



## ΚΕΦΑΛΑΙΟ 4ο

### ΤΕΧΝΙΚΕΣ ΣΧΕΔΙΑΣΗΣ

### ΑΛΓΟΡΙΘΜΩΝ

#### 4.1. Προσδοκώμενα αποτελέσματα



Ολοκληρώνοντας αυτό το κεφάλαιο προσδοκάται πως θα έχεις λάβει εκείνες τις γνώσεις, ώστε να τεκμηριώνεις την αναγκαιότητα ανάλυσης των προβλημάτων και σχεδίασης των κατάλληλων αλγορίθμων. Θα μπορείς να διατυπώνεις σύγχρονες τεχνικές σχεδίασης αλγορίθμων και να περιγράφεις την ακολουθία βημάτων για την ανάλυση των αλγορίθμων. Ακόμα θα μπορείς να περιγράφεις τις κυριότερες προσεγγίσεις επίλυσης και ανάλυσης προβλημάτων. Τέλος, θα έχεις αποκτήσει εκείνες τις δεξιότητες ώστε να μπορείς να επιλύεις προβλήματα με χρήση των κυριοτέρων προσεγγίσεων.

#### 4.2. Επιπλέον παραδείγματα



##### Παράδειγμα 1. Ταξινόμηση με επιλογή

Σε μία τηλεφωνική εταιρεία χρειάζεται να γίνει ταξινόμηση σε αύξουσα σειρά των αριθμών τηλεφώνων με βάση την εξής παρατήρηση : «Από τους αριθμούς τηλεφώνων που δεν έχουν ταξινομηθεί στη σωστή σειρά, να βρεθεί ο μικρότερος αριθμός και να τοποθετηθεί στη σειρά των αριθμών που έχουν ήδη ταξινομηθεί». Να καταγραφεί ο σχετικός αλγόριθμος για την ταξινόμηση 1000 αριθμών τηλεφώνων.

Για να υλοποιήσεις τον αλγόριθμο πρέπει να υπάρξει κάποια απόφαση σχετικά με τις δομές δεδομένων που θα χρειασθείς. Είναι δεδομένο ότι θα έχεις 1000 αριθμούς τηλεφώνων που πρέπει να βάλεις σε σωστή αύξουσα σειρά. Στις περιπτώσεις προβλη-



μάτων όπου είναι δεδομένος ο συνολικός αριθμός από «αντικείμενα» που θα έχει το πρόβλημα χρησιμοποιούμε τη δομή του πίνακα, έτσι ώστε κάθε αντικείμενο να έχει τη δική του θέση. Επομένως γίνεται αρχικά η ανάθεση των αριθμών τηλεφώνων σε κάθε μία από τις θέσεις του πίνακα, με τον τρόπο που έχει περιγραφεί σε προηγούμενα κεφάλαια. Ο παρακάτω αλγόριθμος υποθέτει ότι ο πίνακας THL έχει πάρει τους αριθμούς τηλεφώνων σε τυχαία σειρά και επομένως δεν είναι ταξινομημένος.

```

Αλγόριθμος Αριθμοί_Τηλεφώνων
Δεδομένα // THL//
Για i από 1 μέχρι 1000
    j ← i
    Για k από i+1 μέχρι 1000
        Αν THL[k] < THL[j] τότε
            j ← k
    Τέλος_αν
Τέλος_επανάληψης
Αντιμετάθεση THL[i], THL[j]
Τέλος_επανάληψης
Αποτελέσματα // THL //
Τέλος Αριθμοί_Τηλεφώνων
  
```



Ο παραπάνω αλγόριθμος αποτελεί μία απλή πρόταση για την ταξινόμηση στοιχείων και είναι γνωστός ως αλγόριθμος **ταξινόμησης με επιλογή** (selection sort). Η ονομασία του οφείλεται στη λογική που χρησιμοποιεί για την ταξινόμηση, η οποία βασίζεται στην επιλογή του μικρότερου στοιχείου από αυτά που δεν έχουν ταξινομηθεί σε κάθε βήμα.

