρητικά "προδιάθεσης"**

**18 / 20**

**Όμοια το επί μέρους πρόβλημα (2) Θεραπεία, μπορεί να αναλυθεί ως εξής :**

**(2.1) Δημιουργία νέων κρατικών θεραπευτικών κοινο-τήτων   
(2.2) Ενίσχυση υπαρχόντων θεραπευτικών κοινοτήτων   
(2.3) Δημιουργία κατάλληλων τμημάτων στα δημόσια νοσοκομεία**

**Παρόμοια το επιμέρους πρόβλημα (3) Επανένταξη, μπορεί να αναλυθεί ως ακολούθως:**

**(3.1) Καταπολέμηση της κοινωνικής προκατάληψης έναντι των απεξαρτημένων  
(3.2) Επιδότηση θέσεων εργασίας για απεξαρτημένους πρώην χρήστες**

**Στη συνέχεια και το πρόβλημα (1.1) μπορεί να αναλυθεί σε απλούστερα:**

**(1.1.1) Ενημέρωση των εφήβων μέσα από κατάλληλα προγράμματα στα σχολεία   
(1.1.2) Ενημέρωση των γονέων με προγράμματα του Δήμου  
(1.1.3) Ενημέρωση κάθε άλλου ενδιαφερόμενου πολίτη με προγράμματα του Υπουργείου Υγείας**

**Μια παρόμοια παραπέρα ανάλυση θα μπορούσε να γίνει και για το πρόβλημα (1.2), το οποίο θα μπορούσε να αναλυθεί στα εξής απλούστερα προβλήματα :**

**(1.2.1) Οργάνωση πολιτιστικών δραστηριοτήτων στα σχολεία   
(1.2.2) Δημιουργία δημόσιων χώρων άθλησης στις γειτονιές για τους νέους  
(1.2.3) Παροχή κινήτρων στα παιδιά και στους νέους για παρακολούθηση και συμμετοχή σε καλλιτεχνικά γεγονότα**

**Αν η ανάλυση του αρχικού προβλήματος θεωρείται επαρκής, η διάσπαση των επιμέρους προβλημάτων σε άλλα απλούστερα μπορεί να τερματιστεί. Ο παραπάνω τρόπος περιγραφής και ανάλυσης ενός προβλήματος γίνεται φραστικά. Ο ενδιαφερόμενος για την αντιμε-τώπιση του αρχικού προβλήματος, έχει πλέον μπροστά του να αντιμετωπίσει μια σειρά από επιμέρους προ-βλήματα, τα οποία στο σύνολο τους εκφράζουν και αντιστοιχούν στο αρχικό πρόβλημα.**

**Η ανάλυση αυτή του προβλήματος σε άλλα απλού-στερα αναδύει παράλληλα και τη δομή του προβλή-ματος.**

**19 / 20-21**

**Για τη γραφική απεικόνιση της δομής ενός προβλή-ματος χρησιμοποιείται συχνότατα η διαγραμματική αναπαράσταση. Σύμφωνα με αυτή:**

* **το αρχικό πρόβλημα αναπαρίσταται από ένα ορθογώνιο παραλληλόγραμμο**
* **κάθε ένα από τα απλούστερα προβλήματα στα ο-ποία αναλύεται ένα οποιοδήποτε πρόβλημα, αναπαρίσταται επίσης από ένα ορθογώνιο παραλληλόγραμμο**
* **τα παραλληλόγραμμα που αντιστοιχούν στα απλούστερα προβλήματα στα οποία αναλύεται ένα οποιοδήποτε πρόβλημα, σχηματίζονται ένα επίπεδο χαμηλότερα. Έτσι σε κάθε κατώτερο επίπεδο, δημιουργείται η γραφική αναπαρά-σταση των προβλημάτων στα οποία αναλύονται τα προβλήματα του αμέσως ψηλότερου επιπέ-δου.**

**Η διαγραμματική αναπαράσταση του παραδείγματος που παρατίθεται προηγούμενα φαίνεται στο σχήμα 1.1.**

**20 / 21**

**Σχ. 1.1. Διαγραμματική αναπαράσταση προβλήματος "ΑΝΤΙΜΕΤΩΠΙΣΗ ΝΑΡΚΩΤΙΚΩΝ"**

**21 / 21**

**2.1**

**2.2**

**2.3**

**1.1**

**1.2**

**1.3**

**3.1**

**3.2**

**ΑΝΤΙΜΕΤΩΠΙΣΗ**

**ΝΑΡΚΩΤΙΚΩΝ**

**Πρόληψη**

**Θεραπεία**

**Επανένταξη**

**1.1.11.1.21.1.3**

**1.2.1**

**1.2.2**

**1.2.3**

**Η διαγραμματική αναπαράσταση προσφέρει μια απτή απεικόνιση της δομής του προβλήματος. Η δημιουργία του σχετικού διαγράμματος βοηθάει τόσο στην καλύτερη κατανόηση του ίδιου του προβλήματος, όσο και στη σχεδίαση της λύσης του.**

|  |  |
| --- | --- |
| **1.4** | **Καθορισμός απαιτήσεων** |

**Η σωστή επίλυση ενός προβλήματος προϋποθέτει τον επακριβή προσδιορισμό των δεδομένων που παρέχει το πρόβλημα. Απαιτεί επίσης την λεπτομερειακή καταγραφή των ζητούμενων που αναμένονται σαν αποτελέσματα της επίλυσης του προβλήματος.**

**Θα πρέπει να δοθεί μεγάλη προσοχή στην ανίχνευση των δεδομένων ενός προβλήματος. Επισημαίνεται πως δεν είναι πάντοτε εύκολο να διακρίνει κάποιος τα δεδομένα. Υπάρχουν πολλές περιπτώσεις προβλημάτων όπου τα δεδομένα θα πρέπει να "ανακαλυφθούν" μέσα στα λεγόμενα του προβλήματος. Η διαδικασία αυτή απαιτεί προσοχή, συγκέντρωση και σκέψη. Μεθοδολογία προσδιορισμού των δεδομένων ενός προβλήματος δεν υπάρχει, ούτε και μεθοδολογία εντοπισμού και αποσαφήνισης των ζητούμενων ενός προβλήματος.**

**Το ίδιο προσεκτικά θα πρέπει να αποσαφηνιστούν και τα ζητούμενα του προβλήματος. Δεν είναι πάντοτε ιδιαίτερα κατανοητό τι ακριβώς ζητάει ένα πρόβλημα. Σε μια τέτοια περίπτωση θα πρέπει να θέτονται μια σειρά από ερωτήσεις με στόχο την διευκρίνηση πιθανών αποριών σχετικά με τα ζητούμενα, τον τρόπο παρουσίασής τους, το εύρος τους κ.λπ. Οι ερωτήσεις αυτές μπορούν να απευθύνονται είτε στο δημιουργό του προβλήματος, είτε στον ίδιο μας τον εαυτό αν εμείς καλούμαστε να αντιμετωπίσουμε το πρόβλημα.**

**22 / 21-22**

**ΠΑΡΑΔΕΙΓΜΑ 3**

**Για λόγους αξιολόγησης της εκπαιδευτικής του πολιτικής, το Υπουργείο Παιδείας χρειάζεται να ενημερωθεί για τα πρόσφατα αποτελέσματα φοί-τησης των μαθητών της χώρας. Ζήτησε λοιπόν μεταξύ άλλων, από την Υπηρεσία Πληροφορικής να παρουσιάσει και τα αποτελέσματα που είχαν οι μαθητές Γ' τάξης της Τεχνολογικής Κατεύθυνσης των Ενιαίων Λυκείων στα μαθήματα ειδικότητας.**

**Το πρόβλημα λοιπόν που τίθεται είναι *:* Αποτελέσματα επίδοσης μαθητών Γ' τάξης Τεχνολογικής Κατεύθυνσης στα μαθήματα ειδικότητας.**

**Το πρώτο πράγμα που απασχόλησε την Υπηρεσία Πληροφορικής του Υπουργείου ήταν να καταλάβει τι ακριβώς ζητείται, πράγμα που δεν φαίνεται αμέσως από το ιδιαίτερα σύντομο σημείωμα του προϊστάμενου. Έτσι προχώρησε στις εξής σκέψεις :**

**Προφανώς σε ό,τι αφορά στη χρονική περίοδο, πράγμα το οποίο δεν αναφέρεται, το Υπουργείο χρειάζεται τα αποτελέσματα της προηγούμενης σχολικής χρονιάς για όλη τη χώρα. Γνωρίζουμε σε πόσα και ποια Λύκεια λειτούργησε η Τεχνολογική Κατεύθυνση. Γνωρίζουμε επίσης ότι τα μαθήματα ειδικότητας της κατεύθυνσης αυτής είναι τέσσερα. Έτσι φαίνεται ότι ζητούνται τα αποτελέσματα φοίτησης όλων των μαθητών της Γ' τάξης των πιο πάνω Λυκείων για κάθε ένα από τα τέσσερα μαθήματα ειδικότητας.**

**23 / 22**

**Ποιά είναι όμως ακριβώς τα ζητούμενα; Σε κάθε μάθημα κάθε μαθητής λαμβάνει έναν τελικό βαθμό. Αν συγκε-ντρώσουμε όλες τις βαθμολογίες από όλα τα σχολεία και για κάθε μάθημα, τότε αυτά είναι τα ζητούμενα αποτελέσματα;**

**Σε κάθε Λύκειο όλοι οι μαθητές της Γ' Λυκείου Τεχνο-λογικής Κατεύθυνσης διδάσκονται τα ίδια μαθήματα. Έτσι ότι ζητείται για το σύνολο των σχολείων, ζητείται καταρχήν για ένα σχολείο. Μπορούμε λοιπόν να μελε-τήσουμε το πρόβλημα για ένα σχολείο και μετά να το επεκτείνουμε για όλα τα Λύκεια της χώρας.**

**24 / 22**

|  |  |  |
| --- | --- | --- |
| **Πίνακας 1.1. Απόσπασμα βαθμολογίας μαθήματος**  **25 / 23** | | |
| **ΣΧΟΛΕΙΟ : 9ο ΛΥΚΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ** | | |
| **ΤΑΞΗ :Γ ΤΜΗΜΑ:2** | | |
| **ΜΑΘΗΜΑ : Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον** | | |
| **α/α** | **ΟΝΟΜΑΤΕΠΩΝΥΜΟ** | **ΤΕΛΙΚΟΣ ΒΑΘΜΟΣ** |
| **1** | **Μαϊκούσης Αθανάσιος** | **10** |
| **2** | **Μπουρνέλη Διονυσία** | **17** |
| **3** | **Μυλωνάς Αλέξανδρος** | **15** |

**Κατόπιν αυτών η μελέτη επικεντρώνεται σε ένα σχολείο. Το σχολείο αυτό έχει 100 μαθητές στη Γ' τάξη Λυκείου Τεχνολογικής Κατεύθυνσης, κατανεμημένους σε τρία τμήματα. Οι βαθμολογίες των μαθητών και των τριών τμημάτων σε όλα τα μαθήματα επιλογής έχουν ήδη καταχωρηθεί. Ένα τμήμα της βαθμολογίας των μαθητών του πρώτου τμήματος για το μάθημα "Ανά-πτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον" φαίνεται στον πίνακα 1.1.**

**Υπάρχουν τρεις καταστάσεις βαθμολογίας, καθεμιά με τη βαθμολογία των μαθητών καθενός από τα τρία τμή-ματα. Αν τις ενώσουμε, θα έχουμε μία κατάσταση με τη βαθμολογία όλων των μαθητών της Γ' τάξης Τεχνο-λογικής Κατεύθυνσης του συγκεκριμένου σχολείου. Άρα αυτό είναι το ζητούμενο αποτέλεσμα.**

**26 / 23**

**Ωστόσο αμέσως μετά μπαίνουν τα εξής ερωτήματα: Μπορεί κανείς να βγάλει κάποιο συμπέρασμα διαβά-ζοντας μια λίστα με 100 ονόματα; Ακόμη και για να απαντηθεί το πιο απλό ερώτημα, έστω "πόσοι προά-γονται", θα πρέπει κάποιος να τους μετρήσει. Βέβαια συμφέρει να μετρηθούν αυτοί που δεν προάγονται, οι οποίοι προφανώς θα είναι λιγότεροι, αλλά πάλι δεν αποφεύγεται η σάρωση όλης της λίστας. Θα πρέπει δε να γίνει παρόμοια εργασία για όλα τα σχολεία.**

**Ένα άλλο ερώτημα που τίθεται είναι το εξής: Για τους σκοπούς της ζητούμενης μελέτης ενδιαφέρει ποιοι ακριβώς μαθητές προάγονται και ποιοι όχι, ή μόνο πόσοι προάγονται και πόσοι όχι; Είναι φανερό ότι ενδιαφέρει το δεύτερο. Επομένως το αποτέλεσμα για το εξεταζόμενο σχολείο μπορεί να είναι ότι σε αυτό και στο μάθημα "Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον" προάγονται 92 μαθητές ενώ απορρί-πτονται 8. Έχοντας κατά νου τον τελικό στόχο, δηλαδή όλη τη χώρα, θα μπορούσαμε να πάρουμε σαν τελικό αποτέλεσμα π.χ., ότι επί συνόλου 5809 μαθητών, 5287 μαθητές προάγονται και 522 απορρίπτονται στο συγκε-κριμένο μάθημα. Αντίστοιχοι αριθμοί θα υπάρχουν και για τα άλλα μαθήματα της κατεύθυνσης. Όμως μεγάλοι και πολλοί αριθμοί, αν και είναι ακριβείς, δεν παρέχουν αμέσως το ποιοτικό συμπέρασμα το οποίο είναι το ζητούμενο. Είναι προτιμότερο λοιπόν να τους εκφρά-σουμε σε ποσοστό, αφού έτσι γίνεται πιο εύκολα αντιληπτό το αποτέλεσμα.**

**Όμως το προηγούμενο αποτέλεσμα δίνει έναν πρώτο χωρισμό των μαθητών σε επιτυχόντες και αποτυ-χόντες. Δεν απαντά στο ερώτημα ποιο ποσοστό μαθητών αρίστευσε, ποιο ποσοστό μαθητών πήρε βαθμολογία που κυμάνθηκε από 10 μέχρι 13 κ.λπ. Αν έχει κάποιος αυτής της μορφής τα αποτελέσματα, μπορεί να σχηματίσει πολύ καλύτερη εικόνα για την κατάσταση. Φαίνεται λοιπόν ότι είναι προτιμότερο τα ζητούμενα αποτελέσματα να περιλαμβάνουν και ποσοστά κατά κλίμακα βαθμολογίας, όπως περίπου παρουσιάζονται στον πίνακα 1.2.**

**27 / 23-24**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Πίνακας 1.2. Αποτελέσματα στο μάθημα "Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον" σχολικού έτους 1999-2000.**  **28 / 24** | | | | |
| **Χαρακτηρισμός επίδοσης στο μάθημα** | **Βαθμός από** | **Βαθμός έως** | **Αριθμός μαθητών** | **Ποσοστά μαθητών %** |
| **Απορρίπτονται** | **0** | **9** | **522** | **9,0** |
| **Μέτρια** | **10** | **13** | **1211** | **20,8** |
| **Καλά** | **14** | **15** | **2120** | **36,5** |
| **Πολύ καλά** | **16** | **17** | **1180** | **20,3** |
| **Άριστα** | **18** | **20** | **776** | **13,4** |
| **ΣΥΝΟΛΟ** |  |  | **5809** | **100** |

**Ο πίνακας 1.2 αποκαλείται στη Στατιστική πίνακας συχνοτήτων ή κατανομή συχνοτήτων, διότι παρου-σιάζει το πλήθος των μονάδων ενός πληθυσμού που ανήκει σε ένα διάστημα τιμών (απόλυτη συχνότητα) και το ποσοστό των μονάδων του πληθυσμού που εντάσ-σεται στο ίδιο διάστημα τιμών (σχετική συχνότητα).**

**Μια άλλη σκέψη είναι ότι επειδή τα αποτελέσματα αυτά θα δημοσιοποιηθούν, θα ήταν καλύτερα αν απεικονι-ζόντουσαν γραφικά. Πράγματι με ένα γράφημα, όπως αυτό του σχήματος 1.2, είναι ο καθένας σε θέση με μια ματιά να αποκομίσει σημαντικές πληροφορίες, χωρίς να χρειάζεται την απομνημόνευση των αριθμών. Άλλωστε όπως λέει και μία παλιά κινέζικη παροιμία "μια εικόνα αξίζει όσο 1000 λέξεις".**

**29 / 24**

**Σχ. 1.2. Γράφημα ραβδογράμματος για το χαρακτηρισμό επίδοσης στο μάθημα.**

**Τέλος μπορεί ακόμη να υπολογιστεί η μέση τιμή της βαθμολογίας, καθώς και η τυπική απόκλιση που δείχνει πόσο κατά μέσο όρο απέχουν οι βαθμολογίες από τη μέση τιμή.**

**Για την ολοκλήρωση της ανάλυσης του προβλήματος, θα πρέπει να γίνουν ανάλογες ενέργειες και για τα υπόλοιπα τρία μαθήματα ειδικότητας. Με παρόμοιο τρόπο θα πρέπει να υπολογιστούν καταρχήν οι επιδόσεις των μαθητών της Γ' τάξης Λυκείου Τεχνο-λογικής Κατεύθυνσης του ενός σχολείου και στη συνέ-χεια όλων των σχολείων της χώρας, δημιουργώντας και για αυτά τα μαθήματα τους αντίστοιχους πίνακες και τα κατάλληλα γραφήματα.**

**30 / 24-25**

**Εδώ η Υπηρεσία Πληροφορικής κρίνει ότι έχει ολοκλη-ρώσει την ανάλυση του προβλήματος, που της ζητή-θηκε να λύσει. Θεωρείται ότι το πρόβλημα έχει κατανο-ηθεί πλήρως. Είναι γνωστό πλέον το περιβάλλον του προβλήματος, πρώτα στο επίπεδο σχολείου και κατόπιν στο επίπεδο όλης της χώρας, ενώ η πηγή των δεδομένων είναι οι καταστάσεις βαθμολογιών, οι οποίες υπάρχουν. Τα ζητούμενα αποτελέσματα έχουν αποσαφηνιστεί, και η μορφή που παρέχονται είναι τόσο αριθμητική (απόλυτη και ποσοστιαία), όσο και διαγραμματική.**

**Συμπερασματικά από όλα τα παραπάνω διαφαίνεται πως τα στάδια αντιμετώπισης ενός προβλήματος είναι τρία (σχήμα 1.3):**

* **κατανόηση, όπου απαιτείται η σωστή και πλήρης αποσαφήνιση των δεδομένων και των ζητού-μενων του προβλήματος**
* **ανάλυση, όπου το αρχικό πρόβλημα διασπάται σε άλλα επί μέρους απλούστερα προβλήματα**
* **επίλυση, όπου υλοποιείται η λύση του προβλή-ματος, μέσω της λύσης των επιμέρους προβλη-μάτων.**

**31 / 25**

**ΚΑΤΑΝΟΗΣΗ**

**ΑΝΑΛΥΣΗ**

**ΕΠΙΛΥΣΗ**

**Σχ. 1.3. Στάδια αντιμετώπισης προβλημάτων**

|  |  |
| --- | --- |
| **1.5** | **Κατηγορίες προβλημάτων** |

**Τα προβλήματα που απαντώνται τόσο στους διάφο-ρους επιστημονικούς τομείς, όσο και στην καθημερινή μας ζωή, ποικίλουν ως προς τη φύση τους. Από τα παραδείγματα που έχουμε παραθέσει, έχει γίνει αντιλη-πτό πως τα προβλήματα δεν σχετίζονται υποχρεωτικά και αποκλειστικά με τα μαθηματικά ή γενικότερα με μαθηματικές και υπολογιστικές διαδικασίες με σκοπό την επίτευξη λύσης τους. Η διαφορετική φύση των προβλημάτων επιτρέπει την κατηγοριοποίησή τους σύμφωνα με ποικίλα κριτήρια.**

1. **Με κριτήριο τη δυνατότητα επίλυσης ενός προβλή-ματος, διακρίνουμε τρεις κατηγορίες προβλημά-  
   των :**

**32 / 25**

1. **Επιλύσιμα, είναι εκείνα τα προβλήματα για τα οποί-α η λύση τους είναι ήδη γνωστή και έχει διατυ-πωθεί. Επιλύσιμα μπορεί επίσης να χαρακτη-ριστούν και προβλήματα, των οποίων η λύση δεν έχει ακόμα διατυπωθεί, αλλά η συνάφειά τους με άλλα ήδη επιλυμένα μας επιτρέπει να θεωρούμε σαν βέβαιη τη δυνατότητα επίλυσής τους.**
2. **Ανοικτά, ονομάζονται εκείνα τα προβλήματα για τα οποία η λύση τους δεν έχει μεν ακόμα βρεθεί, αλλά παράλληλα δεν έχει αποδειχθεί, ότι δεν επιδέχονται λύση. Σαν παράδειγμα ανοικτού προβλήματος μπορούμε να αναφέρουμε το πρόβλημα της ενο-ποίησης των τεσσάρων πεδίων δυνάμεων, που αναφέρουμε σε προηγούμενη παράγραφο.**

**33 / 25-26**

1. **Άλυτα, χαρακτηρίζονται εκείνα τα προβλήματα για τα οποία έχουμε φτάσει στην παραδοχή, ότι δεν επιδέχονται λύση. Τέτοιου είδους πρόβλημα είναι το γνωστό από τους αρχαίους ελληνικούς χρόνους πρόβλημα του τετραγωνισμού του κύκλου. Το πρό-βλημα αυτό θεωρείται άλυτο, στην πραγματικότητα η λύση που επιδέχεται είναι προσεγγιστική.**
2. **Με κριτήριο το βαθμό δόμησης των λύσεών τους, τα επιλύσιμα προβλήματα μπορούν να διακριθούν σε τρεις επίσης κατηγορίες :**
3. **Δομημένα, χαρακτηρίζονται εκείνα τα προβλήματα των οποίων η επίλυση προέρχεται από μια αυτό-ματοποιημένη διαδικασία. Για παράδειγμα, η επίλυση της δευτεροβάθμιας εξίσωσης αποτελεί ένα δομημένο πρόβλημα, αφού ο τρόπος επίλυσης της εξίσωσης είναι γνωστός και αυτοματοποιημέ-νος.**
4. **Ημιδομημένα, ονομάζονται τα προβλήματα εκείνα των οποίων η λύση επιδιώκεται στα πλαίσια ενός εύρους πιθανών λύσεων, αφήνοντας στον ανθρώπινο παράγοντα περιθώρια επιλογής της. Σαν παράδειγμα ημιδομημένου προβλήματος μπορούμε να αναφέρουμε ένα πρόβλημα όπου ένας ταξιδιώτης αναζητά να επιλέξει το μεταφορικό μέσο μετακίνησής του από ένα μέρος σε κάποιο άλλο. Το πρόβλημα είναι ημιδομημένο, δεδομένου ότι η λύση που θα επιλεγεί, πρέπει να αναζητηθεί σε ένα σύνολο σαφώς προκαθορισμένο που συμπεριλαμβάνει όλα τα διαθέσιμα μεταφορικά μέσα.**

**34 / 26**

1. **Αδόμητα, χαρακτηρίζονται τα προβλήματα εκείνα των οποίων οι λύσεις δεν μπορούν να δομηθούν ή δεν έχει διερευνηθεί σε βάθος η δυνατότητα δόμησής τους. Πρωτεύοντα ρόλο στην επίλυση αυτού του τύπου προβλημάτων κατέχει η ανθρώ-πινη διαίσθηση. Παράδειγμα αδόμητου προβλή-ματος είναι η επιλογή του τρόπου, του τόπου και του χρόνου ενός εφηβικού πάρτυ. Είναι σαφές ότι δεν υπάρχει κανένας προδιατυπωμένος τρόπος οργάνωσης ενός εφηβικού πάρτυ και όλοι οι παράγοντες που θα το διαμορφώσουν επαφίονται στην ανθρώπινη αίσθηση και προτίμηση των διοργανωτών του.**
2. **Το κάθε πρόβλημα σε ότι αφορά στην επίλυσή του, είναι στενά συνδεδεμένο με την** **έννοια του αλγό-ριθμου που παρουσιάζουμε αναλυτικά στο επόμενο κεφάλαιο. Με κριτήριο το είδος της επίλυσης που επιζητούν, τα προβλήματα διακρίνονται σε τρεις κατηγορίες :**
3. **Απόφασης, όπου η απόφαση που πρόκειται να ληφθεί σαν λύση του προβλήματος που τίθεται, απαντά σε ένα ερώτημα και πιθανόν αυτή η απά-ντηση να είναι ένα "Ναι" ή ένα "Όχι". Αυτό που θέλουμε να διαπιστώσουμε σε ένα πρόβλημα απόφασης είναι αν υπάρχει απάντηση που ικανο-ποιεί τα δεδομένα που θέτονται από το πρόβλημα.**

|  |  |
| --- | --- |
|  | **Με μια απλοποιημένη προσέγγιση, αλγόριθμος είναι μια "συνταγή" που προσδιορίζει τι πρέπει να κάνουμε κάτω από ορισμένες συνθήκες, έτσι ώστε να φτάσουμε στον επιθυμητό σκοπό.** |

**ΠΑΡΑΔΕΙΓΜΑ:**

**Δίδεται ένας ακέραιος αριθμός Ν και το πρόβλημα που τίθεται είναι, αν ο αριθμός Ν είναι πρώτος.**

1. **Υπολογιστικά, όπου το πρόβλημα που τίθεται απαιτεί τη διενέργεια υπολογισμών, για να μπορεί να δοθεί μία απάντηση στο πρόβλημα. Σε ένα υπολογιστικό πρόβλημα ζητάμε να βρούμε την τιμή της απάντησης που ικανοποιεί τα δεδομένα που παρέχει το πρόβλημα.**

**ΠΑΡΑΔΕΙΓΜΑ:**

**Δίδεται ένας ακέραιος αριθμός Ν και ζητείται να βρεθεί πόσες διαφορετικές παραγοντοποιήσεις του Ν υπάρχουν.**

**35 / 26-27**

1. **Βελτιστοποίησης, όπου το πρόβλημα που τίθεται επιζητά το βέλτιστο αποτέλεσμα για τα συγκε-κριμένα δεδομένα που διαθέτει. Σε ένα πρόβλημα βελτιστοποίησης αναζητούμε την απάντηση που ικανοποιεί κατά τον καλύτερο τρόπο τα δεδομένα που παρέχει το πρόβλημα.**

**ΠΑΡΑΔΕΙΓΜΑ:**

**Δίδεται ένας ακέραιος αριθμός Ν και ζητείται ποια είναι η παραγοντοποίηση για το Ν με το μεγαλύτερο πλήθος παραγόντων.**

|  |  |
| --- | --- |
| **1.6** | **Πρόβλημα και υπολογιστής** |

**Τα προβλήματα που καθημερινά συναντάμε δεν είναι προβλήματα του χώρου των μαθηματικών, της φυσικής ή της στατιστικής. Συνήθως είναι λιγότερο ή περισ-σότερο σύνθετα προβλήματα που η αντιμετώπισή τους γίνεται αποκλειστικά και μόνο με βάση τους συλλο-γισμούς μας και τη σκέψη μας. Σε κάποιες ίσως περι-πτώσεις χρειάζεται να πιάσουμε μολύβι και χαρτί, ή έστω και κάποιο κομπιουτεράκι, για να κάνουμε μερι-κούς αναγκαίους υπολογισμούς. Όμως οι όποιες προσπάθειες αντιμετώπισης προβλημάτων στηρίζονται στις νοητικές μας δυνάμεις, στη συλλογιστική μας και στη ικανότητα λήψης αποφάσεων.**

**Και οι υπολογιστές που έχουν κυριολεκτικά εισβάλλει στη ζωή μας, που καθημερινά ερχόμαστε σε επαφή μαζί τους, που αναντίρρητα εφαρμόζονται σε ένα μεγάλο εύρος τομέων και δραστηριοτήτων, σχετίζονται με την επίλυση προβλημάτων, και αν ναι με ποιο τρόπο;**

**36 / 27**

**Αναμφίβολα προβλήματα λυνόντουσαν και πριν τη "γένεση" των υπολογιστών. Η ικανότητα του ανθρώ-που να αντιμετωπίζει και να επιλύει προβλήματα είναι πολύ προγενέστερη της εμφάνισής τους. Οι υπολο-γιστές ήρθαν σχετικά πάρα πολύ πρόσφατα για να δράσουν επικουρικά στην ανθρώπινη δραστηριότητα.**

**Η ένα προς ένα αντιπαράθεση στοιχείων μεταξύ ανθρώπου και υπολογιστή είναι ένα ενδιαφέρον "παιγνίδι". Η Πληροφορική δεν έχει να κάνει πλέον μόνο με τους υπολογιστές, αλλά με τον τρόπο ζωής του ανθρώπου. Το ανθρώπινο είδος, νοήμον ον, διαθέτει έμφυτες όλες εκείνες τις ικανότητες που απαιτούνται για την επιτυχή αντιμετώπιση μιας τεράστιας γκάμας προβλημάτων. Η πολυπλοκότητα των συλλογισμών που μπορεί να εκτελεί ο ανθρώπινος εγκέφαλος είναι εξαιρετικά μεγάλη. Ο ακριβής τρόπος λειτουργίας του ανθρώπινου εγκεφάλου εξακολουθεί να παραμένει ένα αναπάντητο, τουλάχιστον προς το παρόν, ερώτημα.**

**37 / 27-28**

**Η "σύγκριση" λειτουργιών ανθρώπου και υπολογιστή επιφέρει βέβαια μια τεράστια ποιοτική διαφορά υπέρ του ανθρώπου. Ο υπολογιστής δεν είναι ένας ηλεκτρο-νικός εγκέφαλος. Αυτό που κάνει δεν είναι τίποτε περισσότερο από το να χειρίζεται στοιχεία, ενώ το ανθρώπινο πνεύμα μπορεί να σκέπτεται, να παράγει ιδέες. Το σημείο αυτό είναι πρωταρχικής σημασίας, προσδιορίζοντας μια αναμφισβήτητα τεράστια ποιοτική διαφορά. Το σημείο εκείνο στο οποίο ο υπολογιστής υπερτερεί έναντι του ανθρώπου, είναι η ταχύτητα εκτέλεσης των πράξεών του, ταχύτητα η οποία βελτι-ώνεται συνέχεια κατά αλματώδη τρόπο με την πρόοδο της τεχνολογίας.**

**Προβλήματα τα οποία απαιτούν πολλούς υπολογι-σμούς για την αντιμετώπιση τους, ενδείκνυνται για ανάθεση προς επίλυση σε υπολογιστή. Από τα παρα-δείγματα που αναφέρθηκαν στο κεφάλαιο, αυτό το πα-ράδειγμα του οποίου η επίλυση θα μπορούσε να ανα-τεθεί σε υπολογιστή είναι το παράδειγμα 3 σχετικά με τα αποτελέσματα φοίτησης των μαθητών.**

**Οι λόγοι που αναθέτουμε την επίλυση ενός προβλή-ματος σε υπολογιστή σχετίζονται με**

* **την πολυπλοκότητα των υπολογισμών,**
* **την επαναληπτικότητα των διαδικασιών,**
* **την ταχύτητα εκτέλεσης των πράξεων,**

**38 / 28**

* **το μεγάλο πλήθος των δεδομένων.**

**Όσο και αν τυχόν ξαφνιάζει, ο υπολογιστής δεν μπορεί να εκτελεί παρά μόνο τρεις λειτουργίες :**

1. **πρόσθεση, η οποία αποτελεί τη βασική αριθμητική πράξη, δεδομένου ότι και οι άλλες αριθμητικές πράξεις μπορούν να αντιμετωπιστούν, σαν διαδι-κασίες πρόσθεσης**
2. **σύγκριση, η οποία συνιστά τη βασική λειτουργία για την επιτέλεση όλων των λογικών πράξεων,**
3. **μεταφορά δεδομένων, λειτουργία που προηγείται και έπεται της επεξεργασίας δεδομένων.**

**Οι λειτουργίες αυτές είναι αρκετές, ώστε ο υπολογιστής να επιτελέσει με επιτυχία κάθε είδους επεξεργασία. Με βάση αυτές τις τρεις λειτουργίες διεκπεραιώνει όλες τις εργασίες που του αναθέτονται και επιλύει όλα τα προ-βλήματα που αναλαμβάνει.**

**Είναι σαφές ότι ο άνθρωπος θα χρειαζόταν χρόνια ή και αιώνες για να εκτελέσει τις πράξεις εκείνες που ο υπο-λογιστής μπορεί να τις ολοκληρώσει μέσα σε λίγα μόλις λεπτά. Είναι λοιπόν ιδιαίτερα χρήσιμη και σημαντική η προσφορά του. Όμως δεν θα πρέπει να ξεχνάει κανείς πως η ικανότητα που παρουσιάζει ο υπολογιστής εκδηλώνεται σε ποσοτικό επίπεδο και όχι σε ποιοτικό. Μπορεί να αντιμετωπίσει σύνθετα λογικά προβλήματα μόνο εφόσον ο άνθρωπος έχει φροντίσει προηγούμενα, με τη χρήση κατάλληλων προγραμμάτων, να του "διδά-ξει" τον τρόπο αντιμετώπισης και επίλυσης αυτού του είδους των προβλημάτων. Πρέπει να γίνει απόλυτα κατανοητό πως τα προβλήματα και οι λύσεις του προ-ϋπήρξαν και εξακολουθούν να υπάρχουν ανεξάρτητα από τους υπολογιστές.**

**39 / 28-29**

**• Ανακεφαλαίωση**

**Στο κεφάλαιο αυτό παρουσιάστηκε η διαχρονικότητα του προβλήματος και έγινε σαφής η ανεξαρτησία της λύσης του από τον υπολογιστή. Επισημάνθηκαν βασι-κά στοιχεία στην ανάλυση και σύνταξη προβλημάτων που αφορούν στη σαφήνεια της διατύπωσης και κατ' επέκταση στην κατανόηση του προβλήματος. Στη συνέχεια παρουσιάστηκε η έννοια της δομής ενός προβλήματος, που ουσιαστικά ανάγει την ανάλυσή του σε άλλα απλούστερα. Ο καθορισμός δεδομένων και ζητούμενων ήταν αυτό που μας απασχόλησε στη συνέχεια. Ολοκληρώνοντας το πρώτο αυτό κεφάλαιο παρουσιάσαμε διαφορετικές κατηγοριοποιήσεις των προβλημάτων. Τέλος προσδιορίσαμε τους λόγους που συνηγορούν υπέρ της ανάθεσης προβλήματος σε υπολογιστή, εντοπίζοντας ταυτόχρονα τις ποιοτικές διαφορές αντιμετώπισης προβλημάτων ανάμεσα στον υπολογιστή και τον άνθρωπο.**

**Λέξεις κλειδιά**

**Δεδομένο, διαγραμματική αναπαράσταση, δομή προβλήματος, επεξεργασία δεδομένων, κατηγορίες προβλημάτων, πληροφορία, πρόβλημα.**

**40 / 29**

**• Ερωτήσεις Θέματα για συζήτηση**

1. **Να γίνει συζήτηση σχετικά με την αντιμετώπιση προβλημάτων με τη χρήση υπολογιστών.**
2. **Να γίνει συζήτηση σχετικά με τη δημιουργία προβλημάτων εξ αιτίας της χρήσης υπολογιστών.**
3. **Οι μαθητές να αναφέρουν ιεραρχώντας τα σημαντικότερα κοινωνικά προβλήματα που κατά τη γνώμη τους σχετίζονται με την επιστήμη της Πληροφορικής.**
4. **Να δοθεί ο ορισμός των όρων δεδομένο, επεξερ-γασία δεδομένων, πληροφορία.**
5. **Να γίνει η γραφική αναπαράσταση του προβλή-ματος που περιγράφεται στο παράδειγμα της ενότητας 1.4.**
6. **Να αναφερθούν οι κατηγορίες των προβλημάτων.**
7. **Για ποιους λόγους αναθέτεται η επίλυση ενός προβλήματος σε υπολογιστή;**
8. **Περιγράψτε τους τρόπους περιγραφής και αναπαράστασης των προβλημάτων.**

**41 / 29**

**• Βιβλιογραφία**

1. **Jacques Arsac, Les machines a penserDes ordinateurs et des hommes, Seuil, Paris, 1987.**
2. **Emanuel Falkenauer, Genetic algorithms and grouping problems, Wiley, 1998.**
3. **Les Goldschlager & Andrew Lister: Computer Science A modern introduction, Prentice Hall,   
   1990.**
4. **R. Kadesch, Problem Solving-Across the Disciplines, Prentice Hall Engineering, Science & Math, 1996.**
5. **Nicholas Negroponte, Being digital, Alfred Knopf Inc, 1995.**
6. **G. Polya, How to solve it A new aspect of mathematical method, Princeton University Press(2nd edition renewed),1985.**

**42 / 30**

**Διαδίκτυο**

[**http://www.awesomelibrary.org/Classroom/Science/Problem\_Solving/Problem\_Solving.html**](http://www.awesomelibrary.org/Classroom/Science/Problem_Solving/Problem_Solving.html)

**Ιστοσελίδα με αρκετά ενδιαφέρουσες απόψεις σε ότι αφορά στην επίλυση προβλημάτων. Αρκετά υψηλού επιπέδου θεματολογίας και σχετικής δυσκολίας, προτείνεται για μαθητές με αυξημένο ενδιαφέρον.**

[**http://www.hawaii.edu/suremath/click.html**](http://www.hawaii.edu/suremath/click.html)

**Ιστοσελίδα με ιδιαίτερα χαρούμενο και νεανικό περιβάλλον, αναφέρεται στην επίλυση προβλήματος και περιλαμβάνει, εκτός από τη θεωρία περί επίλυ-σης προβλήματος, μια σειρά από προβλήματα που προέρχονται από διαφορετικά σχολικά μαθήματα (άλγεβρα, φυσική, χημεία) και άλλους τομείς δραστη-ριοτήτων.**

**43 / 30**

**Κεφάλαιο 2**

**Βασικές Έννοιες Αλγορίθμων**

**ΠΕΡΙΕΧΟΜΕΝΑ**

|  |  |
| --- | --- |
|  | **Τι είναι αλγόριθμος** |
|  | **Σπουδαιότητα αλγορίθμων** |
|  | **Περιγραφή και αναπαράσταση αλγορίθμων** |
|  | **Βασικές συνιστώσες /εντολές ενός αλγορίθμου** |

**44 / 31**

**• Εισαγωγή**

**Αρχικά εξηγείται ο όρος αλγόριθμος και παραθέτονται τα σπουδαιότερα κριτήρια που πρέπει να πληροί κάθε αλγόριθμος. Στη συνέχεια, η σπουδαιότητα των αλγορίθμων συνδυάζεται με την εξέλιξη της επιστήμης της Πληροφορικής. Η περιγραφή και αναπαράσταση των αλγορίθμων δίνεται αναλυτικά με χρήση των μεθόδων αναπαράστασης ελεύθερου κειμένου, δια-γραμμάτων ροής, φυσικής γλώσσας και κωδικοποί-ησης με πρόγραμμα. Τα προγράμματα παρουσιά-ζονται με τη μορφή ψευδογλώσσας, που ορίζεται και τυποποιείται σε ένα σύνολο εντολών και προγραμ-ματιστικών ακολουθιακών ενοτήτων. Στη συνέχεια, δίνονται παραδείγματα όπου εξετάζονται οι διάφορες συνιστώσες ενός αλγόριθμου, δηλαδή οι απαραίτητες εντολές που στηρίζουν το 'κτίσιμο' ενός αλγόριθμου. Συγκεκριμένα, παρουσιάζονται η δομή ακολουθίας, η δομή της επιλογής, οι επαναληπτικές διαδικασίες, οι διαδικασίες πολλαπλών επιλογών και οι εμφωλι-ασμένες διαδικασίες. Για κάθε τύπο συνιστώσας δίνονται αναλυτικά παραδείγματα σε φυσική γλώσσα, σε ακολουθία διαδοχικών βημάτων και σε μορφή διαγραμμάτων ροής. Στο τέλος του κεφαλαίου παρου-σιάζεται η ανάπτυξη και η αλγοριθμική προσέγγιση για την επίλυση ενός συνθετότερου προβλήματος, του προβλήματος του πολλαπλασιασμού αλά ρωσικά', όπου γίνεται χρήση και συνδυασμός αλγοριθμικών συνιστωσών.**

**45 / 32**

**• Διδακτικοί στόχοι**

**Στόχοι του κεφαλαίου αυτού είναι οι μαθητές:**

* **να διατυπώνουν την έννοια του αλγορίθμου**,
* **να αιτιολογούν τη σπουδαιότητα των αλγορίθμων,**
* **να τεκμηριώνουν την αναγκαιότητα της αλγο-ριθμικής προσέγγισης κατά τη διαδικασία επίλυσης προβλημάτων,**
* **να εφαρμόζουν τυποποιημένη επίλυση με αλγοριθμικές διαδικασίες**,
* **να μπορούν να σχεδιάζουν αλγόριθμους με χρήση συγκεκριμένων τεχνικών.**

**• Προερωτήσεις**

* **Γνωρίζεις τι είναι αλγοριθμική προσέγγιση;**
* **Ξέρεις ότι ήδη έχεις χρησιμοποιήσει πολλούς αλγορίθμους;**
* **Γνωρίζεις, αν ο πολλαπλασιασμός δύο αριθμών μπορεί να γίνει με άλλο τρόπο;**
* **Τι θα κάνεις για να βρεις το άθροισμα 3+6+9+...+ 999;**

**46 / 32**

|  |  |
| --- | --- |
| **2.1** | **Τι είναι αλγόριθμος** |

**Η θεωρία των αλγορίθμων έχει μεγάλη παράδοση και η ηλικία μερικών αλγορίθμων αριθμεί χιλιάδες χρόνια, όπως για παράδειγμα ο αλγόριθμος του Ευκλείδη για την εύρεση του μέγιστου κοινού διαιρέτη δύο αριθμών ή το λεγόμενο κόσκινο του Ερατοσθένη για την εύρεση των πρώτων αριθμών από 1 ως n. Σήμερα το πεδίο της Θεωρίας Αλγορίθμων είναι ένα ιδιαίτερα ευρύ και πλού-σιο πεδίο. Πληθώρα συγγραμμάτων έχει εμφανισθεί στη βιβλιογραφία, ενώ συνεχίζεται η περαιτέρω εμβά-**

**47 / 33**

**θυνση σε νέα σύγχρονα προβλήματα. Οι περισσότεροι από τους αλγορίθμους που συνήθως εξετάζονται στα σχετικά βιβλία έχουν προταθεί τα τελευταία 25 χρόνια, όση περίπου είναι και η ηλικία της Πληροφορικής ως μίας νέας αυθύπαρκτης επιστήμης.**

**Ιστορικό σημείωμα**

**Η λέξη αλγόριθμος (algorithm) προέρχεται από μια μελέτη του Πέρση μαθηματικού Abu Ja'far Mohammed ibn Musa al Khowarizmi, που έζησε περί το 825 μ.Χ. Πέντε αιώνες αργότερα η μελέτη αυτή μεταφράστηκε στα λατινικά και άρχιζε με τη φράση "Algoritmi dixit..." (ο αλγόριθμος λέει ....). Η μελέτη του al Khowarizmi υπήρξε η πρώτη πλήρης πραγματεία άλγεβρας (όρος που και αυτός προέρχεται από το αραβικό al-jabr=αποκατάσταση), γιατί ένας από τους σκοπούς της άλγεβρας είναι και η αποκατάσταση της ισότητας μέσα σε μια εξίσωση. Ο όρος αλγόριθμος επέζησε επί χίλια χρόνια ως σπάνιος όρος, που σήμαινε κάτι σαν "συστηματική διαδικασία αριθμη-**

**τικών χειρισμών". Τη σημερινή του αξία απόκτησε από την αρχή του 20ού αιώνα με την ανάπτυξη της ομώνυμης θεωρίας και φυσικά με την επικαιρότητα των ηλεκτρονικών υπολογιστών.**

**Ο όρος αλγόριθμος, λοιπόν, χρησιμοποιείται για να δηλώσει μεθόδους που εφαρμόζονται για την επίλυση προβλημάτων. Ωστόσο, ένας πιο αυστηρός ορισμός της έννοιας αυτής είναι ο εξής.**

**ΟΡΙΣΜΟΣ**

**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 (input). Καμία, μία ή περισσότερες τιμές δεδομένων πρέπει να δίνονται ως είσοδοι στον αλγόριθμο. Η περίπτωση που δεν δίνονται τιμές δεδομένων εμφανίζεται, όταν ο αλγόριθμος δημιουργεί και επεξεργάζεται κάποιες πρωτογενείς τιμές με τη βοήθεια συναρτήσεων παραγωγής τυχαίων αριθμών ή με τη βοήθεια άλλων απλών εντολών.**

**48 / 33**

1. **Έξοδος (output). Ο αλγόριθμος πρέπει να δημιουργεί τουλάχιστον μία τιμή δεδομένων ως αποτέλεσμα προς το χρήστη ή προς έναν άλλο αλγόριθμο.**
2. **Καθοριστικότητα (definiteness). Κάθε εντολή πρέπει να καθορίζεται χωρίς καμία αμφιβολία για τον τρόπο εκτέλεσής της. Λόγου χάριν, μία εντολή διαίρεσης πρέπει να θεωρεί και την περίπτωση, όπου ο διαιρέτης λαμβάνει μηδενική τιμή.**
3. **Περατότητα (finiteness). Ο αλγόριθμος να τελει-ώνει μετά από πεπερασμένα βήματα εκτέλεσης των εντολών του. Μία διαδικασία που δεν τελειώνει μετά από ένα συγκεκριμένο αριθμό βημάτων δεν αποτελεί αλγόριθμο, αλλά λέγεται απλά υπολο-γιστική διαδικασία (computational procedure).**
4. **Αποτελεσματικότητα (effectiveness). Κάθε μεμονωμένη εντολή του αλγορίθμου να είναι απλή. Αυτό σημαίνει ότι μία εντολή δεν αρκεί να έχει ορισθεί, αλλά πρέπει να είναι και εκτελέσιμη.**

**Η έννοια του αλγόριθμου δεν συνδέεται αποκλειστικά και μόνο με προβλήματα της Πληροφορικής. Ας θεωρή-σουμε, για παράδειγμα, ότι θέλουμε να γευματίσουμε και επομένως πρέπει να εκτελέσουμε τις επόμενες ενέργειες:**

* **να συγκεντρώσουμε τα υλικά,**
* **να προετοιμάσουμε τα σκεύη μαγειρικής,**
* **να παρασκευάσουμε το φαγητό,**
* **να ετοιμάσουμε τη σαλάτα,**
* **να στρώσουμε το τραπέζι,**
* **να γευματίσουμε,**
* **να καθαρίσουμε το τραπέζι, και**
* **να πλύνουμε τα πιάτα και τα κουζινικά.**

**Είναι ευνόητο ότι η προηγούμενη αλληλουχία των ενεργειών οδηγεί στο επιθυμητό αποτέλεσμα. Βέβαια, αυτή η αλληλουχία δεν είναι η μοναδική για την επί-τευξη του σκοπού, αφού, για παράδειγμα, μπορούμε πρώτα να ετοιμάσουμε τη σαλάτα και μετά να παρα-σκευάσουμε το φαγητό, ενώ ακόμη μπορούμε πρώτα να πλύνουμε τα πιάτα και μετά να καθαρίσουμε το τραπέζι. Ωστόσο, το παράδειγμα θέλει να δείξει, ότι η θεώρηση μίας σύνθετης εργασίας με διακριτά βήματα που εκτελούνται διαδοχικά, είναι ένας πολύ χρήσιμος και πρακτικός τρόπος σκέψης για την επίλυση πολλών (αν όχι όλων) προβλημάτων.**

**49 / 33-34**

|  |  |
| --- | --- |
| **2.2** | **Σπουδαιότητα αλγορίθμων** |

**Η έννοια του αλγόριθμου είναι θεμελιώδης για την επιστήμη της Πληροφορικής. Η μελέτη των αλγορίθμων είναι πολύ ενδιαφέρουσα, γιατί είναι η πρώτη ύλη για τη μελέτη και εμβάθυνση, αν όχι σε όλες, τουλάχιστον σε πάρα πολλές γνωστικές περιοχές της επιστήμης αυτής.**