## Παράδειγμα 2. Ανεπιτυχής δυαδική αναζήτηση

Σε ένα παραθεριστικό κέντρο υπάρχουν πολλά καταστήματα και εστιατόρια. Ένας επιχειρηματίας θέλει να ανοίξει ένα κατάστημα και θέλει να του δώσει το όνομα «Άνοιξη». Πρέπει πρώτα να ερευνήσει εάν αυτό το όνομα έχει ήδη δοθεί σε κάποιο άλλο κατάστημα. Έστω ότι όλα τα ονόματα καταστημάτων του παραθεριστικού κέντρου έχουν καταγραφεί σε ένα πίνακα 50 θέσεων. Να προτείνετε τον κατάλληλο αλγόριθμο που θα δώσει την απάντηση στον επιχειρηματία για το εάν μπορεί να ανοίξει το κατάστημα με αυτό το όνομα ή όχι.



Για να βρεις τον κατάλληλο αλγόριθμο πρέπει να χρησιμοποιήσεις την ιδέα της δυαδικής αναζήτησης που έχει παρουσιαστεί με παράδειγμα στο βιβλίο σου (Κεφάλαιο 4). Η διαφοροποίηση σε εκείνο τον αλγόριθμο έχει να κάνει με το ότι, εδώ χρειάζεται να υπάρξει κάποια μέριμνα για το εάν το στοιχείο που αναζητούμε, βρέθηκε στον πίνακα ή όχι. Στις περιπτώσεις αυτές συνηθίζεται να υπάρχει κάποια μεταβλητή που αναλαμβάνει αυτόν το ρόλο και ενημερώνεται με την κατάλληλη τιμή. Συχνά οι μεταβλητές αυτές εκφράζουν δύο καταστάσεις (π.χ. εδώ έχουμε βρέθηκε / δεν βρέθηκε) και για αυτό χαρακτηρίζονται ως δυαδικές μεταβλητές «σημαίες» (boolean flags). Η μεταβλητή found στον παρακάτω αλγόριθμο έχει αυτόν τον ρόλο. Εάν η τιμή της found είναι 0, δεν έχει βρεθεί το όνομα που ψάχνεις, αν η τιμή της είναι 1, έχει βρεθεί.

και επομένως το κατάστημα δεν μπορεί να πάρει το όνομα που έχεις δώσει. Επιπλέον, στον αλγόριθμο που ακολουθεί χρησιμοποιείται ο πίνακας KAT, στοιχεία του οποίου είναι τα ονόματα των 50 καταστημάτων που έχουν διαβασθεί όπως έχει περιγραφεί σε προηγούμενα κεφάλαιο και ΟΝΟΜΑ “Ανοιξη” είναι το όνομα που αναζητούμε για το κατάστημα.

```

Αλγόριθμος Δυαδική_αναζήτηση
Δεδομένα // ΟΝΟΜΑ, KAT //
low ← 0
high ← 50
found ← 0
όσο low ≤ high επανάλαβε
    mid ← (low + high)/2
    Αν KAT[mid] < ΟΝΟΜΑ τότε
        low ← mid+1
    αλλιώς_αν KAT[mid] > ΟΝΟΜΑ τότε
        high ← mid-1
    αλλιώς
        found ← 1
    Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα // found //
Τέλος Δυαδική_αναζήτηση

```

### Παράδειγμα 3. Εύρεση δύο μικρότερων αριθμών.

Σε ένα Τμήμα μίας επιχείρησης χρειάζεται να βρεθούν οι δύο χαμηλότεροι μισθοί με δεδομένο ότι το Τμήμα απασχολεί 50 υπαλλήλους και οι μισθοί τους αποθηκεύονται σε κάποιον πίνακα. Να γραφεί ένας αλγόριθμος που θα υπολογίζει τους δύο μικρότερους μισθούς με δεδομένο τον πίνακα των μισθών των υπαλλήλων.