**Η Πληροφορική, λοιπόν, μπορεί να ορισθεί ως η επιστήμη που μελετά τους αλγορίθμους από τις ακόλουθες σκοπιές:**

1. **Υλικού (hardware). Η ταχύτητα εκτέλεσης ενός αλγορίθμου επηρεάζεται από τις διάφορες τεχνολογίες υλικού, δηλαδή από τον τρόπο που είναι δομημένα σε μία ενιαία αρχιτεκτονική τα διάφορα συστατικά του υπολογιστή (δηλαδή ανάλογα με το αν ο υπολογιστής έχει κρυφή μνήμη και πόση, ανάλογα με την ταχύτητα της κύριας και δευτερεύουσας μνήμης κοκ.).**
2. **Γλωσσών Προγραμματισμού (programming languages). Το είδος της γλώσσας προγραμ-ματισμού που χρησιμοποιείται (δηλαδή, χαμηλό-τερου ή υψηλότερου επιπέδου) αλλάζει τη δομή και τον αριθμό των εντολών ενός αλγορίθμου. Γενικά μία γλώσσα που είναι χαμηλότερου επιπέδου (όπως η assembly ή η γλώσσα C) είναι ταχύτερη από μία άλλη γλώσσα που είναι υψηλο-τέρου επιπέδου (όπως η Basic ή Pascal). Ακόμη, σημειώνεται ότι διαφορές συναντώνται μεταξύ των γλωσσών σε σχέση με το πότε εμφανίσθη-καν. Για παράδειγμα, παλαιότερα μερικές γλώσ-σες προγραμματισμού δεν υποστήριζαν την αναδρομή (έννοια που θα εξετάσουμε σε βάθος αργότερα).**

**50 / 34**

1. **Θεωρητική (theoretical). Το ερώτημα που συχνά τίθεται είναι, αν πράγματι υπάρχει ή όχι κάποιος αποδοτικός αλγόριθμος για την επίλυση ενός προβλήματος. Η εξέταση αυτού του ερωτήματος είναι δύσκολο να σχολιασθεί στα πλαίσια του βιβλίου αυτού, επειδή απαιτεί μεγάλη θεωρητική κατάρτιση. Ωστόσο η προσέγγιση αυτή είναι ιδιαίτερα σημαντική, γιατί προσδιορίζει τα όρια της λύσης που θα βρεθεί σε σχέση με ένα συγκε-κριμένο πρόβλημα.**
2. **Αναλυτική (analytical). Μελετώνται οι υπολογι-στικοί πόροι (computer resources) που απαιτού-νται από έναν αλγόριθμο, όπως για παράδειγμα το μέγεθος της κύριας και της δευτερεύουσας μνήμης, ο χρόνος για λειτουργίες CPU και για λειτουργίες εισόδου/εξόδου κ.λπ. Το αντικείμενο αυτό θα εξηγηθεί πληρέστερα στο Κεφάλαιο 5.**

**51 / 34-35**

|  |  |
| --- | --- |
| **2.3** | **Περιγραφή και αναπαράσταση αλγορίθμων** |

**Στη βιβλιογραφία συναντώνται διάφοροι τρόποι αναπαράστασης ενός αλγορίθμου:**

* **με ελεύθερο κείμενο (free text), που αποτελεί τον πιο ανεπεξέργαστο και αδόμητο τρόπο παρου-σίασης αλγορίθμου. Έτσι εγκυμονεί τον κίνδυνο ότι μπορεί εύκολα να οδηγήσει σε μη εκτελέσιμη πα-ρουσίαση παραβιάζοντας το τελευταίο χαρακτη-ριστικό των αλγορίθμων, δηλαδή την αποτελε-σματικότητα.**
* **με διαγραμματικές τεχνικές, (diagramming techniques), που συνιστούν ένα γραφικό τρόπο παρουσίασης του αλγορίθμου. Από τις διάφορες διαγραμματικές τεχνικές που έχουν επινοηθεί, η πιο παλιά και η πιο γνωστή ίσως, είναι το διάγραμμα ροής (flow chart). Ωστόσο η χρήση διαγραμμάτων ροής για την παρουσίαση αλγορίθμων δεν αποτε-λεί την καλύτερη λύση, γι' αυτό και εμφανίζονται όλο και σπανιότερα στη βιβλιογραφία και στην πράξη.**
* **με φυσική γλώσσα (natural language) κατά βήμα-τα. Στην περίπτωση αυτή χρειάζεται προσοχή, γιατί μπορεί να παραβιασθεί το τρίτο βασικό χαρακτη-ριστικό ενός αλγορίθμου, όπως προσδιορίσθηκε προηγουμένως, δηλαδή το κριτήριο του καθορι-σμού.**

**52/ 35**

* **με κωδικοποίηση (coding), δηλαδή με ένα πρό-γραμμα γραμμένο είτε σε μία ψευδογλώσσα είτε σε κάποια γλώσσα προγραμματισμού που όταν εκτελεσθεί θα δώσει τα ίδια αποτελέσματα με τον αλγόριθμο.**

**Όλοι οι αλγόριθμοι του βιβλίου αυτού είναι κωδικοποι-ημένοι σε μία υποθετική δομημένη ψευδογλώσσα, ωστόσο οι περισσότεροι από αυτούς μπορούν εύκολα σχετικά να προγραμματισθούν σε οποιαδήποτε γλώσσα προγραμματισμού.**

**53 /35**

**Σύμβολα διαγράμματος ροής.**

**Ένα διάγραμμα ροής αποτελείται από ένα σύνολο γεωμετρικών σχημάτων, όπου το καθένα δηλώνει μία συγκεκριμένη ενέργεια ή λειτουργία. Τα γεωμετρικά σχήματα ενώνονται μεταξύ τους με βέλη, που δηλώνουν τη σειρά εκτέλεσης των ενεργειών αυτών. Τα κυριότερα χρησιμοποιούμενα γεωμετρικά σχήματα είναι τα εξής:**

* **έλλειψη, που δηλώνει την αρχή και το τέλος του κάθε αλγορίθμου,**
* **ρόμβος, που δηλώνει μία ερώτηση με δύο ή περισσότερες εξόδους για απάντηση,**

**Ψευδής**

**Αληθής**

**Συνθήκη**

**54 / 36**

* **ορθογώνιο, που δηλώνει την εκτέλεση μίας ή περισσότερων πράξεων και**

**118 / 29**

**Εκτέλεση Πράξεων**

* **πλάγιο παραλληλόγραμμο, που δηλώνει είσοδο ή έξοδο στοιχείων. Πολλές φορές το σχήμα αυτό μπορεί να διαφοροποιείται προκειμένου να προσδιορίζεται και το είδος της συσκευής απ' όπου γίνεται η είσοδος ή η έξοδος.**

**Είσοδος**

**Έξοδος**

**118 / 29**

**55 / 36**

|  |  |
| --- | --- |
| **2.4** | **Βασικές συνιστώσες/εντολές ενός αλγορίθμου** |

**Στη συνέχεια δίνονται παραδείγματα αλγορίθμων όπου εξετάζονται οι διάφορες συνιστώσες ενός αλγορίθμου, δηλαδή οι απαραίτητες εντολές ξεκινώντας από τις απλούστερες και προχωρώντας προς τις συνθετό-τερες. Πιο συγκεκριμένα θα εξετασθούν περιπτώσεις σειριακών εντολών, αναθέσεων τιμών, επιλογής με βάση κριτήρια, διαδικασιών επανάληψης, ενεργειών πολλαπλών επιλογών καθώς και συνδυασμό εμφω-λευμένων περιπτώσεων. Για κάθε περίπτωση παρου-σιάζονται σχετικά παραδείγματα με την εκφώνηση (σε φυσική γλώσσα), την παρουσίαση των βημάτων που πρέπει να ακολουθηθούν καθώς και το αντίστοιχο διάγραμμα ροής.**

**2. 4 .1 Δομή ακολουθίας**

**Η ακολουθιακή δομή εντολών (σειριακών βημάτων) χρησιμοποιείται πρακτικά για την αντιμετώπιση απλών προβλημάτων, όπου είναι δεδομένη η σειρά εκτέ-λεσης ενός συνόλου ενεργει-ών. Ένα απλό παράδειγμα από την καθημερινή ζωή είναι η ακολουθία οδηγιών μίας συνταγής μαγειρικής με στόχο την κατασκευή ενός φαγητού. Τα βήματα και οι ποσότητες που πρέπει να ακολουθηθούν είναι συγκε-κριμένα και οι οδηγίες απόλυτα καθορισμένες και σαφείς. Το παράδειγμα που ακολουθεί παρουσιάζει ένα απλό πρόβλημα που επιλύεται με σειριακή εκτέ-λεση εντολών.**

**56 / 35-36**

|  |  |
| --- | --- |
|  | **Διάβασε =εκτελεστέα εντολή**  **Αλγόριθμος =δηλωτική εντολή** |

**ΠΑΡΑΔΕΙΓΜΑ 1. Ανάγνωση και εκτύπωση αριθμών**

**Να διαβασθούν δυο αριθμοί, να υπολογισθεί και να εκτυπωθεί το άθροισμα τους.**

**Από την εκφώνηση προκύπτει αμέσως ο επόμενος αλγόριθμος**

Αλγόριθμος  **Παράδειγμα\_1**   
**Διάβασε a**

**Διάβασε b**

c a + b

Εκτύπωσε c  
Τέλος **Παράδειγμα\_1**

**Ένας αλγόριθμος διατυπωμένος σε ψευδογλώσσα αρχίζει πάντα με τη λέξη Αλγόριθμος συνοδευόμενη με το όνομα του αλγορίθμου και τελειώνει με τη λέξη Τέλος συνοδευόμενη επίσης με το όνομα του αλ-γορίθμου. Η πρώτη ενέργεια που γίνεται, είναι η εισαγωγή των δεδομένων. Αυτό επιτυγχάνεται με τη χρήση του ρήματος Διαβάζω σε προστακτική. Η λέξη Διάβασε συνοδεύεται με το όνομα μίας ή περισσο-τέρων μεταβλητών, όπως η a και εννοείται ότι μετά την ολοκλήρωση της ενέργειας αυτή, η μεταβλητή a θα έχει λάβει κάποια αριθμητική τιμή ως περιεχόμενο. Κάθε μία λέξη της χρησιμοποιούμενης ψευδογλώσσας, που προσδιορίζει μια σαφή ενέργεια, θα αποκαλείται στο εξής εντολή. Όλες οι εντολές σε έναν αλγόριθμο από-τυπώνονται με διαφορετικό χρώμα από το όνομα του αλγορίθμου και τις διάφορες σταθερές και μεταβλητές.**

**57 / 37**

**Σταθερές (constants). Με τον όρο αυτό αναφερό-μαστε σε προκαθορισμένες τιμές που παραμένουν αμετάβλητες σε όλη τη διάρκεια της εκτέλεσης ενός αλγορίθμου. Οι σταθερές διακρίνονται σε**

* **αριθμητικές, π.χ. 123, +5, -1,25**
* **αλφαριθμητικές π.χ. "Τιμή", "Κατάσταση αποτελεσμάτων"**
* **λογικές που είναι ακριβώς δύο, Αληθής και Ψευδής**

**Μεταβλητές (variables). Μια μεταβλητή είναι ένα γλωσσικό αντικείμενο, που χρησιμοποιείται για να παραστήσει ένα στοιχείο δεδομένου. Στη μεταβλητή εκχωρείται μια τιμή, η οποία μπορεί να αλλάζει κατά τη διάρκεια εκτέλεσης του αλγορίθμου. Ανάλογα με το είδος της τιμής που μπορούν να λάβουν, οι μεταβλητές διακρίνονται σε αριθμητικές, αλφαριθμητικές και λογικές.**

**Τελεστές (operators). Πρόκειται για τα γνωστά σύμ-βολα που χρησιμοποιούνται στις διάφορες πράξεις. Οι τελεστές διακρίνονται σε αριθμητικούς, λογικούς και συγκριτικούς.**

**58 / 37**

**Εκφράσεις (expressions). Οι εκφράσεις διαμορ-φώνονται από τους τελεστέους (operands), που είναι σταθερές και μεταβλητές και από τους τελεστές.**

**Η διεργασία αποτίμησης μιας έκφρασης συνίσταται στην απόδοση τιμών στις μεταβλητές και στην εκτέ-λεση των πράξεων. Η τελική τιμή μιας έκφρασης εξαρτάται από την ιεραρχία των πράξεων και τη χρήση των παρενθέσεων. Μια έκφραση μπορεί να αποτε-λείται από μια μόνο μεταβλητή ή σταθερά μέχρι μια πολύπλοκη μαθηματική παράσταση.**

**Μετά την ανάγνωση των τιμών των μεταβλητών a και b γίνεται ο υπολογισμός του αθροίσματος με την εντολή: c a + b. Η εντολή αυτή αποκαλείται εντολή εκχώ-ρησης τιμής. Η γενική μορφή της είναι:**

**Μεταβλητή**  **Έκφραση**

**και η λειτουργία της είναι "γίνονται οι πράξεις στην έκφραση και το αποτέλεσμα αποδίδεται, μεταβιβάζεται, εκχωρείται στη μεταβλητή". Στην εντολή αυτή χρησι-μοποιείται το αριστερό βέλος, προκειμένου να δείχνει τη φορά της εκχώρησης. Ας σημειωθεί ότι δεν πρό-κειται για εξίσωση, παρόλο που σε άλλα βιβλία μπορεί να χρησιμοποιείται το σύμβολο ίσον "=" για τον ίδιο σκοπό.**

**59 / 37-38**

**Σχ. 2.1. Ο αλγόριθμος του παραδείγματος 1 με διάγραμμα ροής**

**Διάβασε a, b**

**c 🡨 a + b**

**Εκτύπωσε c**

**Ας σημειωθεί επίσης ότι οι διάφορες γλώσσες προ-γραμματισμού χρησιμοποιούν διάφορα σύμβολα για το σκοπό αυτό.**

**Τέλος ο αλγόριθμος ολοκληρώνεται με την εντολή Εκτύπωσε, που αποτυπώνει το τελικό αποτέλεσμα στον εκτυπωτή. Η σύνταξη της εντολής αυτής είναι ανάλογη με αυτή της Διάβασε. Εναλλακτικά μπορεί να χρησιμοποιηθεί και η εντολή Εμφάνισε, που αποτυ-πώνει ένα αποτέλεσμα στην οθόνη.**

**60 / 37-38**

**Στον προηγούμενο αλγόριθμο οι μεταβλητές a και b είναι τα δεδομένα που αποτελούν την είσοδο, ενώ η μεταβλητή c αντιπροσωπεύει το αποτέλεσμα, δηλαδή την έξοδο του αλγορίθμου. Επιπλέον, ο αλγόριθμος έχει απολύτως καθορισμένη την κάθε εντολή (καθορι-στικότητα), τελειώνει μετά από συγκεκριμένο αριθμό βημάτων (περατότητα), ενώ κάθε εντολή του είναι ιδιαίτερα σαφής και απλή (αποτελεσματικότητα). Επομένως ο αλγόριθμος αυτός πληροί τα κριτήρια που χαρακτηρίζουν τον ορισμό της έννοιας του αλγορίθ-μου, όπως αυτά περιγράφηκαν στην παράγραφο 2.1.**

**2.4.2 Δομή Επιλογής**

**Στην πραγματικότητα πολύ λίγα προβλήματα μπορούν να επιλυθούν με τον προηγούμενο τρόπο της σειριακής/ακολουθιακής δομής ενεργειών. Συνήθως τα προ-βλήματα έχουν κάποιες ιδιαιτε-ρότητες και δεν ισχύουν τα ίδια βήματα για κάθε περίπτωση. Η πλέον συνηθισμένη περίπτωση είναι να λαμβάνονται κάποιες αποφάσεις με βάση κάποια δεδομένα κριτήρια, που μπορεί να είναι διαφορετικά για κάθε διαφορετικό στιγμιότυπο ενός προβλήματος. Οι καθημερινές απλές μας ενέργειες περιέχουν αυτή τη διαδικασία επιλογής με βάση κάποια κατάσταση. Για παράδειγμα, το πρόβλημα της προετοιμασίας μας για έξοδο σχετίζεται με τις καιρικές συνθήκες. Έτσι λέμε ότι, "αν βρέχει, θα πάρω ομπρέ-λα, αλλιώς θα πάρω καπέλο". Η συνθήκη εδώ είναι το "αν βρέχει", ενώ η απόφαση είναι είτε να πάρω την "ομπρέλα" είτε το "καπέλο" με βάση την "τιμή" της συνθήκης.**

**61 / 37-38**

**Γενικά η διαδικασία της επιλογής περιλαμβάνει τον έλεγχο κάποιας συνθήκης που μπορεί να έχει δύο τιμές (Αληθής ή Ψευδής) και ακολουθεί η απόφαση εκτέλεσης κάποιας ενέργειας με βάση την τιμή της λογικής αυτής συνθήκης. Στη συνέχεια δίνονται δύο παραδείγματα ενεργειών με βάση κάποια συνθήκης επιλογής. Το πρώτο παράδειγμα αφορά στην εκτέλεση κάποιας ενέργειας όταν η συνθήκη είναι Αληθής, ενώ το δεύτερο παράδειγμα αφορά στην εκτέλεση μίας ενέργειας όταν η συνθήκη είναι Αληθής και κάποιας άλλης ενέργειας όταν η συνθήκη είναι Ψευδής.**

**ΠΑΡΑΔΕΙΓΜΑ 2. Σύγκριση αριθμών με απλή επιλογή**

**62 / 38-39**

**Να διαβαστεί ένας αριθμός και να εκτυπωθεί η απόλυτη τιμή του.**

|  |  |
| --- | --- |
|  | **|+5| = 5 και |-5|=5** |

**Όπως είναι γνωστό, η απόλυτη τιμή ενός αριθμού είναι ο ίδιος ο αριθμός, αν αυτός είναι θετικός ή μηδέν και ο αντίθετος του, αν είναι αρνητικός. Έτσι προκειμένου να βρεθεί η απόλυτη τιμή, αρκεί να ελεγχθεί, αν τυχόν ο δεδομένος αριθμός είναι αρνητικός, οπότε στην περί-πτωση αυτή πρέπει να βρεθεί ο αντίθετος του. Ο συλλογισμός αυτός οδηγεί στον επόμενο αλγόριθμο.**

**Αλγόριθμος  Παράδειγμα\_2   
Διάβασε a**

**Αν a < 0 τότε** a 🡨 a \* (-1)

Εκτύπωσε a  
Τέλος **Παράδειγμα\_2**

**Στην παράσταση αλγορίθμων με ψευδογλώσσα η επιλογή υλοποιείται με την εντολή Αν...τότε. Η σύνταξη της εντολής είναι:**

**Αν συνθήκη τότε εντολή**

**και η λειτουργία της είναι: Αν ισχύει η συνθήκη (δηλα-δή αν είναι αληθής), τότε μόνο εκτελείται η εντολή. Σε κάθε περίπτωση εκτελείται στη συνέχεια η εντολή, που ακολουθεί.**

|  |  |
| --- | --- |
|  | **Η συνθήκη είναι μια λογική έκφραση.** |

**Στην εντολή Αν...τότε είναι πιθανό, όταν ισχύει η συν-θήκη, να απαιτείται η εκτέλεση περισσότερων από μία εντολές. Στην περίπτωση αυτή οι διαδοχικές εντολές γράφονται από κάτω και σε εσοχή, ενώ το σχήμα επιλογής κλείνει με τη λέξη Τέλος\_αν. Π.χ**

**63 / 39**

**Αν συνθήκη τότε**

**εντολή\_1**

**εντολή\_1**

**…………….**

**εντολή\_ν**

**Τέλος\_αν**

**Όπως και στον αλγόριθμο του προηγούμενου παρα-δείγματος, εύκολα προκύπτει ότι η τιμή a είναι και είσοδος αλλά και έξοδος του αλγορίθμου. Επιπλέον, ο αλγόριθμος έχει καθορισμένη κάθε του εντολή (καθο-ριστικότατα), τελειώνει μετά από πεπερασμένο αριθμό βημάτων (περατότητα), ενώ κάθε εντολή του είναι ιδιαίτερα απλή κατά την εκτέλεσή της (αποτελεσμα-τικότητα). Έτσι προκύπτει ότι ο αλγόριθμος αυτός πράγματι πληροί τα κριτήρια που περιγράφηκαν στην παράγραφο 2.1.**

**64 / 39**

**Σχ. 2.2. Ο αλγόριθμος του παραδείγματος 2 με διάγραμμα ροής**

**ΝΑΙ**

**ΟΧΙ**

**Διάβασε a**

**a 🡨 a \* (-1**)

**Εκτύπωσε a**

**a < 0**

**ΠΑΡΑΔΕΙΓΜΑ 3 Σύγκριση αριθμών με σύνθετη επιλογή**

**Να διαβασθούν δύο αριθμοί και σε περίπτωση που ο πρώτος αριθμός είναι μικρότερος του δεύτερου, να υπολογισθεί και να εκτυπωθεί το άθροισμα τους, διαφορετικά να υπολογισθεί και να εκτυπωθεί το γινόμενο τους.**

Αλγόριθμος  **Παράδειγμα\_3**   
**Διάβασε a, b**

**Αν a < b τότε** c 🡨 a + b

αλλιώς

c 🡨 a \* b

**Τέλος\_αν**

Εκτύπωσε c  
Τέλος **Παράδειγμα\_3**

**Στο παράδειγμα αυτό χρησιμοποιείται η γενική μορφή της εντολής επιλογής, που είναι:**

**Αν συνθήκη τότε**

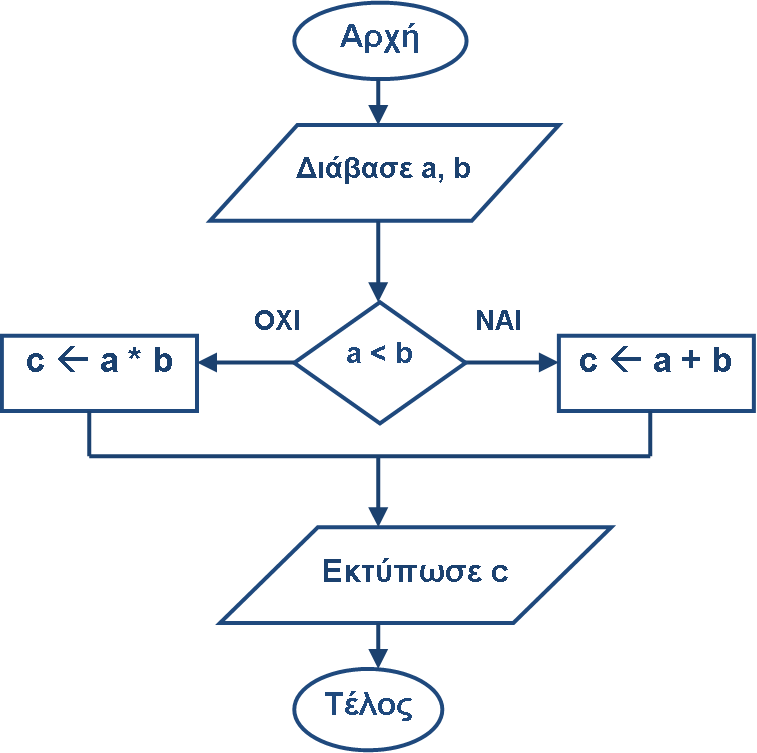
**εντολή ή εντολές**

αλλιώς

**εντολή ή εντολές**

**Τέλος\_αν**

**65 / 40**

****

**66 / 40**

**Σχ. 2.3. Ο αλγόριθμος του παραδείγματος 3 με διά-γραμμα ροής**

**2.4.3 Διαδικασίες πολλαπλών επιλογών**

**Οι διαδικασίες των πολλαπλών επιλογών εφαρμό-ζονται στα προβλήματα όπου μπορεί να ληφθούν διαφορετικές αποφάσεις ανάλογα με την τιμή που παίρνει μία έκφραση. Για παράδειγμα, κάθε γράμμα της αλφαβήτου μπορεί να αντιστοιχηθεί σε κάποιον ακέραιο αριθμό από το 1 μέχρι και 24, για τις ανάγκες κάποιας κωδικοποίησης. Στο παράδειγμα που ακο-λουθεί παρουσιάζεται μία περίπτωση πολλαπλών επιλογών με διαφορετική ακολουθία εντολών σε κάθε περίπτωση.**

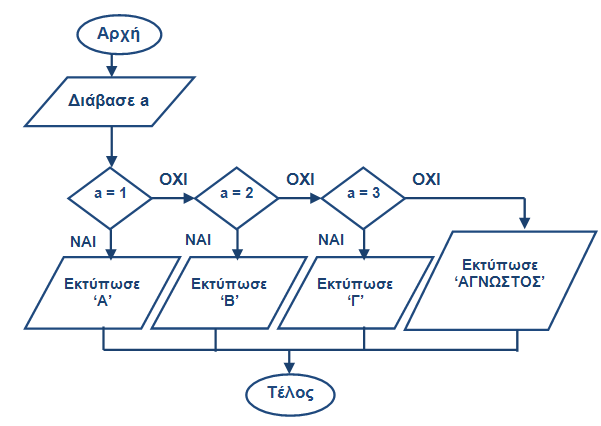
**67 / 40-41**

**ΠΑΡΑΔΕΙΓΜΑ 4 Ανάθεση γραμμάτων σε αριθμούς**

**Να διαβασθεί ένας ακέραιος και να εκτυπωθεί το αντίστοιχο γράμμα της αλφαβήτου, αν ο ακέραιος έχει τιμή 1 ή 2 ή 3 διαφορετικά να εκτυπωθεί η λέξη "άγνωστος".**

Αλγόριθμος  **Παράδειγμα\_4**   
**Διάβασε a**

**Αν a = 1 τότε** Εκτύπωσε “Α” αλλιώς\_αν **a = 2 τότε** Εκτύπωσε “Β” αλλιώς\_αν **a = 2 τότε** Εκτύπωσε “Γ” αλλιώς Εκτύπωσε “άγνωστος” **Τέλος\_αν**Τέλος **Παράδειγμα\_4**

**Σχ. 2.4. Ο αλγόριθμος του παραδείγματος 4 με διάγραμμα ροής**

**68 / 41**

**154 / 22**

**Αν οι διαφορετικές επιλογές είναι πολλές, τότε είναι προτιμότερο να χρησιμοποιηθεί το σχήμα** **πολλαπλής επιλογής Επίλεξε...Τέλος επιλογών (select case), όπως στο παράδειγμα που ακολουθεί.**

**ΠΑΡΑΔΕΙΓΜΑ 5 Επιλογή ορίων**

**Να εισαχθεί ένας ακέραιος που αντιστοιχεί σε μια ηλικία και να βρεθεί σε ποια όρια εντάσσεται η δεδομένη ηλικία εμφανίζοντας σχετικό μήνυμα.**

Αλγόριθμος  **Παράδειγμα\_5**

**Εμφάνισε “Σε ποια ηλικία άρχισες να μαθαίνεις προγραμματισμό;”**

**Διάβασε age**

**Επίλεξε age**

**Περίπτωση < 0** Εμφάνισε “Είπαμε ηλικία…” **Περίπτωση < 5** Εμφάνισε “Μάλλον τα παραλές!!” **Περίπτωση < 60** Εμφάνισε “Μπράβο”

**Περίπτωση < 100** Εμφάνισε “Ποτέ δεν είναι αργά”

**Περίπτωση αλλιώς**

Εμφάνισε “Κάλλιο αργά παρά ποτέ”

Τέλος\_επιλογών

Τέλος **Παράδειγμα\_5**

**69 / 41-42**

**Στο σχήμα πολλαπλής επιλογής Επίλεξε...Τέλος επιλογών, που χρησιμοποιήθηκε στο παράδειγμα αυτό, εξετάζεται μια έκφραση (εδώ είναι μια μόνο μεταβλητή, η age).**

**Ανάλογα με την τιμή της έκφρασης εκτελούνται οι εντολές μετά την Περίπτωση που αντιστοιχεί στην τιμή της έκφρασης.**

**Οι τιμές που συνοδεύουν κάθε Περίπτωση μπορεί να είναι μία ή περισσότερες διακριτές τιμές, περιοχή τιμών από - έως ή να υπακούουν σε μια συνθήκη (όπως στο παράδειγμα αυτό).**

**Αν η τιμή της έκφρασης δεν αντιστοιχεί σε καμία Περίπτωση, τότε εκτελούνται οι εντολές που ακο-λουθούν την Περίπτωση αλλιώς.**

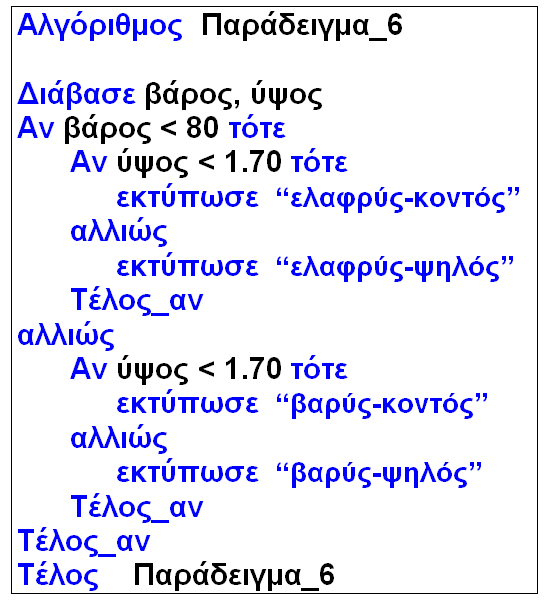
**Μετά την εκτέλεση μιας περίπτωσης, ο αλγόριθμος συνεχίζει με την εντολή, που ακολουθεί το Τέλος επιλογών**

**70 / 42**

**2.4.4 Εμφωλευμένες Διαδικασίες**

**Πολλαπλές επιλογές μπορούν να γίνουν και με μία εμφωλευμένη δομή. Το επόμενο παράδειγμα περι-γράφει τον τρόπο με τον οποίο μία εντολή Αν...τότε είναι η εντολή που εκτελείται, όταν ισχύει (ή δεν ισχύει) η συνθήκη μίας άλλης εντολής Αν...τότε. Βέβαια η λογική αυτή μπορεί να επεκταθεί, δηλαδή να έχουμε νέα εμφωλευμένη δομή μέσα σε μία εμφωλευμένη δομή κ.ο.κ.**

**ΠΑΡΑΔΕΙΓΜΑ 6. Χαρακτηρισμός ατόμων**

**Να διαβάζονται δυο αριθμοί που αντιστοιχούν στο ύψος και βάρος ενός άνδρα. Να εκτυπώνεται ότι ο άνδρας είναι "ελαφρός", αν το βάρος του είναι κάτω από 80 κιλά, ή να εκτυπώνεται "βαρύς" στην αντίθε-τη περίπτωση. Επίσης να εκτυπώνεται "κοντός" αν το ύψος του είναι κάτω από 1.70, αλλιώς να εκτυ-πώνεται "ψηλός".**

**71 / 42**



**Σχ. 2.5. Ο αλγόριθμος του παραδείγματος 6 με διάγραμμα ροής**

**72 / 43**

**152 / 38**

**Σε πολλές περιπτώσεις η συνθήκη είναι αρκετά πιο "δύσκολη", δηλαδή εμπεριέχει αποφάσεις που πιθανόν να βασίζονται σε περισσότερα από ένα κριτήρια. Ο συνδυασμός των κριτηρίων αυτών καθορίζει και τις "λογικές" πράξεις που μπορούν να γίνουν μεταξύ δια-φορετικών συνθηκών. Πολύ συχνά στην καθημερινή ζωή κάποιες αποφάσεις βασίζονται σε συνδυασμούς κριτηρίων και λογικών πράξεων. Για παράδειγμα, το πρόβλημα της προετοιμασίας μας για έξοδο μπορεί να επεκταθεί ως εξής "αν βρέχει ή αν χιονίζει θα πάρω ομπρέλα", είτε στην πρόταση "αν έχει ήλιο και αν έχει ζέστη θα πάρω καπέλο", είτε στην πρόταση "αν δεν έχει ήλιο θα πάρω ομπρέλα". Οι τρεις αυτές προτάσεις περιγράφουν και τις τρεις λογικές πράξεις που μπορεί να ισχύουν μεταξύ διαφορετικών συνθηκών. Η λογική πράξη ή είναι αληθής όταν οποιαδήποτε από τις δύο προτάσεις είναι αληθής. Η λογική πράξη και είναι αληθής όταν και οι δύο προτάσεις είναι αληθείς ενώ η λογική πράξη όχι (η λέξη "δεν" στο παράδειγμά μας) είναι αληθής όταν η πρόταση που την ακολουθεί είναι ψευδής. Ο επόμενος πίνακας δίνει τις τιμές των τριών αυτών λογικών πράξεων για όλους τους συνδυασμούς τιμών.**

**73 / 43**

**156/ 39**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Πρόταση Α**  **74 / 43** | **Πρόταση Β** | **Α ή Β** | **Α και Β** | **όχι Α** |
| **Αληθής** | **Αληθής** | **Αληθής** | **Αληθής** | **Ψευδής** |
| **Αληθής** | **Ψευδής** | **Αληθής** | **Ψευδής** | **Ψευδής** |
| **Ψευδής** | **Αληθής** | **Αληθής** | **Ψευδής** | **Αληθής** |
| **Ψευδής** | **Ψευδής** | **Ψευδής** | **Ψευδής** | **Αληθής** |

**2.4.5 Δομή Επανάληψης**

**Η διαδικασία της επανάληψης είναι ιδιαίτερα συχνή, αφού πλήθος προβλημάτων μπορούν να επιλυθούν με κατάλληλες επαναληπτικές διαδικασίες. Η λογική των επαναληπτικών διαδικασιών εφαρμόζεται στις περιπτώσεις, όπου μία ακολουθία εντολών πρέπει να εφαρμοσθεί σε ένα σύνολο περιπτώσεων, που έχουν κάτι κοινό. Για παράδειγμα, όλες οι τράπεζες κάθε εξάμηνο αποδίδουν τόκους των καταθέσεων ταμιευτηρίου. Ο υπολογισμός των τόκων πρέπει να γίνει για όλους τους λογαριασμούς της τράπεζας, άρα η πράξη**

**τόκος = ποσό \* επιτόκιο**\_

**πρέπει να εκτελεσθεί για όλους τους τραπεζικούς λογαριασμούς. Οι επαναληπτικές διαδικασίες μπορεί να έχουν διάφορες μορφές και συνήθως εμπεριέχουν και συνθήκες επιλογών (όπως αυτές περιγράφηκαν στην προηγούμενη υποπαράγραφο). Στη συνέχεια δίνεται ένα σύνολο παραδειγμάτων που εντάσσονται στις πλέον γνωστές κατηγορίες επαναληπτικών διαδικασιών.**

**75 / 44**

**ΠΑΡΑΔΕΙΓΜΑ 7. Εκτύπωση διαδοχικών αριθμών με επαναληπτική εντολή: όσο...επανάλαβε**

**Να γραφεί αλγόριθμος που να εμφανίζει τους αριθμούς από 1 έως 100.**

|  |  |
| --- | --- |
|  | **i i +1 δηλαδή  “η νέα τιμή της μεταβλητής i είναι η προηγούμενη συν ένα”.** |

**Στον αλγόριθμο αυτό επιζητείται η παρουσίαση μίας σειράς αριθμών. Αν οι αριθμοί αυτοί ήσαν λίγοι, τότε αυτό θα μπορούσε να γίνει με την παράθεση αντίστοιχων εντολών εμφάνισης. Το ίδιο θα συμβεί και στην περίπτωση που οι αριθμοί είναι περισσότεροι, αλλά δεν έχουν καμία σχέση μεταξύ τους π.χ. 5, 207, -32 κοκ. Όμως στο ζητούμενο αλγόριθμο παρατηρούμε ότι κάθε αριθμός παράγεται από τον προηγούμενό του με απλό τρόπο δηλαδή προσθέτοντας κάθε φορά το 1. Μπορεί λοιπόν να χρησιμοποιηθεί μια μεταβλητή, έστω i, η οποία αρχίζει από το 1 και καταλήγει στο 100 αυξανόμενη κατά 1. Η εκάστοτε αύξηση της μεταβλητής αυτής μπορεί να γίνει με τη χρήση της εντολής εκχώρησης**

**i i + 1**

**Η αρχική τιμή της μεταβλητής i ορίζεται εύκολα με την εντολή i 1.**

**76 / 44**

**Το ζητούμενο είναι να εκτελεστεί 100 φορές η εντολή   
i i + 1.**

**Αυτό επιτυγχάνεται με τη χρήση της εντολής Όσο ...επανάλαβε. Η σύνταξη της εντολής αυτής είναι:**

**Όσο συνθήκη επανάλαβε**

**εντολές**

**Τέλος\_επανάληψης**

**αληθής**

**συνθήκη**

**Εκτέλεση Πράξεων**

**ψευδής**

**Η λειτουργία της εντολής είναι η εξής: Επαναλαμ-βάνεται η εκτέλεση των εντολών, όσο η συνθήκη είναι αληθής. Όταν η συνθήκη γίνει ψευδής, τότε ο αλγό-ριθμος συνεχίζεται με την εντολή που ακολουθεί το 'Τέλος\_επανάληψης'. Με την εισαγωγή της εντολής αυτής η σχεδίαση του ζητούμενου αλγορίθμου είναι:**

**77 / 44-45**

Αλγόριθμος  **Παράδειγμα\_7**   
i 🡨 1

**Όσο** i **≤ 100 επανάλαβε**

Εμφάνισε i  
 i 🡨 i + 1

Τέλος\_επανάληψης

Τέλος **Παράδειγμα\_7**

**ΠΑΡΑΔΕΙΓΜΑ 8. Επαναληπτική είσοδος στοιχείων**

**Να γραφεί αλγόριθμος που να διαβάζει ένα άγνωστο πλήθος αριθμών και να εμφανίζει τον κάθε αριθμό.**

**Το πρόβλημα αυτό παρουσιάζει την εξής ιδιομορφία: ενώ φαίνεται ότι θα χρησιμοποιηθεί για τη λύση του κάποια επαναληπτική διαδικασία, δεν προσδιορίζεται ο τρόπος τερματισμού της. Καταρχήν, λοιπόν, ας εξετά-σουμε τον αλγόριθμο που εκτελεί ένας άνθρωπος, όταν αντιγράφει κάποιους αριθμούς, όπως για παράδειγμα όταν συγκεντρώνονται τα έξοδα από διάφορους λογα-ριασμούς. Ο αλγόριθμος αυτός είναι:**

**78 / 45**

**Βήμα 1. Διάβασε έναν αριθμό**

**Βήμα 2. Γράψε τον αριθμό**

**Βήμα 3. Επανάλαβε τη διαδικασία από το βήμα 1.**

**Ο αλγόριθμος αυτός έχει μια ατέλεια, δεν διαθέτει τρόπο τερματισμού (ατέρμων βρόχος). Η έλλειψη αυτή είναι φυσική, εφόσον ο αλγόριθμος εκτελείται από έναν άνθρωπο. Αυτός θα σταματήσει να γράφει, όταν δεν υπάρχουν πλέον άλλοι αριθμοί. Ωστόσο είναι δυνατόν να διορθωθεί αυτή η ατέλεια, αν το βήμα 3 λάβει την εξής μορφή:**

**Βήμα 3. Αν υπάρχουν άλλοι αριθμοί, επανάλαβε τη διαδικασία από το βήμα 1, αλλιώς σταμάτησε.**

**Ο αλγόριθμος τώρα είναι σωστός και μπορεί να εκτε-λεστεί και από μία μηχανή. Όμως έχει ένα άλλο μειο-νέκτημα: ο τερματισμός γίνεται μέσα από την εντολή Αν ... τότε ... αλλιώς ..., πράγμα που δεν συνιστάται και πρέπει να αποφεύγεται, γιατί εύκολα μπορεί να χάσει ο προγραμματιστής τον έλεγχο της ροής του προγράμ-ματος και να οδηγηθεί σε λάθος. Για την άρση του μειονεκτήματος αυτού πρέπει να χρησιμοποιηθεί μία εντολή επαναληπτικής διαδικασίας, όπως η εντολή** **Όσο...επανάλαβε. Ο τελικός αλγόριθμος είναι ο εξής:**

Αλγόριθμος  **Παράδειγμα\_8**   
Διάβασε x  
**Όσο** x > 0 **επανάλαβε** Εμφάνισε x  
 Διάβασε x  
Τέλος\_επανάληψης  
Τέλος **Παράδειγμα\_8**

|  |  |
| --- | --- |
| πιν.jpg | **Ο βρόχος επανάληψης μπορεί να μην εκτελεσθεί καμία φορά, αν η πρώτη τιμή που διαβάζεται είναι αρνητική.** |

**Στον προηγούμενο αλγόριθμο η επαναληπτική διαδι-κασία τερματίζεται, όταν διαβασθεί ένας αρνητικός ή μηδενικός αριθμός. Δηλαδή, θεωρείται ότι οι εισαγό-μενοι αριθμοί πρέπει να είναι θετικοί. Αν αυτό δεν συμβαίνει, τότε μπορεί να χρησιμοποιηθεί ως συνθήκη τερματισμού οποιαδήποτε συγκεκριμένη τιμή έχει συμφωνηθεί ότι θα χρησιμοποιείται για το σκοπό αυτό, π.χ. η 999999. Προφανώς αυτή η τιμή δεν μπορεί να ανήκει στις εισαγόμενες τιμές. Στην περίπτωση αυτή η εντολή Όσο...επανάλαβε θα γραφεί ως εξής:**

**Όσο x ≠ 999999 επανάλαβε**

**ΠΑΡΑΔΕΙΓΜΑ 9. Εκτύπωση Θετικών αριθμών με εντολή: αρχή\_επανάληψης, μέχρις\_ότου**

**Να διαβάζονται και να εκτυπώνονται όσοι θετικοί αριθμοί δίνονται από το πληκτρολόγιο. Ο αλγόριθ-μος τελειώνει, όταν δοθεί ένας αρνητικός αριθμός.**

**79/ 45-46**

Αλγόριθμος  **Παράδειγμα\_9**   
Αρχή\_επανάληψης

Διάβασε x  
 Εμφάνισε x  
**Μέχρις\_ότου** x < 0

Τέλος **Παράδειγμα\_9**

**αληθής**

**συνθήκη**

**Εκτέλεση Πράξεων**

**ψευδής**

|  |  |
| --- | --- |
| πιν.jpg | **Η εντολή Αρχή\_επανάληψης... Μέχρις\_ότου εκτελείται οπωσδήποτε μια φορά.** |

**Ας σημειωθεί ότι, στο παράδειγμα αυτό ο βρόχος επανάληψης θα εκτελεσθεί οπωσδήποτε τουλάχιστον μία φορά ακόμα και αν η αρχική τιμή της μεταβλητής χ είναι αρνητική. Η βασική διαφοροποίηση αυτής της μορφής επαναληπτικής διαδικασίας σε σχέση με την επαναληπτική διαδικασία που παρουσιάσθηκε στο προηγούμενο παράδειγμα, οφείλεται στη θέση της λογικής συνθήκης στη ροή εκτέλεσης των εντολών.**

**80 / 46**

**ΠΑΡΑΔΕΙΓΜΑ 10.Υπολογισμός αθροίσματος αριθμών με επαναληπτική εντολή: για...από...μέχρι**

**Να βρεθεί και να εκτυπωθεί το άθροισμα των 100 ακεραίων από το 1 μέχρι το 100.**

**Όταν ο αριθμός των φορών που θα εκτελεστεί μια επα-ναληπτική διαδικασία είναι γνωστός εκ των προτέρων, τότε είναι προτιμότερο να χρησιμοποιείται η εντολή Για...από...μέχρι. Έτσι ο ζητούμενος αλγόριθμος είναι:**

Αλγόριθμος  **Παράδειγμα\_10**   
**Sum**  🡨 0

**81 / 46**

Για i από 1 μέχρι 100  
 **Sum**  🡨 Sum + i

Τέλος\_επανάληψης

Εκτύπωσε Sum  
Τέλος **Παράδειγμα\_10**

|  |  |
| --- | --- |
|  | **Sum** 🡨 **Sum + i  δηλαδή η νέα τιμή του Sum είναι η παλιά συν i.** |

**Επίσης γίνεται φανερό, η εντολή Για...από...μέχρι περιλαμβάνει όλα τα απαιτούμενα στοιχεία για την επανάληψη, δηλαδή αρχική τιμή επίσης μεταβλητής i (=1) και τελική τιμή (=100). Το βήμα μεταβολής επίσης μεταβλητής i είναι 1, το οποίο υπονοείται και δεν ση-μειώνεται, όταν είναι 1. Η μεταβλητή Sum που υπο-δέχεται το άθροισμα των διαδοχικών αριθμών, πρέπει να εκκινήσει με τιμή 0, ενώ το εκάστοτε μερικό άθροισμα υπολογίζεται με την εντολή εκχώρησης εντός του βρόχου. Στο τέλος η μεταβλητή Sum θα περιέχει το τελικό άθροισμα.**

**ΟΧΙ**

**ΝΑΙ**

**i ≤100**

**Εκτύπωσε  
sum**

**i 🡨1**

**sum 🡨 0**

**sum 🡨sum + i**

**i🡨 i+1**

**Σχ. 2.6. Ο αλγόριθμος του παραδείγματος 10 με διά-γραμμα ροής**

**ΠΑΡΑΔΕΙΓΜΑ 11. Υπολογισμός αθροίσματος με επαναληπτική εντολή: για...από...μέχρι...με \_βήμα**

**82 / 46-47**

**Να βρεθεί και να εκτυπωθεί το άθροισμα των άρτιων αριθμών από το 1 μέχρι το 100.**

**Η λύση αυτού του προβλήματος είναι παρόμοια με αυτή του προηγούμενου. Η μόνη αλλαγή είναι στην εντολή επανάληψης όπου προσδιορίζεται η ποσότητα βήμα, η οποία κάθε φορά προστίθεται στην τιμή επίσης μετα-βλητής ί. Έτσι έχουμε**

Αλγόριθμος  **Παράδειγμα\_11**   
**άθροισμα**  🡨 0

Για i από 2 μέχρι 100 με\_βήμα 2  
 **άθροισμα**  🡨 **άθροισμα**  + i

Τέλος\_επανάληψης

Εκτύπωσε **άθροισμα**    
Τέλος **Παράδειγμα\_11**

**Από τα προηγούμενα δύο παραδείγματα γίνεται φανε-ρός ο τρόπος χρήσης επίσης εντολής Για...από...μέχρι. Επίσης σημειωθεί ωστόσο, ότι υπάρχουν κάποιες δεσμεύσεις μεταξύ των τιμών από, μέχρι και βήμα. Έτσι το βήμα δεν μπορεί να είναι μηδέν, γιατί τότε ο βρόχος εκτελείται επ’ άπειρον. Είναι δυνατόν επίσης το βήμα να έχει αρνητική τιμή, αρκεί η τιμή από να είναι μεγα-λύτερη από την τιμή μέχρι, επίσης για παράδειγμα στην επόμενη εντολή:**

**Για k από 100 μέχρι 0 με\_βήμα -1**

**83 / 47**

**Επίσης οι τιμές από, μέχρι και βήμα δεν είναι απαραί-τητο να είναι ακέραιες. Μπορούν λάβουν οποιαδήποτε πραγματική τιμή. Για παράδειγμα, όταν ζητείται να βρεθούν διαδοχικές τιμές μιας συνάρτησης f(x) για χ από 0 έως 1, τότε μπορεί να γραφεί η επόμενη εντολή**

**Για χ από 0 μέχρι 1 με\_βήμα 0,01**

|  |  |
| --- | --- |
|  | **O βρόχος Για k από 5 μέχρι 5 εκτελείται ακριβώς μία φορά O βρόχος Για k από 5 μέχρι 1 δεν εκτελείται καμία φορά.** |

**ΠΑΡΑΔΕΙΓΜΑ 12.Πολλαπλασιασμός αλά ρωσικά**

**Στη συνέχεια προχωρούμε στην ανάπτυξη ενός συνθε-τότερου προβλήματος, όπου για την αλγοριθμική του επίλυση γίνεται χρήση αρκετών από τις προηγούμενες δομές. Ας θεωρήσουμε την πράξη του πολλαπλα-σιασμού δύο ακεραίων αριθμών και ας θυμηθούμε πώς αυτή υλοποιείται χειρωνακτικά. Τοποθετούμε, λοιπόν, τους δύο αριθμούς τον ένα κάτω από τον άλλο και πολλαπλασιάζουμε κάθε ψηφίο του κάτω αριθμού με όλα τα ψηφία του επάνω αριθμού. Πιο συγκεκριμένα, για κάθε ψηφίο του κάτω αριθμού παράγεται ένα μερικό γινόμενο, ενώ τα μερικά γινόμενα τοποθετούνται το ένα κάτω από το άλλο με μία μετατόπιση από τα δεξιά προς τα αριστερά καθώς θεωρούμε διαδοχικά τα ψηφία των μονάδων, των δεκάδων, των εκατοντάδων κ.λπ. Στη συνέχεια γίνεται η πρόσθεση των επιμέρους γινομένων, αφού τα τοποθετήσουμε στην κατάλληλη διάταξη όπως φαίνεται στο σχήμα 2.7.**