Το πρόβλημα της ανεύρεσης των δύο μικρότερων στοιχείων ενός πίνακα επιδέχεται διάφορες τεχνικές και τρόπους σχεδίασης. Ο αλγόριθμος που παρουσιάζεται στη συνέχεια είναι αρκετά απλός και δεν έχει καλή αποδοτικότητα.



```

Αλγόριθμος Δύο_Μικρότεροι
Δεδομένα // M //
low1 ← M[1]
pos ← 1
Για i από 2 μέχρι 50
    Αν M[i] < low1 τότε
        low1 ← M[i]
        pos ← i
    Τέλος_αν
Τέλος_επανάληψης

```

```

Αν pos <> 1 τότε
    low2 ← M[1]
αλλιώς
    low2 ← M[2]
Τέλος_αν
Για i από 2 μέχρι 50
    Αν (i <> pos και M[i] < low2) τότε
        low2 ← M[i]
    Τέλος_αν
Τέλος_επανάληψης
Αποτελέσματα // low1 , low2 //
Τέλος Δύο_Μικρότεροι

```

### 4.3. Δραστηριότητες - ασκήσεις



#### Στην τάξη

**ΔΤ1.** Να παρακολουθήσετε την πορεία των αλγορίθμων που προτείνεται στο Παράδειγμα 3 για παραδείγματα πινάκων κάποιων θέσεων. Να συζητήσετε την πιθανότητα να προταθεί κάποια βελτίωση και διαφορετικότητα στην τεχνική σχεδίασης του προτεινόμενου αλγορίθμου.

**ΔΤ2.** Εστω ότι ο κατάλογος των μουσείων της πόλης σας υπάρχει αποθηκευμένος σε έναν πίνακα ο οποίος περιέχει το όνομα κάθε μουσείου. Εστω ότι κάποιος επισκέπτης θέλει να μάθει εάν κάποιο συγκεκριμένο μουσείο (π.χ. Λαογραφικό) υπάρχει στην πόλη σας. Να προτείνετε δύο τρόπους για την αναζήτηση ενός συγκεκριμένου μουσείου από αυτόν τον πίνακα και να συζητήσετε στην τάξη τη διαφορά και τον τρόπο λειτουργίας κάθε αλγορίθμου.

**ΔΤ3.** Να συζητηθεί και να αναλυθεί ο αλγόριθμος που υλοποιεί τη δημιουργία ενός μαγικού τετραγώνου όπως αυτό περιγράφεται στη συνέχεια:

Ένα μαγικό τετράγωνο είναι ένας  $n \times n$  πίνακας από ακέραιους από το 1 μέχρι το  $n^2$  που έχει κατασκευασθεί έτσι ώστε το άθροισμα κάθε γραμμής, κάθε στήλης και κάθε διαγωνίου να είναι το ίδιο, όπως φαίνεται και στο παρακάτω σχήμα :

15	8	1	24	17
16	14	7	5	23
22	20	13	6	4
3	21	19	12	10
9	2	25	18	11



Στο παραπάνω σχήμα παρουσιάζεται ένα μαγικό τετράγωνο με  $n=5$ , στο οποίο κάθε γραμμή ή στήλη ή διαγώνιος έχει άθροισμα 65. Έχει προταθεί κάποιος συγκεκριμένος κανόνας για τη δημιουργία ενός μαγικού τετραγώνου για  $n$  περιττό αριθμό. Ο κανόνας αυτός συνοψίζεται στα εξής :

«Ξεκινούμε τοποθετώντας τον αριθμό 1 στη μεσαία θέση της πρώτης γραμμής. Στη συνέχεια προχωρούμε αναθέτοντας τους αριθμούς 2, 3, 4, ... κλπ μετακινούμενοι συνεχώς προς τα επάνω και αριστερά μέχρι να γεμίσει το μαγικό τετράγωνο. Όταν κάποια μετακίνηση προς τα επάνω ή προς τα αριστερά μας οδηγεί εκτός των ορίων του τετραγώνου πηγαίνουμε στο αντι-διαμετρικό άκρο της γραμμής ή της στήλης στην οποία βρεθήκαμε. Επίσης αν η μετακίνηση μας οδηγεί σε κατειλημμένη θέση, τότε επιλέγεται η θέση κάτω από αυτήν όπου έγινε η τελευταία ανάθεση».