**84 / 47-48**

|  |  |
| --- | --- |
|  | 45 |
| × | 19 |
|  | 405 |
| + | 45 |
|  | 855 |

**Σχ. 2.7. Χειρωνακτικός τρόπος πολλαπλασιασμού.**

**Ωστόσο, η πράξη του πολλαπλασιασμού δεν εκτελείται από τον υπολογιστή με τον τρόπο αυτό. Πιο συγκε-κριμένα, ο χρησιμοποιούμενος τρόπος είναι ο λεγό-μενος πολλαπλασιασμός αλά ρωσικά. Χωρίς βλάβη της γενικότητας θεωρούμε ότι οι ακέραιοι είναι θετικοί (μεγαλύτεροι του μηδενός), αλλά η μέθοδος μπορεί εύκολα να μετατραπεί, ώστε να περιγράψει και την περίπτωση των αρνητικών ακεραίων. Πώς ακριβώς λειτουργεί η μέθοδος, θα φανεί με το επόμενο παρά-δειγμα, όπου περιγράφεται ο αλγόριθμος με ελεύθερο κείμενο.**

**85 / 48**

|  |  |  |
| --- | --- | --- |
| **45** | **19** | **45** |
| **90** | **9** | **90** |
| **180** | **4** |  |
| **360** | **2** |  |
| **720** | **1** | **720** |
| **Άθροισμα=** | | **855** |
| **Σχ.2.8.Πολλαπλασιασμός αλά ρωσικά** | | |

**Έστω, λοιπόν, ότι δίνονται δύο θετικοί ακέραιοι αριθ-μοί, οι αριθμοί 45 και 19. Οι αριθμοί γράφονται δίπλα-δίπλα και ο πρώτος διπλασιάζεται, ενώ ο δεύτερος υποδιπλασιάζεται αγνοώντας το δεκαδικό μέρος. Στο σχήμα 2.8 παρουσιάζεται η επαναλαμβανόμενη δια-δικασία, που συνεχίζεται μέχρις ότου στη δεύτερη στήλη να προκύψει μονάδα. Τελικώς, το γινόμενο ισούται με το άθροισμα των στοιχείων της πρώτης στήλης, όπου αντίστοιχα στη δεύτερη στήλη υπάρχει περιττός αριθμός. Για το παράδειγμά μας, τα στοιχεία αυτά παρουσιάζονται στην τελευταία στήλη.**

**86 / 48**

**ΟΛΙΣΘΗΣΗ (SHIFT)**

**Στα κυκλώματα του υπολογιστή τα δεδομένα αποθη-κεύονται με δυαδική μορφή, δηλαδή 0 και 1, ανεξάρτητα από το πως τα ορίζει ο προγραμματιστής, όπως ακε-ραίους ή πραγματικούς σε δεκαδικό σύστημα, ή ακόμη χαρακτήρες κ.λπ. Έτσι ο αριθμός 17 του δεκαδικού συστήματος ισοδυναμεί με τον αριθμό 00010001 του δυαδικού συστήματος, ο οποίος μπορεί να αποθη-κευθεί σε ένα byte. Αν μετακινήσουμε τα ψηφία αυτά κατά μία θέση προς τα αριστερά, δηλαδή αν προσθέ-σουμε ένα 0 στο τέλος του αριθμού και αγνοήσουμε το αρχικό 0, τότε προκύπτει ο αριθμός 00100010 του δυαδικού συστήματος, που ισοδυναμεί με το αριθμό 34 του δεκαδικού συστήματος. Επίσης, με παρόμοιο τρόπο, αν μετακινήσουμε τα ψηφία κατά μία θέση δεξιά, δηλαδή αποκόψουμε το τελευταίο ψηφίο 1 και θεωρή-σουμε ένα ακόμη αρχικό 0, τότε προκύπτει ο αριθμός 00001000 του δυαδικού συστήματος, που ισοδυναμεί με τον αριθμό 8 του δεκαδικού συστήματος. Άρα η ολίσθη-ση προς τα αριστερά ισοδυναμεί με πολλαπλασιασμό επί δύο, ενώ η ολίσθηση προς τα δεξιά ισοδυναμεί με την ακέραια διαίρεση διά δύο.**

**87 / 48**

**Στοιχεία ψευδογλώσσας**

**1. Σταθερές**

**Αριθμητικές: χρησιμοποιούνται οι αριθμητικοί χαρακτήρες, το +, το - και το κόμμα ως δεκαδικό σημείο,  
Αλφαριθμητικές: σχηματίζονται από οποιουσδήποτε χαρακτήρες εντός διπλών εισαγωγικών,  
Λογικές: υπάρχουν δύο,   
οι Αληθής και Ψευδής.**

**2. Μεταβλητές**

**Για τη σύνθεση του ονόματος μιας μεταβλητής χρησιμοποιούνται οι αριθμητικοί χαρακτήρες, οι αλφαβητικοί χαρακτήρες πεζοί και κεφαλαίοι, κα-θώς και ο χαρακτήρας \_ (underscore).**

**Οι μεταβλητές μπορούν επίσης να είναι αριθμητικές, αλφαριθμητικές και λογικές.**

**3. Τελεστές**

**Αριθμητικοί: + , - , \*, /, Λ, div, mod  
Συγκριτικοί: ≤, <, =, ≠, >, ≥  
Λογικοί: και (σύζευξη), ή (διάζευξη), όχι (άρνηση).**

**4. Εκφράσεις**

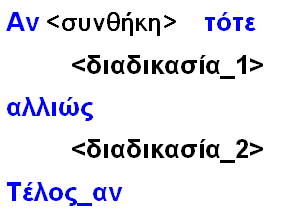
**Σχηματίζονται από σταθερές, μεταβλητές, συναρτή-σεις, τελεστές και παρενθέσεις**

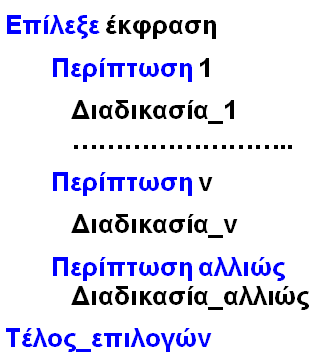
**5. Εντολή εκχώρησης**

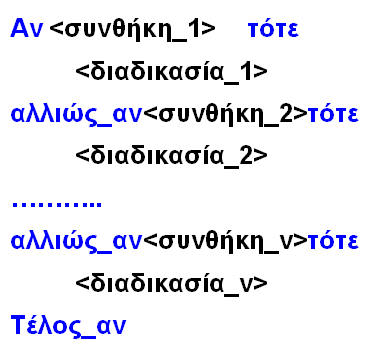
**Μεταβλητή 🡨 έκφραση**

**88 / 49**

**6. Σχήματα λογικών υποθέσεων**

****

****

****

**όπου ως διαδικασία λαμβάνεται ένα σύνολο εντολών**

**89 /49**

**7. Επαναληπτικές διαδικασίες.**

* **Επαναληπτικό σχήμα με έλεγχο επανάληψης στην αρχή  
  Όσο <συνθήκη> επανάλαβε   
  Διαδικασία   
  Τέλος\_επανάληψης**
* **Επαναληπτικό σχήμα με έλεγχο επανάληψης στο τέλος  
  Αρχή\_επανάληψης   
  Διαδικασία   
  Μέχρις\_ότου <συνθήκη>**
* **Επαναληπτικό σχήμα ορισμένων φορών επανάληψης  
  Για μεταβλητή από τ1 μέχρι τ2 με\_βήμα β   
  Διαδικασία   
  Τέλος\_επανάληψης**

**90 / 50**

**8. Ρήματα σε προστακτική**

**Για παράδειγμα, "Διάβασε", "Γράψε", "Εκτέλεσε" κ.λπ.**

**9. Ουσιαστικά**

**Σε ορισμένες περιπτώσεις όταν οι ζητούμενες ενέργειες είναι πολλές ή προφανείς, καθορίζονται με τη χρήση ουσιαστικών αντί ρημάτων, όπως "εισαγωγή δεδομένων", "εμφάνιση πεδίων στην οθόνη" κ.λπ.**

**10. Σχόλια**

**Προκειμένου να διαχωρίζονται οι επεξηγηματικές φράσεις από τις λέξεις-κλειδιά του αλγορίθμου, στις πρώτες προτάσσεται το σύμβολο !, για παράδειγμα !Σχόλια.**

**11. Πρώτη και τελευταία γραμμή**

**ενός αλγορίθμου είναι αντίστοιχα   
 Αλγόριθμος όνομα\_αλγορίθμου> και**

**Τέλος <όνομα\_αλγορίθμου>**

**12. Δεδομένα και αποτελέσματα**

**Τα δεδομένα εισόδου (αν υπάρχουν) περιγρά-φονται στη δεύτερη γραμμή του αλγορίθμου εντός των συμβόλων //... //.**

**Αντίστοιχα τα αποτελέσματα εξόδου δίνονται στην προτελευταία γραμμή του αλγορίθμου εντός των συμβόλων //... //**

**Η μέθοδος αυτή χρησιμοποιείται πρακτικά στους υπολογιστές, γιατί υλοποιείται πολύ πιο απλά απ' ότι ο γνωστός μας χειρωνακτικός τρόπος πολλαπλασια-σμού. Πιο συγκεκριμένα, απαιτεί πολλαπλασιασμό επί δύο, διαίρεση διά δύο και πρόσθεση. Σε αντίθεση η γνωστή μας διαδικασία πολλαπλασιασμού απαιτεί πολλαπλασιασμό με οποιοδήποτε ακέραιο και πρό-σθεση. Σε επίπεδο, λοιπόν, κυκλωμάτων υπολογιστή ο πολλαπλασιασμός επί δύο και η διαίρεση δια δύο μπορούν να υλοποιηθούν ταχύτατα με μία απλή εντολή ολίσθησης (shift), σε αντίθεση με τον πολλαπλασιασμό με οποιοδήποτε ακέραιο που θεωρείται πιο χρονοβόρα διαδικασία. Το τελευταίο γεγονός είναι ο λόγος που ο πολλαπλασιασμός αλά ρωσικά είναι προτιμότερος απ' ότι ο χειρωνακτικός τρόπος πολλαπλασιασμού δύο ακεραίων.**

**91 / 50-51**

**Στη συνέχεια παρουσιάζεται ο αλγόριθμος πολλαπλα-σιασμού ακεραίων αλά ρωσικά με φυσική γλώσσα κατά βήματα.**

|  |  |
| --- | --- |
| **Αλγόριθμος: Πολλαπλασιασμός δύο θετικών ακεραίων (αλά ρωσικά)** | |
| **Είσοδος:** | **Δύο ακέραιοι Μ1 και Μ2, όπου Μ1, Μ2 > 1** |
| **Έξοδος:** | **Το γινόμενο Ρ=Μ1\*Μ2** |
| **Βήμα 1** | **Θέσε Ρ=0** |
| **Βήμα 2** | **Αν Μ2>0, τότε πήγαινε στο Βήμα 3, αλλιώς πήγαινε στο Βήμα 7** |
| **Βήμα 3** | **Αν ο Μ2 είναι περιττός, τότε θέσε Ρ=Ρ+Μ1** |
| **Βήμα 4** | **Θέσε Μ1=Μ1\*2** |
| **Βήμα 5** | **Θέσε Μ2=Μ2/2 (θεώρησε μόνο το ακέραιο μέρος)** |
| **Βήμα 6** | **Πήγαινε στο Βήμα 2** |
| **Βήμα 7** | **Τύπωσε τον Ρ.** |

**Ακολουθεί ο αλγόριθμος σε ψευδοκώδικα για το ίδιο πρόβλημα του πολλαπλασιασμού αλά ρωσικά**

Αλγόριθμος **Πολλαπλασιασμός\_αλά\_ρωσικά**   
 **Δεδομένα // Μ1,Μ2 ακέραιοι //**

**P** 🡨 0

Όσο Μ2 >0 επανάλαβε  
Αν **Μ2** mod **2 = 1** τότε P🡨 P+Μ1

Μ1 🡨 Μ1 \* 2

Μ2 🡨 Μ2 div 2

Τέλος\_επανάληψης

Αποτελέσματα // P, το γινόμενο των ακεραίων Μ1,Μ2 //

Τέλος **Πολλαπλασιασμός\_αλά\_ρωσικά**

**92 / 51**

**• Ανακεφαλαίωση**

**Στο κεφάλαιο αυτό έγινε η πρώτη γνωριμία με την έννοια του αλγορίθμου. Δόθηκαν οι απαραίτητοι ορισμοί που συνοδεύτηκαν με αρκετά παραδείγματα. Αλγόριθμος είναι η διαδικασία της λύσης ενός προ-βλήματος. Η παράσταση των αλγορίθμων μπορεί να γίνει με αρκετούς τρόπους, ωστόσο η έμφαση δόθηκε στην παράσταση με χρήση ψευδογλώσσας. Στο κεφάλαιο αυτό αναπτύχθηκαν οι κύριες αλγοριθμικές δομές, δηλαδή η ακολουθία, η επιλογή και η επανά-ληψη ή ανακύκλωση, που θα χρησιμοποιηθούν στους αλγορίθμους των επόμενων κεφαλαίων.**

**Λέξεις κλειδιά**

**Αλγόριθμος, ακολουθία, επιλογή, διάγραμμα ροής, ψευδογλώσσα, εμφωλευμένος, βρόχος.**

**• Ερωτήσεις Θέματα για συζήτηση**

1. **Να δοθεί ο ορισμός του όρου αλγόριθμος.**
2. **Ποιά είναι τα κριτήρια που πρέπει να ικανοποιεί κάθε αλγόριθμος;**
3. **Υπό ποία πρίσματα η Πληροφορική επιστήμη μελετά τους αλγορίθμους;**

**93 / 51-52**

1. **Ποιά η διαφορά της θεωρητικής από την ανα-λυτική προσέγγιση στην επίλυση ενός προ-βλήματος με χρήση αλγορίθμου;**
2. **Περιγράψτε τους τρόπους περιγραφής και αναπαράστασης των αλγορίθμων.**
3. **Ποιοι είναι οι βασικοί τύποι συνιστωσών/εντολών ενός αλγορίθμου ;**

**94 / 52**

1. **Να περιγραφεί η δομή της ακολουθίας και να δοθεί σε διάγραμμα ροής ένα παράδειγμα αυτής της αλγοριθμικής προσέγγισης.**
2. **Να περιγραφεί η δομή της επιλογής και να δοθεί με ακολουθία βημάτων ένα παράδειγμα αυτής της αλγοριθμικής προσέγγισης.**
3. **Να περιγραφεί η δομή των επαναληπτικών διαδικασιών και να δοθεί με ακολουθία βημάτων και με διάγραμμα ροής ένα παράδειγμα αυτής της αλγοριθμικής προσέγγισης.**
4. **Να περιγραφεί η δομή των διαδικασιών πολλα-πλών επιλογών και να δοθεί με ακολουθία βημάτων και με διάγραμμα ροής ένα παράδειγμα αυτής της αλγοριθμικής προσέγγισης.**
5. **Να περιγραφεί η δομή των εμφωλευμένων διαδι-κασιών και να δοθεί με ακολουθία βημάτων και με διάγραμμα ροής ένα παράδειγμα αυτής της αλγοριθμικής προσέγγισης.**
6. **Να περιγραφεί με ακολουθία βημάτων το πρό-βλημα του 'πολλαπλασιασμού αλά ρωσικά'.**
7. **Ποιά η πρακτική σημασία του αλγορίθμου του 'πολλαπλασιασμού αλά ρωσικά'; Πότε γίνεται χρήση αυτού του τρόπου πολλαπλασιασμού   
   δύο ακεραίων;**

**• Βιβλιογραφία**

1. **Ν.Ιωαννίδης, Κ.Μαρινάκης, Σπ.Μπακογιάννης, Δομημένη Σχεδίαση Προγράμματος, Εκδόσεις Ελιξ, Αθήνα 1991.**
2. **Χρήστος Κοίλιας, Δομές Δεδομένων και Οργα-νώσεις Αρχείων, Εκδόσεις Νέων Τεχνολογιών, 1993, Αθήνα.**

**95 / 52**

1. **Ιωάννης Μανωλόπουλος, Δομές Δεδομένων - μία Προσέγγιση με Pascal, Εκδόσεις Art of Text, Θεσσαλονίκη, 1998.**
2. **Σκανδάλης κ.α. Στοιχεία Θεωρίας Αλγορίθμων, Πανεπιστημιακές Εκδόσεις Κρήτης, Κρήτη,   
   1990.**
3. **D. Brunskill and J. Turner, Understanding Algorithms and Data Structures, McGraw-Hill, 1996.**
4. **D. E. Knuth, The Art of Computer Programming: Fundamental Algorithms, Vol.1, 3rd edition,   
   Addison Wesley, 1997.**
5. **M.A. Weiss, Data Structures and Algorithm Analysis, 2nd edition, Benjamin/Cummings,   
   1995**

**Διαδίκτυο**

[**www.nist.gov/dads/**](http://www.nist.gov/dads/)

**Κόμβος με ευρετήριο όρων για αλγορίθμους, Δομές Δεδομένων και Προβλήματα (Algorithms, Data Structures, and Problems Terms and Definitions for the CRC Dictionary of Computer Science, Engineering and Technology)**

**96 / 52**

**Κεφάλαιο 3**

**Δομές δεδομένων και**

**Αλγόριθμοι**

**ΠΕΡΙΕΧΟΜΕΝΑ**

|  |  |
| --- | --- |
|  | **Δεδομένα** |
|  | **Αλγόριθμοι + Δομές δεδομένων =Προγράμματα** |
|  | **Πίνακες** |
|  | **Στοίβα και ουρά** |
|  | **Αναζήτηση και Ταξινόμηση** |
|  | **Αναδρομή** |
|  | **Άλλες δομές δεδομένων** |

**97 / 53**

**• Εισαγωγή**

**Εκτός από τους αλγορίθμους, σημαντική έννοια για την Πληροφορική είναι και η έννοια των "δεδομένων". Τα δεδομένα αποθηκεύονται στον υπολογιστή με τη βοήθεια των λεγόμενων "δομών δεδομένων". Θεω-ρώντας τους αλγόριθμους και τις δομές δεδομένων μία αδιάσπαστη ενότητα μπορεί να λεχθεί, ότι η ενότητα αυτή τελικά αποτελεί τη βάση ενός προγράμματος, που επιλύει ένα πρόβλημα. Στο κεφάλαιο αυτό γίνεται μία εισαγωγή στις σπουδαιότερες δομές δεδομένων και τις αντίστοιχες πράξεις που μπορούμε να κάνουμε με αυτές, όπως είναι η αναζήτηση, η εισαγωγή και η εξαγωγή στοιχείων, καθώς και η ταξινόμηση.**

**• Διδακτικοί στόχοι**

**Στόχοι του κεφαλαίου αυτού είναι οι μαθητές:**

* **να αιτιολογούν τη σπουδαιότητα των δεδομένων για την επίλυση ενός προβλήματος,**
* **να επισημαίνουν την αδιάσπαστη ενότητα αλγό-ριθμων και δομών δεδομένων,**
* **να εκτελούν γενικές ασκήσεις και ασκήσεις ανα-ζήτησης και ταξινόμησης με χρήση της δομής του πίνακα,**

**98 / 54**

* **να ορίζουν τις δομές της στοίβας και της ουράς με τις αντίστοιχες λειτουργίες,**
* **να ορίζουν την έννοια της αναδρομής και να εκτελούν απλές σχετικές ασκήσεις,**
* **να γνωρίζουν τις δομές της λίστας και του δένδρου.**

**• Προερωτήσεις**

* **Έχεις ακούσει για τον όρο FIFO;**
* **Γνωρίζεις ότι μπορεί να εξομοιωθεί στον υπολο-γιστή μια ουρά ανθρώπων, τρένων ή προγραμ-μάτων;**
* **υπάρχει νοηματική διαφορά ανάμεσα στους όρους δεδομένο και πληροφορία;**
* **όταν αναφερόμαστε σε προβλήματα, έμμεσα δη-λώνουμε την ανάγκη χρήσης υπολογιστών για την αντιμετώπισή τους;**
* **Ξέρεις, αν υπάρχουν πολλές μέθοδοι για να ταξινο-μηθούν κάποια αντικείμενα;**

**99 / 54**

|  |  |
| --- | --- |
| **3.1** | **Δεδομένα** |

**Τα δεδομένα (data) είναι η αφαιρετική αναπαράσταση της πραγματικότητας και συνεπώς μία απλοποιημένη όψη της. Για παράδειγμα, έστω ένα αρχείο μαθητών ενός σχολείου. Τα χρήσιμα δεδομένα που αποθη-κεύονται είναι το ονοματεπώνυμο, η ηλικία, το φύλο, η τάξη, το τμήμα κλπ., όχι όμως το βάρος, το ύψος κλπ. Τα δεδομένα, λοιπόν, είναι ακατέργαστα γεγονότα, και κάθε φορά η επιλογή τους εξαρτάται από τον τύπο του προβλήματος. Η συλλογή των ακατέργαστων δεδο-μένων και ο συσχετισμός τους δίνει ως αποτέλεσμα την πληροφορία (information). Δεν είναι εύκολο να δοθεί επακριβής ορισμός της έννοιας της πληρο-φορίας, αλλά μπορεί να θεωρηθεί ότι ο αλγόριθμος είναι το μέσο για την παραγωγή πληροφορίας από τα δεδομένα. Με βάση τις δεδομένες πληροφορίες λαμβά-νονται διάφορες αποφάσεις και γίνονται ενέργειες. Στη συνέχεια αυτές οι ενέργειες παράγουν νέα δεδομένα, νέες πληροφορίες, νέες αποφάσεις, νέες ενέργειες κοκ. Η μέτρηση, η κωδικοποίηση, η μετάδοση της πληρο-φορίας αποτελεί αντικείμενο μελέτης ενός ιδιαίτερου κλάδου, της Θεωρίας Πληροφοριών (Information Theory), που είναι ένα ιδιαίτερα σημαντικό πεδίο της Πληροφορικής.**

**Όπως η Πληροφορική ορίζεται ως επιστήμη σε συνάρτηση με την έννοια του αλγορίθμου, κατά τον ίδιο τρόπο η Πληροφορική ορίζεται και σε σχέση με την έννοια των δεδομένων. Έτσι, Πληροφορική θεωρείται η επιστήμη που μελετά τα δεδομένα από τις ακόλουθες σκοπιές:**

**100 / 55**

1. **Υλικού. Το υλικό (hardware), δηλαδή η μηχανή, επιτρέπει στα δεδομένα ενός προγράμματος να αποθηκεύονται στην κύρια μνήμη και στις περι-φερειακές συσκευές του υπολογιστή με διάφο-ρες αναπαραστάσεις (representations). Τέτοιες μορφές είναι η δυαδική, ο κώδικας ASCII (βλ. παράρτημα), ο κώδικας EBCDIC, το συμπλή-ρωμα του 1 ή του 2 κ.λπ.**
2. **Γλωσσών προγραμματισμού. Οι γλώσσες προγραμματισμού υψηλού επιπέδου (high level programming languages) επιτρέπουν τη χρήση διάφορων τύπων (types) μεταβλητών (variables) για να περιγράψουν ένα δεδομένο. Ο μεταφρα-στής κάθε γλώσσας φροντίζει για την αποδο-τικότερη μορφή αποθήκευσης, από πλευράς υλικού, κάθε μεταβλητής στον υπολογιστή.**
3. **Δομών Δεδομένων. Δομή δεδομένων (data structure) είναι ένα σύνολο δεδομένων μαζί με ένα σύνολο επιτρεπτών λειτουργιών επί αυτών. Για παράδειγμα, μία τέτοια δομή είναι η εγγραφή (record), που μπορεί να περιγράφει ένα είδος, ένα πρόσωπο κλπ. Η εγγραφή αποτελείται από τα πεδία (fields) που αποθηκεύουν χαρακτη-ριστικά (attributes) διαφορετικού τύπου, όπως για παράδειγμα ο κωδικός, η περιγραφή κλπ.**
4. **Ανάλυσης Δεδομένων. Τρόποι καταγραφής και αλληλοσυσχέτισης των δεδομένων μελετώνται έτσι ώστε να αναπαρασταθεί η γνώση για πραγ-ματικά γεγονότα. Οι τεχνολογίες των Βάσεων Δεδομένων (Databases), της Μοντελοποίησης Δεδομένων (Data Modeling) και της Αναπα-ράστασης Γνώσης (Knowledge Representation) ανήκουν σε αυτή τη σκοπιά μελέτης των δεδομένων.**

**101 / 55-56**

|  |  |
| --- | --- |
|  | **Μια θέση μνήμης (byte) έχει ως περιεχόμενο 11110001. Η τιμή μπορεί να παριστάνει:**  **102 / 55-56**   * **Το χαρακτήρα \_ στον κώδικα ASCII 437** * **Το χαρακτήρα ρ στον κώδικα ΕΛΟΤ 928** * **Το χαρακτήρα 1 στον κώδικα EBCDIC** * **Την τιμή 241 στο δυαδικό σύστημα (ως μη προσημασμένο ακέραιο)** * **Την τιμή -14 στο δυαδικό σύστημα (ως προσημασμένο ακέραιο στο συμπλήρωμα ως προς 1)** * **Την τιμή -15 στο δυαδικό σύστημα (ως προσημασμένο ακέραιο στο συμπλήρωμα ως προς 2)**   **Ακόμη μπορεί να είναι τμήμα ενός ακέραιου σε 2 ή 4 bytes, καθώς και ενός αριθμού κινητής υποδιαστολής.**  **Όσον αφορά στη φυσική σημασία της, αν μεν είναι χαρακτήρας, τότε αποτελεί μέρος μιας αλφαριθμητικής σταθεράς, ενώ αν πρόκειται για αριθμητική τιμή, τότε μπορεί να είναι δεδομένο, διεύθυνση μνήμης ή κώδικας εντολής προγράμματος.** |

|  |  |
| --- | --- |
| **3.2** | **Αλγόριθμοι+Δομές Δεδομένων=Προγράμματα** |

**Τα δεδομένα ενός προβλήματος αποθηκεύονται στον υπολογιστή, είτε στην κύρια μνήμη του είτε στη δευτε-ρεύουσα μνήμη του. Η αποθήκευση αυτή δεν γίνεται κατά ένα τυχαίο τρόπο αλλά συστηματικά, δηλαδή χρησιμοποιώντας μία δομή. Η έννοια της δομής δεδομένων (data structure) είναι σημαντική για την Πληροφορική και ορίζεται με τον ακόλουθο τυπικό ορισμό.**

**ΟΡΙΣΜΟΣ**

**Δομή Δεδομένων είναι ένα σύνολο αποθηκευμένων δεδομένων που υφίστανται επεξεργασία από ένα σύνολο λειτουργιών.**

**Κάθε μορφή δομής δεδομένων αποτελείται από ένα σύνολο κόμβων (nodes). Οι βασικές λειτουργίες (ή αλλιώς πράξεις) επί των δομών δεδομένων είναι οι ακόλουθες:**

* **Προσπέλαση (access), πρόσβαση σε ένα κόμβο με σκοπό να εξετασθεί ή να τροποποιηθεί το περιεχό-μενό του.**
* **Εισαγωγή (insertion), δηλαδή η προσθήκη νέων κόμβων σε μία υπάρχουσα δομή.**
* **Διαγραφή (deletion), που αποτελεί το αντίστροφο της εισαγωγής, δηλαδή ένας κόμβος αφαιρείται από μία δομή.**
* **Αναζήτηση (searching), κατά την οποία προσπε-λαύνονται οι κόμβοι μιας δομής, προκειμένου να εντοπιστούν ένας ή περισσότεροι που έχουν μια δεδομένη ιδιότητα.**

**103 / 56**

* **Ταξινόμηση (sorting), όπου οι κόμβοι μιας δομής διατάσσονται κατά αύξουσα ή φθίνουσα σειρά.**
* **Αντιγραφή (copying), κατά την οποία όλοι οι κόμβοι ή μερικοί από τους κόμβους μίας δομής αντιγράφονται σε μία άλλη δομή.**
* **Συγχώνευση (merging), κατά την οποία δύο ή περισσότερες δομές συνενώνονται σε μία ενιαία δομή.**
* **Διαχωρισμός (separation), που αποτελεί την αντίστροφη πράξη της συγχώνευσης.**

**Στην πράξη σπάνια χρησιμοποιούνται και οι οκτώ λειτουργίες για κάποια δομή. Συνηθέστατα παρα-τηρείται το φαινόμενο μία δομή δεδομένων να είναι αποδοτικότερη από μία άλλη δομή με κριτήριο κάποια λειτουργία, για παράδειγμα την αναζήτηση, αλλά λιγότερο αποδοτική για κάποια άλλη λειτουργία, για παράδειγμα την εισαγωγή. Αυτές οι παρατηρήσεις εξηγούν αφ' ενός την ύπαρξη διαφορετικών δομών, και αφ' ετέρου τη σπουδαιότητα της επιλογής της κατάλ-ληλης δομής κάθε φορά.**

**Στη συνέχεια του βιβλίου αυτού θα γίνει πληρέστερη παρουσίαση εναλλακτικών δομών δεδομένων. Ωστόσο, στο σημείο αυτό τονίζεται ότι υπάρχει μεγάλη εξάρτηση μεταξύ της δομής δεδομένων και του αλγό-ριθμου που επεξεργάζεται τη δομή. Μάλιστα, το πρό-γραμμα πρέπει να θεωρεί τη δομή δεδομένων και τον αλγόριθμο ως μία αδιάσπαστη ενότητα.**

**104 / 56-57**

**Η παρατήρηση αυτή δικαιολογεί την εξίσωση που διατυπώθηκε το 1976 από τον Wirth (που σχεδίασε και υλοποίησε τη γλώσσα Pascal)**

**Αλγόριθμοι + Δομές Δεδομένων = Προγράμματα**

**Ωστόσο για την πληρέστερη κατανόηση της σχέσης αυτής στη συνέχεια θα εξετασθεί ένα τέτοιο πρόβλημα.**

**ΠΑΡΑΔΕΙΓΜΑ**

**Έστω ότι πρέπει να γραφεί ένας αλγόριθμος που να δέχεται ως είσοδο το όνομα ενός συνδρομητή του ΟΤΕ και να δίνει ως έξοδο το τηλέφωνο του.**

**Πρώτη Λύση.**

**Δομή Δεδομένων: Δημιουργείται μία ακολουθία (Ο1,Τ1), (Ο2,Τ2), ..., (On,Τη), όπου οι μεταβλητές Οi, και Τi, αναφέρονται στο όνομα και στο τηλέφωνο του  
 i-οστού συνδρομητή, για i = 1,2,...,n.**

**Αλγόριθμος: Η ακολουθία ανιχνεύεται μέχρι να βρεθεί το ζητούμενο όνομα του συνδρομητή Ok και εκτυ-πώνεται το τηλέφωνο Tk. Ο αλγόριθμος αυτός είναι αποδοτικός για συνδρομητές ενός χωριού ή μίας κωμόπολης, αλλά για συνδρομητές μίας μεγάλης πόλης είναι χρονοβόρος.**

**Δεύτερη Λύση.**

**Δομή Δεδομένων: Χρησιμοποιείται και πάλι η ακο-λουθία της πρώτης λύσης, αλλά αυτή τη φορά οι συνδρομητές είναι ταξινομημένοι λεξικογραφικά. Επιπλέον δημιουργείται μία δεύτερη ακολουθία με τα στοιχεία (Α,n1),(Β,n2),..., (Ω,n24). Κάθε στοιχείο της δεύτερης αυτής ακολουθία δίνει για κάθε γράμμα του αλφαβήτου τη θέση ni για i = 1, 2, ..., 24) στην πρώτη ακολουθία με το πρώτο όνομα συνδρομητή που αρχίζει από το γράμμα αυτό.**

**105 / 57**

**Αλγόριθμος: Αφήνεται για άσκηση στο μαθητή.**

**Οι δομές δεδομένων διακρίνονται σε δύο μεγάλες κατηγορίες: τις στατικές (static) και τις δυναμικές (dynamic). Οι δυναμικές δομές δεν αποθηκεύονται σε συνεχόμενες θέσεις μνήμης αλλά στηρίζονται στην τεχνική της λεγόμενης δυναμικής παραχώρησης μνήμης (dynamic memory allocation). Με άλλα λόγια, οι δομές αυτές δεν έχουν σταθερό μέγεθος, αλλά ο αριθμός των κόμβων τους μεγαλώνει και μικραίνει καθώς στη δομή εισάγονται νέα δεδομένα ή δια-γράφονται κάποια δεδομένα αντίστοιχα. Όλες οι σύγχρονες γλώσσες προγραμματισμού προσφέρουν τη δυνατότητα δυναμικής παραχώρησης μνήμης. Ωστόσο, εμείς στη συνέχεια θα εξετάσουμε μόνο τις στατικές δομές που είναι ευκολότερες στην κατανόηση και την υλοποίησή τους**

|  |  |
| --- | --- |
| **3.3** | **Πίνακες** |

**Με τον όρο στατική δομή δεδομένων εννοείται ότι το ακριβές μέγεθος της απαιτούμενης κύριας μνήμης καθορίζεται κατά τη στιγμή του προγραμματισμού τους, και κατά συνέπεια κατά τη στιγμή της μετάφρα-σής τους και όχι κατά τη στιγμή της εκτέλεσης τους προγράμματος. Μία άλλη σημαντική διαφορά σε σχέση με τις δυναμικές δομές είναι ότι τα στοιχεία των στατικών δομών αποθηκεύονται σε συνεχόμενες θέσεις μνήμης.**

**106 / 57-58**

 **Στην πράξη, οι στατικές δομές υλοποιούνται με πίνακες που μας είναι γνωστοί από άλλα μαθήματα και υποστηρίζονται από κάθε γλώσ-σα προγραμματισμού. Μπορούμε να ορίσουμε τον πίνακα ως μια δομή που περιέχει στοιχεία του ίδιου τύπου (δηλαδή ακέραιους, πραγματικούς κ.λπ). Η δήλωση των στοιχείων ενός πίνακα και η μέθοδος αναφοράς τους εξαρτάται από τη συγκεκριμένη γλώσσα υψηλού επιπέδου που χρησι-μοποιείται. Όμως, γενικά η αναφορά στα στοιχεία ενός πίνακα γίνεται με τη χρήση του συμβολικού ονόματος του πίνακα ακολουθούμενου από την τιμή ενός ή πε-ρισσότερων δεικτών (indexes) σε παρένθεση ή αγκύ-λη.**

**Ένας πίνακας μπορεί να είναι μονοδιάστατος, αλλά στη γενικότερη περίπτωση μπορεί να είναι δισδιά-στατος, τρισδιάστατος και γενικά ν-διάστατος πίνακας. Όσον αφορά στους δισδιάστατους πίνακες σημει-ώνεται ότι αν το μέγεθος των δύο διαστάσεων είναι ίσο, τότε ο πίνακας λέγεται τετραγωνικός (square) και γενικά συμβολίζεται ως πίνακας n x n. Μάλιστα μπο-ρούμε να θεωρήσουμε το δισδιάστατο πίνακα ότι είναι ένας μονοδιάστατος πίνακας, όπου κάθε θέση του περιέχει ένα νέο μονοδιάστατο πίνακα. Στη συνέχεια δίνουμε δύο απλά παραδείγματα χρήσης πινάκων, τα οποία στηρίζονται σε αλγορίθμους του προηγουμένου κεφαλαίου.**

**107 / 58**

**Σχ. 3.1 Παραδείγματα πινάκων (μονοδιάστατος, δισδιάστατος, τρισδιάστατος)**

**108 / 58**

**ΠΑΡΑΔΕΙΓΜΑ 1. Εύρεση του μικρότερου στοιχείου ενός μονοδιάστατου πίνακα**

**Δίνεται ένας μονοδιάστατος πίνακα table 100 στοιχείων. Να σχεδιασθεί αλγόριθμος που να βρίσκει το μικρότερο στοιχείο του.**

Αλγόριθμος **Ελάχ\_Πίνακα   
Δεδομένα // table //**

**Min**  🡨 table[1]

Για i από 2 μέχρι 100

Αν table[i]< **Min** τότε **Min** 🡨 table[i]

Τέλος\_επανάληψης

Αποτελέσματα **//** Min **//**

Τέλος **Ελάχ\_Πίνακα**

**Στον αλγόριθμο αυτό αρχικά το πρώτο στοιχείο του πίνακα εκχωρείται στη μεταβλητή Min. Στη συνέχεια κάθε επόμενο στοιχείο του πίνακα εξετάζεται, αν είναι μικρότερο της Min και αν ναι, τότε αντικαθιστά το προηγούμενο. Έτσι στο τέλος θα υπάρχει στη μετα-βλητή Min το μικρότερο στοιχείο όλου του πίνακα table.**

**ΠΑΡΑΔΕΙΓΜΑ 2. Εύρεση αθροίσματος στοιχείων δισδιάστατου πίνακα**

**Δίνεται ο δισδιάστατος πίνακας table με m γραμμές n στήλες. Να βρεθεί το άθροισμα κατά γραμμή, κατά στήλη και συνολικά.**

**Στη συνέχεια ακολουθεί ο αλγόριθμος που επιλύει το πρόβλημα. Για καλύτερη κατανόηση σημειώνεται, ότι οι δύο πρώτοι βρόχοι μηδενίζουν τις αντίστοιχες μεταβλητές που θα υποδεχθούν τα αθροίσματα. Αυτό είναι μία τακτική που πρέπει να εφαρμόζεται οποτε-δήποτε στα προβλήματά μας έχουμε να υπολογίσουμε αθροίσματα.**

**109 / 58-59**

Αλγόριθμος **Αθρ\_Πίνακα   
Δεδομένα // m, n, table //**

**sum**  🡨 0

Για i από 1 μέχρι m

row[ i ]🡨 0

Τέλος\_επανάληψης

Για j από 1 μέχρι n

col[ j ]🡨 0

Τέλος\_επανάληψης

Για i από 1 μέχρι m

Για j από 1 μέχρι n

sum[ i ]🡨 sum + **table[ i,j ]**

row[ i ]🡨 row[ i ] **+ table[ i,j ]**

col [ j ]🡨 col[ j ] **+ table[ i,j ]**

Τέλος\_επανάληψης

Τέλος\_επανάληψης

Αποτελέσματα **//** row, col, sum **//**

Τέλος  **Αθρ\_Πίνακα**

**Ο διπλός εμφωλευμένος βρόχος που ακολουθεί τους δύο πρώτους απλούς βρόχους, είναι η καρδιά του αλγορίθμου, όπου γίνονται οι υπολογισμοί που ζητά η εκφώνηση του παραδείγματος. Γενικά σε εμφωλευ-μένους βρόχους, μία τιμή μεταβλητής του εξωτερικού βρόχου παραμένει σταθερή, όσο μεταβάλλεται η τιμή της μεταβλητής του εσωτερικού βρόχου. Πιο συγκε-κριμένα, στον αλγόριθμό μας αρχικά το i λαμβάνει την τιμή 1 και το j διαδοχικά τις τιμές 1,2,...,n. Κατόπιν, το i λαμβάνει την τιμή 2, ενώ το j και πάλι λαμβάνει δια-δοχικά τις τιμές 1,2,...,n. Η διαδικασία αυτή επανα-λαμβάνεται μέχρι το i να λάβει την τιμή m.**

**110 / 59**

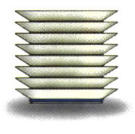
**Ο επόμενος πίνακας table είναι ένας δισδιάστατος πίνακας 5x5. Αν ο προηγούμενος αλγόριθμος εφαρμοσθεί στον πίνακα αυτό, τότε οι τιμές των στοιχείων του πίνακα row παρουσιάζονται στην τελευταία κατακόρυφη στήλη, ενώ οι τιμές των στοιχείων του πίνακα col παρουσιάζονται στην τελευταία γραμμή του πίνακα. Τέλος το συνολικό άθροισμα sum παρουσιάζεται στην κάτω-δεξιά γωνία.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Πίνακας table** | | | | |  | **Πίνακας row** | |
|  | **4** | **16** | **5** | **21** | **7** |  | **53** |  |
|  | **28** | **9** | **38** | **13** | **51** |  | **139** |  |
|  | **17** | **67** | **22** | **40** | **30** |  | **176** |  |
|  | **20** | **40** | **10** | **3** | **13** |  | **86** |  |
|  | **21** | **34** | **48** | **29** | **16** |  | **158** |  |
| **Πίνακας col** | | |  |  |  |  |  |  |
|  | **9** | **166** | **123** | **106** | **127** |  | **612** | **Sum** |

**Οι πίνακες χρησιμεύουν για την αποθήκευση και διαχείριση δύο σπουδαίων δομών, της στοίβας (stack) και της ουράς (queue), που θα εξετασθούν λεπτομε-ρέστερα στη συνέχεια, επειδή χρησιμοποιούνται σε πληθώρα πρακτικών εφαρμογών.**

**111 / 59-60**

|  |  |
| --- | --- |
| **3.4** | **Στοίβα** |

**Μία στοίβα δεδομένων μοιάζει με μία στοίβα από πιάτα. Για παράδειγμα, κάθε πιάτο που πλένεται τοποθετείται στην κορυφή (top) της στοίβας των πιάτων, ενώ για σκούπισμα λαμβάνεται και πάλι το πιάτο της κορυφής. Αντίστοιχα, τα δεδομένα που βρίσκονται στην κορυφή της στοίβας λαμβάνονται πρώτα, ενώ αυτά που βρίσκονται στο βάθος της στοίβας λαμβάνονται τελευταία. Αυτή η μέθοδος επεξεργασίας ονομάζεται Τελευταίο μέσα, Πρώτο έξω ή απλούστερα με την αγγλική συντομογραφία LIFO (Last-In-First-Out).**

**112 / 60**

|  |  |
| --- | --- |
| **C**  **B**  **A**  **ΕΙΣΕΡΧΟ-ΜΕΝΑ**  **ΣΤΟΙΧΕΙΑ**  **ΕΞΕΡΧΟΜΕ-ΝΑ**  **ΣΤΟΙΧΕΙΑ** | |
| **Μια στοίβα με τρία στοιχεία.** | |
| **C**  **B**  **A**  **D** | **C**  **B**  **A** |
| **Η στοίβα μετά την Εισαγωγή του  στοιχείου D (λειτουργία ΩΘΗΣΗ)** | **Η στοίβα μετά την Εξαγωγή του  στοιχείου D (λειτουργία ΑΠΩΘΗΣΗ)** |
| **Σχ. 3.2. Λειτουργίες στοίβας**  **113 / 60** | |

**Στα αριστερά του επόμενου σχήματος δίνεται μια στοίβα με τρία στοιχεία, ενώ στο κέντρο παρουσιάζεται η ίδια στοίβα με ένα πρόσθετο στοιχείο στην κορυφή της.**

**Δύο είναι οι κύριες λειτουργίες σε μία στοίβα:**

1. **η ώθηση (push) στοιχείου στην κορυφή της στοίβας, και**
2. **η απώθηση (pop) στοιχείου από τη στοίβα.**

**Η διαδικασία της ώθησης πρέπει οπωσδήποτε να ελέγχει, αν η στοίβα είναι γεμάτη, οπότε λέγεται ότι συμβαίνει υπερχείλιση (overflow) της στοίβας. Αντίστοιχα, η διαδικασία απώθησης ελέγχει, αν υπάρχει ένα τουλάχιστον στοιχείο στη στοίβα, δηλαδή ελέγχει αν γίνεται υποχείλιση (underflow) της στοίβας.**

**Μια στοίβα μπορεί να υλοποιηθεί πολύ εύκολα με τη βοήθεια ενός μονοδιάστατου πίνακα, όπως φαίνεται στο σχήμα 3.3. Μια βοηθητική μεταβλητή (με όνομα συ-νήθως top) χρησιμοποιείται για να δείχνει το στοιχείο που τοποθετήθηκε τελευταίο στην κορυφή της στοίβας. Για την εισαγωγή ενός νέου στοιχείου στη στοίβα (ώθηση) αρκεί να αυξηθεί η μεταβλητή top κατά ένα και στη θέση αυτή να εισέλθει το στοιχείο. Αντίθετα για την εξαγωγή ενός στοιχείου από τη στοίβα (απώθηση) εξέρχεται πρώτα το στοιχείο που δείχνει η μεταβλητή top και στη συνέχεια η top μειώνεται κατά ένα για να δείχνει τη νέα κορυφή.**

**114 / 60-61**

**Stack**

**n**

**n-1**

**1**

**2**

**3**

**4**

**Top**

**115 / 60**

**Σχ. 3.3 Υλοποίηση στοίβας με χρήση πίνακα**

|  |  |
| --- | --- |
| **3.5** | **Ουρά** |

**Οι ουρές είναι καθημερινό φαινόμενο. Για παράδειγμα, ουρές δημιουργούνται όταν άνθρωποι, αυτοκίνητα, εργασίες, προγράμματα κ.λπ. περιμένουν για να εξυπηρετη-θούν. Το θέμα είναι τόσο σημαντικό και με τέτοιες πρακτικές επιπτώσεις, ώστε ένας ιδιαίτερος κλάδος των Μαθημα-τικών, η Επιχειρησιακή Έρευνα (Operations Research), και ιδιαίτερα η θεωρία Ουρών (Queueing Theory), μελετά τη συμπεριφορά και την επίδοση των ουρών. Σε μία ουρά αναμονής με ανθρώ-πους, συμβαίνει να εξυπηρετείται εκείνος που στάθηκε στην ουρά πρώτος από όλους τους άλλους (αν κι υπάρχουν εξαιρέσεις που όμως δεν θα εξετασθούν στο βιβλίο αυτό).Η μέθοδος αυτή επεξεργασίας ονομάζεται Πρώτο μέσα, Πρώτο έξω ή απλούστερα ακολουθώ-ντας την αγγλική συντομογραφία FIFO (First-In-First-Out).**

**Δύο είναι οι κύριες λειτουργίες που εκτελούνται σε μία ουρά:**

1. **η εισαγωγή (enqueue) στοιχείου στο πίσω άκρο της ουράς, και**
2. **η εξαγωγή (dequeue) στοιχείου από το εμπρός άκρο της ουράς.**

**Άρα, σε αντίθεση με τη δομή της στοίβας, στην περίπτωση της ουράς απαιτούνται δύο δείκτες: ο εμπρός (front) και ο πίσω (rear) δείκτης, που μας δίνουν τη θέση του στοιχείου που σε πρώτη ευκαιρία θα εξαχθεί και τη θέση του στοιχείου που μόλις εισήλθε.**

**Στο σχήμα 3.4 φαίνεται μια ουρά με τέσσερα στοιχεία (α), στην οποία εισάγεται ένα νέο στοιχείο (β) και ακολούθως εξάγεται ένα στοιχείο.**

**116 / 61**

**Εμπρός (front)**

**Πίσω  
(rear)**

**5**

**12**

**7**

**4**

**⭠**

**⭠**

**(3.4.α) Μια ουρά με 4 στοιχεία**

**5**

**12**

**7**

**4**

**8**

**εμπρός (front)**

**Πίσω  
(rear)**

**(3.4.β) Η ουρά μετά την εισαγωγή του στοιχείου 8**

**(3.4.γ) Η ουρά μετά την εξαγωγή του στοιχείου 5**

**12**

**7**

**4**

**8**

**εμπρός (front)**

**Πίσω  
(rear)**

**Σχ. 3.4. Εισαγωγή και εξαγωγή από ουρά**

**Μια ουρά μπορεί να υλοποιηθεί με τη βοήθεια ενός μονοδιάστατου πίνακα, όπως φαίνεται στο σχ.3.5. Για την εισαγωγή ενός νέου στοιχείου στην ουρά αυξάνεται ο δείκτης rear κατά ένα και στη θέση αυτή αποθηκεύεται το στοιχείο. Αντίστοιχα για τη λειτουργία της εξαγωγής, εξέρχεται το στοιχείο που δείχνει ο δείκτης front, ο οποίος στη συνέχεια αυξάνεται κατά ένα, για να δείχνει το επόμενο στοιχείο που πρόκειται να εξαχθεί. Σε κάθε περίπτωση όμως, πρέπει να ελέγχεται πριν από οποιαδήποτε ενέργεια, αν υπάρχει ελεύθερος χώρος στον πίνακα για την εισαγωγή και αν υπάρχει ένα τουλάχιστον στοιχείο για την εξαγωγή.**