Ο παρακάτω αλγόριθμος υλοποιεί αυτόν τον κανόνα χρησιμοποιώντας τη δομή ενός διδιάστατου πίνακα (square) για να κρατηθούν οι τιμές των στοιχείων του μαγικού τετραγώνου

**Αλγόριθμος** Μαγικό\_τετράγωνο

**Δεδομένα** //  $n$  //

**Για**  $i$  **από** 1 **μέχρι**  $n$

**Για**  $j$  **από** 1 **μέχρι**  $n$

$\text{square}[i,j] \leftarrow 0$

**Τέλος\_επανάληψης**

**Τέλος\_επανάληψης**

$i \leftarrow 1$

$j \leftarrow (n+1)/2$

$\text{square}[i,j] \leftarrow 1$

**Για**  $\text{key}$  **από** 2 **μέχρι**  $n*n$

**Αν**  $i > 1$  **τότε**

$k \leftarrow i-1$

**αλλιώς**

$k \leftarrow n-1$

**Τέλος\_αν**

**Αν**  $j > 1$  **τότε**

$l \leftarrow j-1$

**αλλιώς**

$l \leftarrow n-1$

**Τέλος\_αν**

**Αν**  $\text{square}[k,l] > 0$  **τότε**

$i \leftarrow i+1$

**Αν**  $i = n+1$  **τότε**  $i \leftarrow 1$

**αλλιώς**

$i \leftarrow k$

$j \leftarrow l$

**Τέλος\_αν**

$\text{square}[i,j] \leftarrow \text{key}$

**Τέλος\_επανάληψης**

**Αποτελέσματα** // square //

**Τέλος** Μαγικό\_τετράγωνο



**ΔΤ4.** Ενα πρακτορείο ταξιδιών διοργανώνει μία εκδρομή για το γύρο του κόσμου σε 80 ημέρες. Για να κάνει το σχεδιασμό του ταξιδιού χρειάζεται να επιλέξει κάποιες πόλεις και συγκεκριμένη διαδρομή. Να συζητήσετε στην τάξη και να καταγράψετε τα βασικά βήματα ενός αλγορίθμου που θα σχεδιάζει τη διαδρομή που θα πρέπει να ακολουθήσει ο ταξιδιώτης ξεκινώντας από μία πόλη και καταλήγοντας πάλι σε αυτήν αφού περάσει μία φορά από τις πόλεις που έχουν επιλεγεί. Είναι χρήσιμο στο σχεδιασμό της διαδρομής να παίρνει κανείς την απόφαση για τη μικρότερη δυνατή διαδρομή.



### Στο σπίτι

**Στο τετράδιο σας αντιμετωπίστε τα παρακάτω προβλήματα :**

**ΔΣ1.** Εστω ότι έχεις να παίξεις ένα παιχνίδι όπου προσπαθείς να μαντέψεις ένα αριθμό από το 1 μέχρι το 10 που έχει γράψει κάποιος συμμαθητής σου σε ένα χαρτί. Σε κάθε προσπάθεια, ο συμμαθητής σου απαντά δηλώνοντας αν ο αριθμός του είναι μικρότερος, μεγαλύτερος ή ίσος με το δικό σου. Να γράψεις έναν αλγόριθμο που θα σε οδηγήσει με γρήγορο τρόπο στο να βρεις τον αριθμό που έγραψε ο συμμαθητής σου. Πόσα βήματα θα χρειαστείς μέχρι να βρεις τον αριθμό;

**ΔΣ2.** Στο προηγούμενο κεφάλαιο είχες ασχοληθεί με την ταξινόμηση των δίσκων των CD σου σε χρονολογική σειρά. Να επεκτείνεις τον αλγόριθμο έτσι ώστε να υπάρχει δυνατότητα να βρίσκεις εάν ένα CD με συγκεκριμένο τίτλο υπάρχει στη συλλογή σου ή όχι, δίνοντας τον τίτλο του ( δυαδική αναζήτηση).

**ΔΣ3.** Εστω ότι έχεις τον παρακάτω αλγόριθμο :

```

Αλγόριθμος Expo
Δεδομένα // x, n //
m ← n
pow ← 1
z ← x
Οσο m > 0 επανάλαβε
    Οσο ( m MOD 2 ) = 0 επανάλαβε
        m ← m/2
        z ← z * z
    Τέλος_επανάληψης
    m ← m-1
    pow ← pow*z
Τέλος_επανάληψης
Αποτελέσματα // pow //
Τέλος Expo
  
```

Η «πράξη» MOD έχει ως αποτέλεσμα το υπόλοιπο της ακέραιης διαίρεσης δύο αριθμών (π.χ.  $14 \text{ MOD } 3 = 2$ ,  $4 \text{ MOD } 3 = 1$ ), ενώ το  $m/2$  αναφέρεται στον αμέσως μικρότερο ακέραιο από τον αριθμό που προκύπτει ως αποτέλεσμα της διαίρεσης (π.χ.  $13/2 = 6$ ,  $25/3 = 8$ ).