**117 / 61-62**

**118 / 62**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**10**

**n-1**

**Q**

**εμπρός (front)**

**Πίσω  
(rear)**

**Σχ. 3.5 Υλοποίηση ουράς με χρήση πίνακα**

**119 / 62**

**FIFO και LIFO**

**Όπως είδαμε η δομή της στοίβας λειτουργεί με τη με-θοδο FIFO. Οι δυο αυτές μέθοδοι έχουν αρκετές χρήσεις σε πραγματικά προβλήματα. Ας θεωρήσουμε για παράδειγμα τη\ περίπτωση ενός αποθηκευτικού χώρου μιας επιχείρησης. Σε κάθε αποθήκη γίνονται εισαγωγές ειδών που προέρχονται από αγορές από προμηθευτές, αν η επιχείρηση είναι εμπορική ή από την παραγωγή, αν πρόκειται για βιομηχανική επιχείρηση. Τα εμπο-ρεύματα ή προϊόντα τοποθετούνται σε κάποιους χώ-ρους, αποθήκες, ράφια κ.λπ. Όταν γίνονται πωλήσεις κάποιων ειδών, τα είδη αυτά βγαίνουν από την από-θήκη και αποστέλλονται στους πελάτες. Έτσι εισα-γωγές και εξαγωγές ειδών γίνονται συνεχώς στην αποθήκη ανάλογα με τη διαδικασία προμηθειών και τη ροή των πωλήσεων.**

**120 / 62**

**Σε μια δεδομένη στιγμή για κάποιο είδος μπορεί να υπάρχουν αποθηκευμένα κάποια τεμάχια που προ-έρχονται από μια παραλαβή και κάποια άλλα που υπήρχαν πιο πριν. Όταν πρέπει να εξαχθεί λοιπόν ένα τεμάχιο από αυτό το είδος, προκύπτει το πρόβλημα, από ποιά παρτίδα πρέπει να είναι;**

**Η απάντηση στο ερώτημα αυτό έχει φυσική και λογι-στική αξία. Αν το είδος αυτό δεν επηρεάζεται από το χρόνο, τότε ίσως δεν έχει μεγάλη σημασία η επιλογή. Αν όμως πρόκειται για είδος που μπορεί να αλλοιωθεί ή έχει ημερομηνία λήξης (π.χ. φάρμακα), τότε είναι φανερό ότι πρέπει να επιλεγεί το παλαιότερο. Στην περίπτωση αυτή λοιπόν πρέπει η εξαγωγή των ειδών να γίνεται με τη μέθοδο FIFO και συνήθως επαφίεται στον αποθηκάριο να κάνει τη σωστή επιλογή.**

**Εξ ίσου δύσκολο είναι το πρόβλημα αυτό από την οικονομική και λογιστική σκοπιά, που μάλιστα αφορά όλα τα είδη με ή χωρίς ημερομηνία λήξης. Ας υποθέ-σουμε ότι μια επιχείρηση έχει πραγματοποιήσει τις επόμενες αγορές και πωλήσεις για ένα είδος.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Αγορές** |  |  |  |
|  |  |  |  |
| **Ημ/νία** | **Ποσότητα** | **Τιμή μονάδας** | **Αξία** |
| **1/1/02** | **4** | **10** | **40** |
| **15/1/02** | **6** | **12** | **72** |
| **ΣΥΝΟΛΟ** | **10** |  | **112** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Πωλήσεις** |  |  |  |
|  |  |  |  |
| **Ημ/νία** | **Ποσότητα** | **Τιμή μονάδας** | **Αξία** |
| **30/1/02** | **5** | **20** | **100** |

**121 / 62-63**

**Από τα παραπάνω στοιχεία αγορών και πωλήσεων δημιουργείται η επόμενη καρτέλα είδους.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Καρτέλα είδους** | | | |  |  |  |  | |
|  |  |  |  |  |  |  |  | |
| **Ημ/νία** | **Αιτιολογία** | **Ποσότητα** | | | **Αξία κόστους** | | | |
| **Εισαγωγή** | **Εξαγωγή** | **Υπόλοιπο** | **Εισαγωγή** | **Εξαγωγή** | | **Υπόλοιπο** |
| **1/1/02** | **Αγορά** | **4** |  | **4** | **40** |  | | **40** |
| **15/1/02** | **Αγορά** | **6** |  | **10** | **72** |  | | **112** |
| **30/1/02** | **Πώληση** |  | **5** | **5** |  | **Χ** | | **Ψ** |

**122 / 63**

**Το πρόβλημα που ανακύπτει στις εφαρμογές αυτές είναι ο καθορισμός των τιμών Χ και Υ. Από τις τιμές αυτές εξάγεται στη συνέχεια το καθαρό κέρδος, με το οποίο η επιχείρηση θα φορολογηθεί.**

**α) Λειτουργία LIFO  
Στις 30/1/02 τα 5 τεμάχια που πουλήθηκαν θεωρού-νται ότι ανήκουν στα 6 τεμάχια της τελευταίας αγοράς, δηλαδή με τιμή μονάδας 12 €.   
Άρα Χ = 5x12 = 60 € και   
Υ = 112-52 = 60.   
Τώρα, το καθαρό κέρδος της πώλησης γίνεται   
100-60 = 40.**

**β) Λειτουργία FIFO  
Στις 30/1/02 από τα 5 τεμάχια που πουλήθηκαν, τα 4 είναι από την αγορά της 1/1/02 και το 1 από την αγορά της 15/1/02. Άρα το κόστος τους είναι   
Χ = 4x10 + 1x12 = 52 και   
Υ = 112-52 = 60.   
Τώρα, το καθαρό κέρδος της πώλησης γίνεται 100-52 = 48.**

**123 / 63**

**γ) Λειτουργία με τη σταθμική μέση τιμή   
Λόγω της αυξημένης πολυπλοκότητας των αντίστοι-χων προγραμμάτων, αλλά και των απαιτούμενων διαδικασιών οι περισσότερες επιχειρήσεις εφαρ-μόζουν τη μέθοδο της σταθμικής μέσης τιμής. Η τελευταία για το προηγούμενο παράδειγμα είναι 112/10 = 11,2. Άρα Χ = 5x11,2 = 56 και Υ = 112-56 = 56. Στην περίπτωση αυτή το καθαρό κέρδος γίνεται  
 100-56 = 44 €.**

|  |  |
| --- | --- |
| **3.6** | **Αναζήτηση** |

**Το πρόβλημα της αναζήτησης (searching) ενός στοι-χείου σε πίνακα είναι ιδιαίτερα ενδιαφέρον λόγω της χρησιμότητάς του σε πλήθος εφαρμογών. Υπάρχουν αρκετές μέθοδοι αναζήτησης σε πίνακα που εξαρτώνται κυρίως από το, αν ο πίνακας είναι ταξινομημένος ή όχι. Μια άλλη παράμετρος είναι, αν ο πίνακας περιέχει στοιχεία που είναι όλα διάφορα μεταξύ τους ή όχι. Τα στοιχεία του πίνακα μπορεί να είναι αριθμητικά ή αλφαριθμητικά.**

**Η πιο απλή μορφή αναζήτησης στοιχείου σε πίνακα είναι η σειριακή (sequential) ή γραμμική (linear) μέθοδος. Έτσι για τον επόμενο αλγόριθμο Sequential Search υποτίθεται ότι αναζητείται η τιμή key στον μη ταξινομημένο πίνακα table. Μετά την εκτέλεση του αλγορίθμου η μεταβλητή position επιστρέφει την τιμή 0, αν η αναζήτηση είναι ανεπιτυχής, ενώ αν η αναζήτηση είναι επιτυχής, τότε επιστρέφει τη θέση του στοιχείου στον πίνακα (δηλαδή, έναν αριθμό από 1 ως η).**

**124 / 63-64**

Αλγόριθμος **Sequential\_Search   
Δεδομένα // n, table, key //**

**done** 🡨 Ψευδής

**position** 🡨 0

i 🡨 1

Όσο **(done** = Ψευδής) και (i<=n) επανάλαβε

**Αν table [ i ] = key τότε   
 done** 🡨 Αληθής

**position** 🡨 i

αλλιώς

i 🡨 i + 1

Τέλος\_αν

Τέλος\_επανάληψης

Αποτελέσματα **//** done, position **//**

Τέλος  **Sequential\_Search**

**Όπως αναφέρθηκε, τα στοιχεία που περιέχονται στον πίνακα table δεν είναι ταξινομημένα. Επίσης, ο προ-ηγούμενος αλγόριθμος ισχύει για την περίπτωση όπου κάθε στοιχείο υπάρχει μία μόνο φορά στον πίνακα. Αν κάποιο στοιχείο εμφανίζεται στον πίνακα περισσότερο από μία φορές, τότε ο αλγόριθμος πρέπει να τροπο-ποιηθεί κατά το εξής: η μεταβλητή done είναι περιττή και η αναζήτηση συνεχίζεται μέχρι το τέλος του πίνακα ελέγχοντας με τη συνθήκη i ≤ n. Εξάλλου αν τα στοιχεία του πίνακα είναι ταξινομημένα, τότε ο αλγόριθμος πρέπει να σταματήσει, μόλις συναντήσει κάποιο στοιχείο που είναι μεγαλύτερο από το αναζητούμενο.**

**52**

**12**

**71**

**56**

**5**

**10**

**19**

**90**

**45**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**125 / 64**

**Για παράδειγμα, στο προηγούμενο σχήμα παρου-σιάζεται ένας πίνακας που περιέχει εννέα αταξινό-μητους ακεραίους. Έτσι, για την επιτυχή αναζήτηση της τιμής 56 απαιτούνται 4 προσπελάσεις. Αντίθετα, για την αναζήτηση της (ανύπαρκτης) τιμής 11 απαιτούνται 9 προσπελάσεις στον πίνακα, δηλαδή σάρωση ολό-κληρου του πίνακα.**

**Στο επόμενο σχήμα παρουσιάζεται ένας πίνακας που περιέχει τα ίδια στοιχεία αλλά σε ταξινομημένη μορφή. Στον πίνακα αυτό η ανεπιτυχής αναζήτηση για την τιμή 11 τερματίζει μετά την τρίτη προσπάθεια και την ανάγνωση του αριθμού 12.**

**5**

**10**

**12**

**19**

**45**

**52**

**56**

**71**

**90**

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**Η σειριακή μέθοδος αναζήτησης είναι η πιο απλή, αλλά και η λιγότερο αποτελεσματική μέθοδος αναζήτησης. Έτσι, δικαιολογείται η χρήση της μόνο σε περιπτώσεις όπου:**

* **ο πίνακας είναι μη ταξινομημένος,**
* **ο πίνακας είναι μικρού μεγέθους (για παράδειγμα, n ≤ 20 )**
* **η αναζήτηση σε ένα συγκεκριμένο πίνακα γίνεται σπάνια**

**Σε επόμενο κεφάλαιο θα εξετασθεί μία αποτελεσματικότερη μέθοδος αναζήτησης, η δυαδική αναζήτηση.**

**126 / 64-65**

**1η δοκιμή**

**2η δοκιμή**

**3η δοκιμή**

**4η δοκιμή**

**5η δοκιμή**

**Key ≠ table[1]**

**Key ≠ table[2]**

**Key ≠ table[3]11**

**Key ≠ table[4]n**

**Key ≠ 1.1.1**

**1.1.1**

**1.1.2**

**1.1.3**

**127 / 65**

**Σχ. 3.6. Σειριακή αναζήτηση**

|  |  |
| --- | --- |
| **3.7** | **Ταξινόμηση** |

**Η τακτοποίηση των κόμβων μίας δομής με μία ιδιαίτερη σειρά είναι μία πολύ σημαντική λειτουργία που ονομά-ζεται ταξινόμηση (sorting) ή διάταξη (ordering). Συνήθως η σειρά αυτή είναι η αύξουσα τάξη (ascending sequence) της τιμής των μεγεθών προς ταξινόμηση. Από το προ-ηγούμενο παράδειγμα έγινε σαφές ότι σκοπός της ταξινόμησης είναι να διευκολυνθεί στη συνέχεια η ανα-ζήτηση των στοιχείων του ταξινομημένου πίνακα. Η χρησιμότητα της ταξινόμησης αποδεικνύεται στην πράξη σε αναρίθμητες περιπτώσεις αναζήτησης αριθμητικών ή αλφαβητικών δεδομένων, όπως σε βιβλιοθηκονομικά συστήματα, λεξικά, τηλεφωνικούς καταλόγους, καταλόγους φόρου εισοδήματος και γενικά παντού όπου γίνεται αναζήτηση αποθηκευμένων αντι-κειμένων. Στη συνέχεια δίνεται ένας τυπικός ορισμός της ταξινόμησης.**

**128 / 65**

**ΟΡΙΣΜΟΣ**

**Δοθέντων των στοιχείων a1 ,a2 ,...,an η ταξινόμηση συνίσταται στη μετάθεση (permutation) της θέσης των στοιχείων, ώστε να τοποθετηθούν σε μία** **σειρά ak1,ak2,...,akn έτσι ώστε, δοθείσης μίας συνάρτησης διάταξης (ordering function), f, να ισχύει:  
f(ak1)≤ f(ak2)≤...≤f(akn).**

**Αξίζει να σημειωθεί ότι η προηγούμενη συνάρτηση διάταξης μπορεί να τροποποιηθεί, ώστε να καλύπτει και την περίπτωση που η ταξινόμηση γίνεται με φθίνουσα τάξη (descending sequence) μεγέθους.**

**Ταξινόμηση ευθείας ανταλλαγής**

**Η μέθοδος της ταξινόμησης ευθείας ανταλλαγής (straight exchange sort) βασίζεται στην αρχή της σύ-γκρισης και ανταλλαγής ζευγών γειτονικών στοιχείων, μέχρις ότου διαταχθούν όλα τα στοιχεία. Σύμφωνα με τη μέθοδο αυτή κάθε φορά γίνονται διαδοχικές προσπε-λάσεις στον πίνακα και μετακινείται το μικρότερο κλειδί της ακολουθίας προς το αριστερό άκρο του πίνακα. Αν ο πίνακας θεωρηθεί σε κατακόρυφη θέση αντί σε οριζόντια και οι ακέραιοι θεωρηθούν -επιστρατεύοντας αρκετή φαντασία- ως φυσαλίδες (bubbles) σε μία δεξαμενή νερού με βάρη σύμφωνα με την τιμή τους, τότε κάθε προσπέλαση στον πίνακα έχει ως από-τέλεσμα την άνοδο της φυσαλίδας στο κατάλληλο επίπεδο βάρους. Η μέθοδος είναι γνωστή ως ταξι-νόμηση φυσαλίδας (bubblesort).**

**129 / 65-66**

**Δομές Δεδομένων δευτερεύουσας μνήμης**

**Σε μεγάλες πρακτικές εμπορικές/ επιστημονικές εφαρμογές, το μέγεθος της κύριας μνήμης δεν επαρκεί για την αποθήκευση των δεδομένων. Στην περίπτωση αυτή χρησιμοποιούνται ειδικές δομές για την από-θήκευση των δεδομένων στη δευτερεύουσα μνήμη, δηλαδή κυρίως στο μαγνητικό δίσκο. Οι ειδικές αυτές δομές ονομάζονται αρχεία (files). Είναι γνωστό ότι μία σημαντική διαφορά μεταξύ κύριας μνήμης και μαγνη-τικού δίσκου είναι ότι στην περίπτωση του δίσκου, τα δεδομένα δεν χάνονται, αν διακοπεί η ηλεκτρική παροχή. Έτσι, τα δεδομένα των αρχείων διατηρούνται ακόμη και μετά τον τερματισμό ενός προγράμματος, κάτι που δεν συμβαίνει στην περίπτωση των δομών της κύριας μνήμης, όπως είναι οι πίνακες, όπου τα δεδομένα χάνονται όταν τελειώσει το πρόγραμμα. Τα στοιχεία ενός αρχείου ονομάζονται εγγραφές (records), όπου κάθε εγγραφή αποτελείται από ένα ή περισσότερα πεδία (fields), που ταυτοποιούν την εγγραφή, και από άλλα πεδία που περιγράφουν διάφορα χαρακτηριστικά της εγγραφής. Για παρά-δειγμα, έστω η εγγραφή ενός μαθητή με πεδία: Αριθμός Μητρώου, Ονοματεπώνυμο, Έτος Γέννησης, Τάξη, Τμήμα. Το πεδίο Αριθμός Μητρώου ταυτοποιεί την εγγραφή και ονομάζεται πρωτεύον κλειδί (primary key) ή απλά κλειδί. Το πεδίο Ονοματεπώνυμο επίσης ταυτοποιεί την εγγραφή και γι' αυτό αποκαλείται δευτερεύον κλειδί (secondary keys), αν υπάρχει πρωτεύον κλειδί. Το πρόβλημα της αναζήτησης (searching) μίας εγγραφής με βάση την τιμή του πρωτεύοντος ή ενός δευτερεύοντος κλειδιού σε αρχεία είναι ιδιαίτερα ενδιαφέρον, αν ληφθεί υπ' όψη η μεγάλη ποικιλία των χαρακτηριστικών τόσο της δομής (για**

**130 / 66**

**παράδειγμα, στατική ή δυναμική, τρόπος οργάνωσης, μέσο αποθήκευσης κ.λπ.), του τύπου των δεδομένων (για παράδειγμα, ακέραιοι, κείμενο, χαρτογραφικά δεδομένα, χρονοσειρές κ.λπ), όσο και της αναζήτησης (δηλαδή, με βάση το πρωτεύον ή το δευτερεύον κλειδί, κ.λπ.) .**

**ΠΑΡΑΔΕΙΓΜΑ.**

**Έστω ότι ο αρχικός πίνακας αποτελείται από εννέα κλειδιά τα εξής: 52, 12, 71, 56, 5, 10, 19, 90 και 45.   
Η μέθοδος εφαρμοζόμενη σε αυτά τα εννέα κλειδιά εξελίσσεται όπως φαίνεται στο επόμενο σχήμα. Κάθε φορά το ταξινομημένο τμήμα του πίνακα εμφανίζεται με χρώμα, ενώ τα στοιχεία που σαν φυσαλίδες ανέρχονται μέσα στον πίνακα εντοπίζονται με το αντίστοιχο βέλος στα δεξιά τους. Κάθε φορά εμφανίζεται η τάξη της επανάληψης (i).**

**131 / 66**



**Σχ. 3.7. Ταξινόμηση φυσαλίδας.**

**000 / 000**

**132 / 67**

**Η ταξινόμηση φυσαλίδας υλοποιείται με τον επόμενο αλγόριθμο**..

Αλγόριθμος **Φυσαλίδα   
Δεδομένα // table, n //**

Για i **από 2** μέχρι n

Για j **από n** μέχρι i με\_βήμα -1

**Αν table [ j-1 ] > table [ j ] τότε**

αντιμετάθεσε **table [ j-1 ], table [ j ]**

Τέλος\_αν

Τέλος\_επανάληψης

Τέλος\_επανάληψης

Αποτελέσματα **//** **table //**

Τέλος  **Φυσαλίδα**

|  |  |
| --- | --- |
|  | **Για την ταξινόμηση δεδομένων έχουν εκπονηθεί πάρα πολλοί αλγόριθμοι. Άλλοι σχετικά απλοί αλγόριθμοι είναι η ταξινόμηση με επιλογή και η ταξινόμηση με παρεμβολή. Ο πιο γρήγορος αλγό-ριθμος ταξινόμησης είναι η "γρήγορη ταξινόμηση" (quicksort). Η ταξινόμηση φυσαλίδας είναι ο πιο απλός και ταυτό-χρονα ο πιο αργός αλγόριθμος ταξινό-μησης.**  **133 / 67** |

**Στον αλγόριθμο αυτό ως είσοδος δίνεται η μεταβλητή table με n ακεραίους που πρέπει να ταξινομηθούν. Φυσικά η επιλογή του ακέραιου τύπου για το κλειδί είναι αυθαίρετη, αφού μπορεί να χρησιμοποιηθεί οποιοσδήποτε άλλος τύπος, όπου ορίζεται μία συνάρτηση διάταξης, όπως για παράδειγμα ο τύπος του χαρακτήρα.**

**Σημειώνεται ότι η εντολή "αντιμετάθεσε table[ j-1 ], table[ j ]" ανταλλάσσει το περιεχόμενο δύο θέσεων με τη βοήθεια μίας βοηθητικής θέσης. Εναλλακτικά αυτό μπορεί να γίνει με τις εξής τρεις εντολές:**

**temp** 🡨 **table [ j-1 ]**

**table [ j-1 ]** 🡨 **table [ j ]**

**table [ j ]** 🡨 **temp**

|  |  |
| --- | --- |
| **3.8** | **Αναδρομή** |

**Στο κεφάλαιο αυτό θα εξετάσουμε την έννοια της αναδρομής (recursion), που είναι μία σπουδαία εφαρμογή των στοιβών που εξετάσθηκαν προηγου-μένως. Η τεχνική της αναδρομής χρησιμοποιείται ευρύτατα τόσο από το λογισμικό συστήματος όσο και στο λογισμικό εφαρμογών. Πιο συγκεκριμένα, η αναδρομή στηρίζεται στη δυνατότητα που προσφέρεται από όλες τις σύγχρονες γλώσσες προγραμματισμού, μία διαδικασία ή συνάρτηση να καλεί τον εαυτό της.**

**134 / 67-68**

**3.8.1 Υπολογισμός του παραγοντικού**

**Η έννοια του παραγοντικού είναι γνωστή από τα μαθη-ματικά. Πιο συγκεκριμένα, για έναν ακέραιο n, το η παραγοντικό, που συμβολίζεται με n!, ορίζεται από τη σχέση:**

****

**Ισοδύναμος είναι ο εξής ορισμός**

****

**Ο ορισμός αυτός διακρίνεται από το βασικό χαρακτη-ριστικό ότι το παραγοντικό εκφράζεται με βάση μία απλούστερη περίπτωση του εαυτού του. Δηλαδή εξ ορισμού το παραγοντικό είναι μία αναδρομική αλγεβρική συνάρτηση. Με τον επόμενο αλγόριθμο υπολογίζεται με πολύ απλό αναδρομικό τρόπο το n παραγοντικό. Με τον επόμενο αλγόριθμο υπολογίζεται με πολύ απλό αναδρομικό τρόπο το n παραγοντικό.**

**135 / 68**

Αλγόριθμος **Παραγοντικό   
Δεδομένα // n //**

**Αν n = 0 τότε   
 product** 🡨 1

αλλιώς

**product** 🡨 **n \* Παραγοντικό (n-1)**

Τέλος\_αν

Αποτελέσματα **//** **product //**

Τέλος  **Παραγοντικό**

**Ωστόσο, το n! μπορεί να υπολογισθεί και με επανα-ληπτική μέθοδο, όπως παρουσιάζεται στον επόμενο αλγόριθμο Παραγοντικό2.**

Αλγόριθμος **Παραγοντικό2   
Δεδομένα // n //**

**product** 🡨 1

Για **i από** 2 **μέχρι** n

**product** 🡨 **product \* i**

Τέλος\_επανάληψης

Αποτελέσματα **//** **product //**

Τέλος  **Παραγοντικό2**

**Επομένως, δοθέντων αυτών των δυο λύσεων στο πρόβλημα υπολογισμού του n παραγοντικού προκύπτει το εύλογο ερώτημα, ποια μέθοδος είναι καλύτερη. Αφού εξετάσουμε μερικά ακόμη παρα-δείγματα, στη συνέχεια θα δοθεί απάντηση στην ερώτηση αυτή.**

**3.8.2 Υπολογισμός του μέγιστου κοινού διαιρέτη**

**Ένα ιστορικό πρόβλημα είναι η εύρεση του μέγιστου κοινού διαιρέτη (μκδ) δύο ακεραίων αριθμών. Ο αλγόριθμος εύρεσης του μκδ ανήκει στον Ευκλείδη. Ωστόσο η υλοποίηση και αυτού του αλγορίθμου μπορεί να γίνει κατά πολλούς τρόπους. Στη συνέχεια δίνεται ένας πρώτος αλγόριθμος για τον υπολογισμό του μκδ δύο ακεραίων x και y. Η μέθοδος αυτή είναι αρκετά αργή (και δεν στηρίζεται στον αλγόριθμο του Ευκλείδη), αλλά απλώς δίνεται για να διαφανεί η βελτίωση στην επίδοση από τους επόμενους αλγορίθμους. Ουσιαστικά λαμβάνει το μικρότερο από τους δύο ακεραίους, τον z, και εξετάζει με τη σειρά όλους τους ακεραίους ξεκι-νώντας από τον z και μειώνοντας συνεχώς κατά μία μονάδα μέχρι και οι δύο αριθμοί, x και y, να διαιρούνται από τη νέα τιμή του z.**

Αλγόριθμος  **Μέγιστος\_Κοινός\_Διαιρέτης   
Δεδομένα // x, y //**

**Αν x < y τότε   
 z** 🡨 x

αλλιώς

**z** 🡨 y

Τέλος\_αν

Όσο **(x mod** z ≠ 0) **ή (y mod** z ≠ 0) επανάλαβε

**z** 🡨 **z – 1**

**136 / 68-69**

Τέλος\_επανάληψης

**137 / 68-69**

Αποτελέσματα **//** **z //**

Τέλος  **Μέγιστος\_Κοινός\_Διαιρέτης**

**Η επόμενη μέθοδος για την εύρεση του μκδ αποδίδεται στον Ευκλείδη, και προφανώς βελτιώνει την προηγού-μενη εκδοχή, επειδή δεν εξετάζει με τη σειρά όλους τους ακεραίους.**

Αλγόριθμος  **Ευκλείδης   
Δεδομένα // x, y //**

**Αν x < y τότε   
 z** 🡨 y

Όσο **(**z ≠ 0) επανάλαβε

**z** 🡨 **x mod** y

**x** 🡨 y

**y** 🡨 z

Τέλος\_επανάληψης

Αποτελέσματα **//** x **//**

Τέλος  **Ευκλείδης**

**ΠΑΡΑΔΕΙΓΜΑ**

**Για καλύτερη κατανόηση της μεθόδου, ας παρακο-λουθήσουμε πώς ο αλγόριθμος αυτός βρίσκει το μκδ των αριθμών 150 και 35 . Ο επόμενος πίνακας δείχνει τις τιμές των μεταβλητών z, x και y, κατά τη διάρκεια των επαναλήψεων. Δηλαδή, πριν την έναρξη της επαναληπτικής δομής οι αρχικές τιμές των x και y είναι 150 και 35 (όπως φαίνεται στη δεύτερη γραμμή του πίνακα). Ο αλγόριθμος σταματά όταν γίνει 0 η τιμή του z, οπότε ο μκδ είναι η τιμή του x, δηλαδή το 5.**

|  |  |  |
| --- | --- | --- |
| **z** | **x** | **y** |
| **35** | **150** | **35** |
| **10** | **35** | **10** |
| **5** | **10** | **5** |
| **0** | **5** | **0** |

**Ωστόσο η μέθοδος του Ευκλείδη μπορεί να υλοποιηθεί και με έναν εναλλακτικό αναδρομικό τρόπο, που δίνεται στη συνέχεια. Η τρίτη αυτή εκδοχή είναι πολύ απλή στον προγραμματισμό και την κατανόησή της.**

Αλγόριθμος  **Ευκλείδης   
Δεδομένα // x, y //**

**Αν y = 0 τότε   
 z** 🡨 **x**

**αλλιώς**

**z** 🡨 **Ευκλείδης ( y, x mod** y )

Τέλος\_αν

Αποτελέσματα **//** z **//**

Τέλος  **Ευκλείδης**

**3.8.3 Υπολογισμός αριθμών ακολουθίας Fibonacci**

**Για καλύτερη κατανόηση της διαφοράς μεταξύ επανα-ληπτικών και αναδρομικών μεθόδων, στη συνέχεια θα εξετασθεί ένα τελευταίο παράδειγμα, όπου υπολογίζεται η ακολουθία αριθμών Fibonacci πρώτης τάξης, που ορίζεται ως εξής:**

**138 / 69-70**



**ενώ οι πρώτοι όροι της ακολουθίας Fibonacci πρώτης τάξης είναι:**

**0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 κλπ.**

**Από τον ορισμό φαίνεται ανάγλυφα η αναδρομική φύση της συνάρτησης. Οι δύο επόμενοι αλγόριθμοι υπο-λογίζουν τον αριθμό Fibonacci πρώτης τάξης Fn με επαναληπτική και αναδρομική μέθοδο. Υποτίθεται ότι κατά την κλήση του αλγορίθμου μία μη αρνητική τιμή περνά ως όρισμα στη μεταβλητή n.**

**139 / 70**

Αλγόριθμος **Fibonacci1   
Δεδομένα // n //**

**Αν n ≤ 1 τότε Fib** 🡨 **n**

**f0** 🡨 0

f1 🡨 1

Για **i από** 2 **μέχρι** n

**Fib** 🡨 **f0 + f1**

**f0** 🡨 f1

f1 🡨 Fib

Τέλος\_επανάληψης

Αποτελέσματα **//** **Fib //**

Τέλος  **Fibonacci1**

Αλγόριθμος **Fibonacci2   
Δεδομένα // n //**

**Αν n ≤ 1 τότε**

**Fib** 🡨 **n**

αλλιώς

**Fib** 🡨 **Fibonacci2(n-1)+Fibonacci2(n-2)**

Τέλος\_Αν

Αποτελέσματα **//** **Fib //**

Τέλος  **Fibonacci2**

**Συχνά η χρήση αναδρομής διευκολύνει τον προγραμ-ματιστή στην υλοποίηση και τον έλεγχο του προγράμ-ματος. Αν και πολλές φορές η αναδρομή φαίνεται ως πιο φυσικός τρόπος προγραμματισμού, ωστόσο πρέπει να χρησιμοποιείται με μέτρο. Μεταξύ ενός απλού επαναληπτικού προγράμματος και ενός αναδρομικού προγράμματος προτιμάται το πρώτο. Ο λόγος είναι ότι κάθε κλήση μίας συνάρτησης ή μίας διαδικασίας έχει χρονικό κόστος μη αμελητέο. Έτσι το κέρδος σε χρόνο προγραμματισμού δημιουργεί απώλεια σε χρόνο εκτέλεσης. Επομένως, αν ο χρόνος απόκρισης είναι κρίσιμος για την εφαρμογή μας, τότε είναι βέβαιο ότι πρέπει να προτιμηθεί η επαναληπτική μέθοδος.**

**Οι προηγούμενοι αλγόριθμοι για τον υπολογισμό των αριθμών Fibonacci πρώτης τάξης εκτός της διαφοράς τους στην επίδοση λόγω του ότι, η πρώτη είναι επαναληπτική ενώ η δεύτερη είναι αναδρομική, έχουν και μία ακόμη σημαντική διαφορά. Η διαφορά αυτή έγκειται στο γεγονός ότι, η δεύτερη καλεί περισσότερο από μία φορά τον εαυτό της για τις ίδιες τιμές. Αυτό θα γίνει κατανοητό δοκιμάζοντας για παράδειγμα τον υπολογισμό του F5. Σε επόμενο κεφάλαιο θα επα-νέλθουμε στη μελέτη των αλγορίθμων αυτών για τον υπολογισμό των αριθμών Fibonacci πρώτης τάξης.**

**140 / 70-71**

|  |  |
| --- | --- |
| **3.9** | **Άλλες δομές δεδομένων** |

**Κοινό γνώρισμα των δομών που εξετάσθηκαν προ-ηγουμένως είναι ότι οι διαδοχικοί κόμβοι αποθη-κεύονται σε συνεχόμενες θέσεις της κύριας μνήμης. Στην παράγραφο αυτή γίνεται μια παρουσίαση τριών πολύ σπουδαίων δομών δεδομένων, στις οποίες οι κόμβοι δεν είναι απαραίτητο να κατέχουν συνεχόμενες θέσεις μνήμης. Πρόκειται για τις λίστες, τα δένδρα και τους γράφους.**

**3.9.1 Λίστες**

**Στις λίστες το κύριο χαρακτηριστικό είναι ότι οι κόμβοι τους συνήθως βρίσκονται σε απομακρυσμένες θέσεις μνήμης και η σύνδεσή τους γίνεται με δείκτες. Ο δείκτης (pointer) είναι ένας ιδιαίτερος τύπος που προσφέρεται από τις περισσότερες σύγχρονες γλώσσες προ-γραμματισμού. Ο δείκτης δεν λαμβάνει αριθμητικές τιμές όπως ακέραιες, πραγματικές κ.α., αλλά οι τιμές του είναι διευθύνσεις στην κύρια μνήμη και χρησι-μοποιείται ακριβώς για τη σύνδεση των διαφόρων στοιχείων μιας δομής, που είναι αποθηκευμένα σε μη συνεχόμενες θέσεις μνήμης. Συνήθως ο δείκτης είναι ένα πεδίο κάθε κόμβου της δομής, όπως φαίνεται στο σχήμα 3.8. Το πεδίο Δεδομένα μπορεί να περιέχει μία ή περισσότερες αλφαριθμητικές ή αριθμητικές πληροφορίες.**

|  |  |
| --- | --- |
| **Δεδομένα** | **Δείκτης** |

**Σχ. 3.8 Δομή κόμβου λίστας**

**141 / 71**

**Στο σχήμα 3.9 παρουσιάζεται μια λίστα με τέσσερις κόμβους, όπου οι δείκτες έχουν τη μορφή βέλους, προ-κειμένου να φαίνεται ο κόμβος που παραπέμπουν.**

**Σχ. 3.9. Μία λίστα με τέσσερις κόμβους**

|  |  |
| --- | --- |
| πιν.jpg | **Οι όροι index και pointer αποδίδονται στα ελληνικά ως δείκτης. Και οι δύο παραπέμπουν σε θέσεις, πίνακα ο πρώτος και μνήμης ο δεύτερος.** |

**Με τη χρήση δεικτών διευκολύνονται οι λειτουργίες της εισαγωγής και της διαγραφής δεδομένων στις λίστες. Στο σχήμα 3.10 φαίνεται η εισαγωγή ενός νέου κόμβου μεταξύ του δεύτερου και τρίτου κόμβου της προηγούμενης λίστας.**

**Σχ. 3.10. Εισαγωγή σε λίστα**

**142 / 71-72**

**Όπως φαίνεται και στο σχήμα, οι απαιτούμενες ενέργειες για την εισαγωγή (παρεμβολή) του νέου κόμβου είναι ο δείκτης του δεύτερου κόμβου να δείχνει τον τρίτο κόμβο (δηλαδή να πάρει την τιμή που είχε πριν την εισαγωγή ο δείκτης του δεύτερου κόμβου). Έτσι οι κόμβοι της λίστας διατηρούν τη λογική τους σειρά, αλλά οι φυσικές θέσεις στη μνήμη μπορεί να είναι τελείως διαφορετικές.**

**143 / 72**

|  |  |
| --- | --- |
|  | **Οι δομές δεδομένων που χρησιμοποιούν δείκτες, αποκαλούνται δυναμικές (dynamic), γιατί η υλοποίησή τους γίνεται έτσι, ώστε να μην απαιτείται εκ των προτέρων καθορισμός του μέγιστου αριθμού κόμβων. Είναι φανερό, ότι οι δομές αυτές είναι πιο ευέλικτες από τη στατική δομή του πίνακα, επειδή επεκτείνονται και συρρικνώνονται κατά τη διάρκεια εκτέλεσης του προγράμματος.** |

**Σχ. 3.11. Διαγραφή κόμβου λίστας**

**Αντίστοιχα για τη διαγραφή ενός κόμβου αρκεί ν' αλλάξει τιμή ο δείκτης του προηγούμενου κόμβου και να δείχνει πλέον τον επόμενου αυτού που διαγράφεται, όπως φαίνεται στο σχήμα 3.11. Ο κόμβος που διαγρά-φηκε (ο τρίτος) αποτελεί "άχρηστο δεδομένο" και ο χώρος μνήμης που καταλάμβανε, παραχωρείται για άλλη χρήση.**

**3.9.2 Δένδρα**

**Τα δένδρα (trees) είναι δομές που στις σύγχρονες γλώσσες προγραμματισμού υλοποιούνται με τη βοή-θεια των δεικτών, όπως εξηγήθηκε στην αρχή αυτής της παραγράφου. Βέβαια, μπορούν να υλοποιηθούν και με στατικές δομές (με πίνακες). Το κύριο χαρακτηριστικό των δένδρων είναι, ότι από ένα κόμβο δεν υπάρχει ένας μόνο επόμενος κόμβος, αλλά περισσότεροι. Υπάρχει ένας μόνο κόμβος, που λέγεται ρίζα, από τον οποίο ξεκινούν όλοι οι άλλοι κόμβοι. Στο σχήμα 3.12. παρατηρούμε ότι από τη ρίζα ξεκινούν δύο κόμβοι. Οι κόμβοι αυτοί λέγονται παιδιά της ρίζας. Με την ίδια λογική, από κάθε παιδί της ρίζας ξεκινούν άλλα παιδιά κ.ο.κ. Στη βιβλιογραφία αναφέρεται μία τεράστια ποικιλία δομών δένδρων, που η αναφορά σε αυτές βρίσκεται εκτός των ορίων του βιβλίου αυτού.**

**144 / 72**

**Α**

**Β**

**Γ**

**Ε**

**Δ**

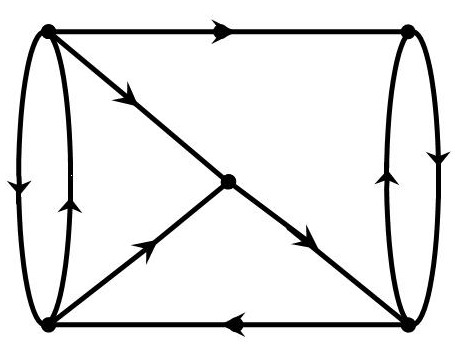
**Ζ**

**Η**

**Σχ. 3.12. Δομή δένδρου**

**3.9.3 Γράφοι**

**Ένας γράφος (graph) αποτελείται από ένα σύνολο κόμβων (ή σημείων ή κορυφών) και ένα σύνολο γραμμών (ή ακμών ή τόξων) που ενώνουν μερικούς ή όλους τους κόμβους. Ο γράφος αποτελεί την πιο γενική δομή δεδομένων, με την έννοια ότι όλες οι προηγού-μενες δομές που παρουσιάστηκαν μπορούν να θεωρη-θούν περιπτώσεις γράφων.**

****

**Σχ. 3.13. Ένας γράφος**

**Πολλά προβλήματα και καταστάσεις της καθημερινής μας ζωής μπορούν να περιγραφούν με τη βοήθεια γράφων. Για παράδειγμα τα σημεία ενός γράφου μπορούν να παριστούν πόλεις και οι γραμμές τις οδικές συνδέσεις μεταξύ τους. Λόγω της μεγάλης πληθώρας και ποικιλίας των προβλημάτων που σχετίζονται με γράφους, έχει αναπτυχθεί ομώνυμη θεωρία, η Θεωρία Γράφων, η οποίο συχνά αποτελεί αυτοδύναμο μάθημα σε πανεπιστημιακά τμήματα.**

**145 / 73**

**• Ανακεφαλαίωση**

**Στο κεφάλαιο αυτό αρχικά ορίσθηκε η Πληροφορική ως η επιστήμη που μελετά τα δεδομένα από τις σκοπιές του υλικού, των γλωσσών προγραμματισμού, των δομών δεδομένων και της ανάλυσης δεδομένων. Δόθηκε ο ορισμός της δομής δεδομένων και ένας κατάλογος με τις λειτουργίες που μπορούν να γίνουν με μία δομή δεδομένων. Η πρώτη δομή που εξετάσθηκε ήταν η δομή του πίνακα (μονοδιάστατου και δισδιά-στατου), που είναι μία στατική δομή, με μέγεθος που δεν μεταβάλλεται χρονικά. Στη συνέχεια παρου-σιάσθηκε η δομή της στοίβας, καθώς και των δύο βασικών πράξεων της ώθησης και της απώθησης των στοιχείων της. Επίσης περιγράφηκε η δομή της ουράς με αναφορά στις πράξεις της εισαγωγής και της εξαγωγής στοιχείων από αυτήν. Στη συνέχεια παρουσιάσθηκαν προβλήματα η λύση των οποίων εντάσσεται στις κατηγορίες της αναζήτησης και της ταξινόμησης. Η τεχνική της σειριακής /γραμμικής αναζήτησης στοιχείων από πίνακα δόθηκε με χρήση σχετικών αλγορίθμων και προσδιορίσθηκαν οι περιπτώσεις όπου η μέθοδος αυτή είναι αποτελε-σματική. Έγινε σημαντική εμβάθυνση στη μέθοδο της αναδρομής μέσω διαφόρων παραδειγμάτων επίλυσης γνωστών προβλημάτων. Τέλος, επιγραμματικά παρου-σιάζονται οι δομές της λίστας, του δένδρου και του γράφου.**

**146 / 73**

**Λέξεις κλειδιά**

**Δεδομένα, Πληροφορία, Δομή δεδομένων, Στατικές και δυναμικές δομές, Πίνακες, Στοίβα, Ουρά, FIFO και LIFO,   
Γραμμική αναζήτηση, Ταξινόμηση, Αναδρομή, Λίστες, Δένδρα, Γράφοι.**

**• Ερωτήσεις Θέματα για συζήτηση**

1. **Τι είναι δεδομένα και τι είναι πληροφορία; Να δοθεί σύντομος ορισμός των όρων αυτών.**
2. **Ποιές είναι οι απόψεις από τις οποίες η επιστήμη της Πληροφορικής μελετά τα δεδομένα;**
3. **Να δοθεί ο ορισμός της δομής δεδομένων.**
4. **Ποιές είναι οι βασικές πράξεις επί των δομών δεδομένων;**
5. **Ποιά είναι η εξάρτηση μεταξύ της δομής δεδομένων και του αλγορίθμου που επεξεργάζεται τη δομή;**
6. **Να περιγραφούν οι δύο κυριότερες κατηγορίες των δομών δεδομένων.**
7. **Να περιγραφεί η δομή του πίνακα και να δοθεί πα-ράδειγμα χρήσης του.**
8. **Να δοθεί ο ορισμός της στοίβας.**
9. **Ποιές είναι οι βασικές λειτουργίες που γίνονται σε μία στοίβα;**
10. **Να δοθεί ο ορισμός της ουράς.**
11. **Ποιές είναι οι βασικές λειτουργίες που γίνονται σε μία ουρά;**
12. **Να περιγραφεί η λειτουργία της αναδρομής και να σχολιασθεί η χρησιμότητα της.**

**147 / 73-74**

1. **Να δοθεί αναδρομικός αλγόριθμος υπολογισμού της δύναμης πραγματικού αριθμού υψωμένου σε ακέραια δύναμη.**
2. **Να περιγραφεί η λειτουργία της αναζήτησης.**
3. **Να δοθεί ένα παράδειγμα για τη σειριακή αναζήτηση στοιχείου σε έναν πίνακα.**
4. **Να δοθεί ο ορισμός της έννοιας της ταξινόμησης.**
5. **Να περιγραφεί η ταξινόμηση ευθείας ανταλλαγής και να δοθεί ένα παράδειγμα.**

•**Βιβλιογραφία**

1. **Νικόλαος Γλυνός, Δομές Δεδομένων, Πανεπιστήμιο Ιωαννίνων, 1996.**
2. **Χρήστος Κοιλίας, Δομές Δεδομένων και Οργάνωση Αρχείων. Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1993.**
3. **Ιωάννης Μανωλόπουλος, Δομές Δεδομένων - μία Προσέγγιση με Pascal, Εκδόσεις Art of Text, Θεσσαλονίκη, 1998.**
4. **Νικόλαος Μισυρλής, Δομές Δεδομένων, Αθήνα, 1993.**
5. **D. Brunskill and J. Turner: "Understanding Algorithms and Data Structures", McGraw-Hill, 1996.**
6. **D. E. Knuth: "The Art of Computer Programming: Fundamental Algorithms", Vol.1, 3rd edition, Addison Wesley, 1997.**

**148 / 74**

1. **M.A. Weiss: "Data Structures and Algorithm Analysis", 2nd edition, Benjamin/Cummings, 1995.**

**Διαδίκτυο**

[**www.nist.gov/dads/**](http://www.nist.gov/dads/)

**Κόμβος με ευρετήριο όρων για αλγορίθμους, Δομές Δεδομένων και Προβλήματα (Algorithms, Data Structures, and Problems Terms and Definitions for   
the CRC Dictionary of Computer Science, Engineering and Technology).**

**149 / 74**

**Περιεχόμενα 1ου τόμου**

|  |  |  |
| --- | --- | --- |
| **1.** | **Ανάλυση προβλήματος………………………….** | **5** |
|  | **1.1 Η έννοια πρόβλημα………............................** | **8** |
|  | **1.2 Κατανόηση προβλήματος…………………...** | **12** |
|  | **1.3 Δομή προβλήματος……...............................** | **16** |
|  | **1.4 Καθορισμός απαιτήσεων……………….……** | **22** |
|  | **1.5 Κατηγορίες προβλημάτων…………………..** | **32** |
|  | **1.6 Πρόβλημα και υπολογιστής…………………** | **36** |
| **2.** | **Βασικές Έννοιες Αλγορίθμων………………….** | **44** |
|  | **2.1 Τι είναι αλγόριθμος……………………………** | **47** |
|  | **2.2 Σπουδαιότητα αλγορίθμων………………….** | **50** |
|  | **2.3 Περιγραφή και αναπαράσταση αλγορίθμων…………………………………….** | **52** |
|  | **2.4 Βασικές συνιστώσες/εντολές ενός αλγορίθμου…………………………..………..**  **150** | **56** |
|  | **2.4.1 Δομή ακολουθίας……………..……………** | **56** |
|  | **2.4.2 Δομή Επιλογής………………..……………** | **61** |
|  | **2.4.3 Διαδικασίες πολλαπλών επιλογών……………………………………..**  **151** | **67** |
|  | **2.4.4 Εμφωλευμένες Διαδικασίες……………….** | **70** |
|  | **2.4.5 Δομή Επανάληψης……………….…………** | **75** |
| **3.** | **Δομές Δεδομένων και Αλγόριθμοι................** | **97** |
|  | **3.1 Δεδομένα……………….……………………...** | **100** |
|  | **3.2 Αλγόριθμοι+Δομές Δεδομένων=Προγράμματα…………………..** | **103** |
|  | **3.3 Πίνακες……….…………………………………** | **106** |
|  | **3.4 Στοίβα……………………………………………** | **112** |
|  | **3.5 Ουρά………………………………………….....** | **116** |
|  | **3.6 Αναζήτηση……………………………………...** | **124** |
|  | **3.7 Ταξινόμηση……………………………………..** | **128** |
|  | **3.8 Αναδρομή……………………………………….** | **134** |
|  | **3.8.1 Υπολογισμός του παραγοντικού………...** | **134** |
|  | **3.8.2 Υπολογισμός του μέγιστου κοινού διαιρέτη…………………………..……………**  **152** | **136** |
|  | **3.8.3 Υπολογισμός αριθμών ακολουθίας Fibonacci……………………………………..** | **138** |
|  | **3.9 Άλλες δομές δεδομένων……………………..** | **141** |
|  | **3.9.1 Λίστες………………………………………….** | **141** |
|  | **3.9.2 Δένδρα………………………………………...** | **144** |
|  | **3.9.3 Γράφοι…………………………………………** | **145** |

**Βάσει του ν. 3966/2011 τα διδακτικά βιβλία του Δημοτικού, του Γυμνασίου, του Λυκείου, των ΕΠΑ.Λ. και των ΕΠΑ.Σ. τυπώνονται από το ΙΤΥΕ - ΔΙΟΦΑΝΤΟΣ και διανέμονται δωρεάν στα Δημόσια Σχολεία. Τα βιβλία μπορεί να διατίθενται προς πώληση, όταν φέρουν στη δεξιά κάτω γωνία του εμπροσθόφυλλου ένδειξη «ΔIΑΤΙΘΕΤΑΙ ΜΕ ΤΙΜΗ ΠΩΛΗΣΗΣ». Κάθε αντίτυπο που διατίθεται προς πώληση και δεν φέρει την παραπάνω ένδειξη θεωρείται κλεψίτυπο και ο παραβάτης διώκεται σύμφωνα με τις διατάξεις του άρθρου 7 του νόμου 1129 της 15/21 Μαρτίου 1946 (ΦΕΚ 1946, 108, Α').**

**Απαγορεύεται η αναπαραγωγή οποιουδήποτε τμήματος αυτού του βιβλίου, που καλύπτεται από δικαιώματα (copyright), ή η χρήση του σε οποιαδήποτε μορφή, χωρίς τη γραπτή άδεια του Υπουργείου Παιδείας και Θρησκευμάτων / IΤΥΕ - ΔΙΟΦΑΝΤΟΣ.**