Να παρακολουθήσεις την πορεία του αλγόριθμου για τα εξής ζεύγη τιμών:

$$x=4, n=2 \quad x=2, n=4 \quad x=3, n=3 \quad x=5, n=2$$

Με βάση την παρακολούθηση που έκανες ποιο είναι το συμπέρασμά σου για το αποτέλεσμα του παραπάνω αλγορίθμου ;

**ΔΣ4.** Στη βιβλιοθήκη ενός σχολείου υπάρχουν πολλά βιβλία σχετικά με τη γεωγραφία και τα ταξίδια. Έστω ότι κάθε βιβλίο έχει ένα μοναδικό κωδικό και καταχωρείται σε ηλεκτρονικό υπολογιστή ο τίτλος και ο συγγραφέας κάθε βιβλίου. Να γραφεί αλγόριθμος που θα διαβάζει το όνομα ενός συγγραφέα και θα βρίσκει τον κωδικό (ή τους κωδικούς) και τον τίτλο (ή τους τίτλους) των βιβλίων αυτού του συγγραφέα που υπάρχουν στη βιβλιοθήκη.

**ΔΣ5.** Ένας διαγωνισμός τραγουδιού στην Ευρώπη διεξάγεται ως εξής. Γίνεται μία πρώτη ακρόαση των τραγουδιών κάθε χώρας από την Κριτική Επιτροπή η οποία δίνει κάποιους βαθμούς σε κάθε τραγούδι (από 1- 100). Έστω ότι είναι γνωστοί οι βαθμοί που δόθηκαν στο τραγούδι κάθε χώρας. Να γραφεί ένας αλγόριθμος που θα επιλέγει για τη συνέχεια στη δεύτερη φάση του διαγωνισμού τις χώρες με τη μεγαλύτερη βαθμολογία κάθε φορά ώστε το άθροισμα της βαθμολογίας όλων των τραγουδιών που θα προχωρήσουν στη δεύτερη φάση να είναι μικρότερο από 1000 βαθμούς.

**ΔΣ6.** Να παρακολουθήσεις το πρόβλημα για τον «Γύρο του κόσμου» που δόθηκε στις δραστηριότητες για την τάξη (ΔΤ4) και να κάνεις ένα σχήμα για 10 πόλεις και των μεταξύ τους αποστάσεων με δεδομένο ότι υπάρχει αεροπορική σύνδεση για κάποιες από αυτές. Στη συνέχεια να δώσεις σχηματικά τη λύση για τη μικρότερη δυνατή διαδρομή.

#### 4.4. Τεστ αυτοαξιολόγησης



**Δίνονται οι παρακάτω ομάδες προτάσεων. Σε κάθε μία από αυτές, να κάνετε τις απαραίτητες διορθώσεις ώστε να ισχύουν οι προτάσεις**

1. Οι μέθοδοι λύσης ενός προβλήματος που προκύπτουν από την υλοποίηση του σε συγκεκριμένο υπολογιστικό σύστημα, οδηγούν στη σχεδίαση ενός αλγορίθμου που συνιστά την ακολουθία βημάτων που πρέπει να ακολουθηθούν για να επιλυθεί το πρόβλημα.
2. Κατά την επίλυση ενός προβλήματος, δεν γίνεται σύγκριση των χαρακτηριστικών και των ιδιοτήτων διαφορετικών τεχνικών σχεδίασης ενός αλγορίθμου αλλά επιλέγεται η πλέον εκτενής τεχνική.
3. Η μέθοδος του Δυναμικού Προγραμματισμού για τη σχεδίαση αλγορίθμων χρησιμοποιείται κυρίως για την επίλυση προβλημάτων υποδιαίρεσεων σε μικρότερα μεγέθη προβλημάτων και κυρίως κατά την ταξινόμηση.



**Συμπλήρωσε τα κενά με το σωστή λέξη που λείπει**

- 4 Η \_\_\_\_\_ μέθοδος προχωρά με την παραδοχή ότι σε κάθε βήμα γίνεται επιλογή της τρέχουσας βέλτιστης επιλογής.
- 5 Η τεχνική της \_\_\_\_\_ εντάσσεται στην κατηγορία αντιμετώπισης προβλημάτων που είτε περιλαμβάνουν την αναζήτηση ενός συνόλου λύσεων, είτε αναζητούν τη βέλτιστη λύση υπό κάποιες προϋποθέσεις.
- 6 Η \_\_\_\_\_ διευκολύνει την αποδοτική ανεύρεση στοιχείου από πίνακα, υποδιαιρώντας τον πίνακα σε δύο μέρη σε κάθε βήμα και συνεχίζοντας με τον κατάλληλο από τους δύο υπο-πίνακες.
- 7 Δύο γνωστές τεχνικές για την ταξινόμηση είναι η \_\_\_\_\_ και η \_\_\_\_\_.

**Χαρακτήρισε τα παρακάτω σαν σωστό ή λάθος**

- 8 Γενικότερα, κάθε τεχνική σχεδίασης χρειάζεται να υποστηρίζει τα εξής :
  - ➡ να αντιμετωπίζει με τα δικά της τρόπο τα δεδομένα.
  - ➡ να έχει τη δική της ακολουθία εντολών.
  - ➡ να διαθέτει τη δική της αποδοτικότητα.
9. Η γραμμική αναζήτηση διευκολύνει την αποδοτική ανεύρεση στοιχείου από πίνακα και δεν υπάρχει καλύτερος τρόπος αναζήτησης στοιχείου από πίνακα.
10. Η τεχνική του Δυναμικού Προγραμματισμού είναι πιο αποδοτική από την τεχνική Διαίρει και Βασίλευε.

**Διάλεξε όλα όσα χρειάζεται μεταξύ των προτεινόμενων**

11. Κατά την ανάλυση ενός προβλήματος θα πρέπει να δοθεί απάντηση σε κάθε μία από τις επόμενες ερωτήσεις:
  - A) Ποιά είναι τα δεδομένα και το μέγεθος του προβλήματος
  - B) Ποιές είναι τα περιφερειακά του συστήματος στο οποίο θα επιλυθεί ο αλγόριθμος
  - Γ) Ποιά είναι η χρησιμότητα του αλγορίθμου
  - Δ) Πώς θα καταγραφεί η λύση σε ένα πρόβλημα; (π.χ. σε ψευδογλώσσα)
  - Ε) Ποιός είναι ο τρόπος υλοποίησης στο συγκεκριμένο υπολογιστικό σύστημα (π.χ. επιλογή γλώσσας προγραμματισμού).
12. Κάθε τεχνική σχεδίασης αλγορίθμου χρειάζεται να υποστηρίζει τα εξής :
  - A) Να υλοποιείται σε συγκεκριμένη γλώσσα προγραμματισμού.
  - B) Να αντιμετωπίζει με το δικό της τρόπο τα δεδομένα.



- Γ) Να έχει τη δική της ακολουθία εντολών.
  - Δ) Να δέχεται συγκεκριμένη είσοδο από πληκτρολόγιο.
  - Ε) Να διαθέτει τη δική της αποδοτικότητα.
13. Η περιγραφή της μεθόδου Διαίρει και Βασίλευε για τη σχεδίαση αλγορίθμων μπορεί να αποδοθεί με τα επόμενα βήματα:
- Α) Δίνεται το όνομα του χρήστη του αλγορίθμου.
  - Β) Δίνεται για επίλυση ένα στιγμιότυπο ενός προβλήματος.
  - Γ) Το συνολικό μέγεθος του προβλήματος διαιρείται δια 2.
  - Δ) Υποδιαίρεση του στιγμιότυπου του προβλήματος σε υπο-στιγμιότυπα του ίδιου προβλήματος.
  - Ε) Δίνεται ανεξάρτητη λύση σε κάθε ένα υπο-στιγμιότυπο.
  - Ζ) Συνδυάζονται όλες οι μερικές λύσεις που βρέθηκαν για τα υπο-στιγμιότυπα, έτσι ώστε να δοθεί η συνολική λύση του προβλήματος.
  - Η) Εκτυπώνεται κάθε φορά πολλά είδη λύσεων.