



## Δομές δεδομένων και αλγόριθμοι

### Δεδομένα

Τα δεδομένα είναι ακατέργαστα γεγονότα. Η συλλογή των ακατέργαστων δεδομένων και ο συσχετισμός τους δίνει ως αποτέλεσμα την *πληροφορία*. Η μέτρηση, η κωδικοποίηση, η μετάδοση της πληροφορίας αποτελεί αντικείμενο μελέτης ενός ιδιαίτερου κλάδου, της *Θεωρίας Πληροφοριών*.

Η πληροφορική θεωρείται η επιστήμη που μελετά τα δεδομένα από τις ακόλουθες σκοπιές:

- **Υλικού.** Το υλικό επιτρέπει στα δεδομένα να αποθηκεύονται στην κύρια μνήμη και τις περιφερειακές συσκευές με διάφορες αναπαραστάσεις. Τέτοιες μορφές είναι η *δυναμική*, ο *κώδικας ASCII*, ο *κώδικας EBCDIC*, το *συμπλήρωμα του 1 ή του 2 κ.λ.π.*
- **Γλωσσών προγραμματισμού.** Οι γλώσσες προγραμματισμού υψηλού επιπέδου επιτρέπουν τη χρήση διάφορων τύπων μεταβλητών για να περιγράψουν ένα δεδομένο. Ο μεταφραστής κάθε γλώσσας φροντίζει για την αποδοτικότερη μορφή αποθήκευσης κάθε μεταβλητής, από πλευράς υλικού.
- **Δομών δεδομένων.** (βλέπε παρακάτω)
- **Ανάλυσης δεδομένων.** Μελετώνται τρόποι καταγραφής και αλληλοσυσχέτισης των δεδομένων προκειμένου να αναπαρασταθεί η γνώση για πραγματικά γεγονότα. Εδώ ανήκουν οι τεχνολογίες των *βάσεων δεδομένων*, της *μοντελοποίησης δεδομένων* και της *αναπαράστασης γνώσης*.

### Αλγόριθμοι + Δομές δεδομένων = Προγράμματα

Τα δεδομένα ενός προβλήματος αποθηκεύονται στον υπολογιστή, είτε στην κύρια μνήμη του είτε στη δευτερεύουσα. Η αποθήκευση δεν γίνεται κατά τυχαίο τρόπο αλλά συστηματικά χρησιμοποιώντας μια *δομή*.

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

Οι βασικές λειτουργίες (ή πράξεις) επί των δομών δεδομένων είναι οι ακόλουθες:

- **Προσπέλαση**, πρόσβαση σε έναν κόμβο με σκοπό να εξεταστεί ή να τροποποιηθεί το περιεχόμενό του.



## Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον Ανασκόπηση Θεωρίας - 3<sup>ο</sup> Κεφάλαιο



- **Εισαγωγή**, δηλαδή η προσθήκη νέων κόμβων σε μια υπάρχουσα δομή.
- **Διαγραφή**, που αποτελεί το αντίστροφο της εισαγωγής, δηλαδή ένας κόμβος αφαιρείται από μια δομή.
- **Αναζήτηση**, κατά την οποία προσπελαύνονται οι κόμβοι μιας δομής, προκειμένου να εντοπιστούν ένας ή περισσότεροι κόμβοι που έχουν μια δεδομένη ιδιότητα.
- **Ταξινόμηση**, όπου οι κόμβοι μιας δομής διατάσσονται κατά αύξουσα ή φθίνουσα σειρά.
- **Αντιγραφή**, κατά την οποία όλοι οι κόμβοι ή μερικοί από τους κόμβους μιας δομής αντιγράφονται σε μια άλλη δομή.
- **Συγχώνευση**, κατά την οποία δυο ή περισσότερες δομές συνενώνονται σε μια ενιαία δομή.
- **Διαχωρισμός**, που αποτελεί την αντίστροφη πράξη της συγχώνευσης.

Στην πράξη σπάνια χρησιμοποιούνται και οι οκτώ λειτουργίες για κάποια δομή. Μια δομή δεδομένων μπορεί να είναι αποδοτικότερη από μια άλλη δομή με κριτήριο κάποια λειτουργία αλλά λιγότερο αποδοτική για κάποια άλλη λειτουργία. Γι' αυτό υπάρχουν διαφορετικές δομές από τις οποίες επιλέγεται κάθε φορά η καταλληλότερη.

Υπάρχει μεγάλη εξάρτηση μεταξύ της δομής δεδομένων και του αλγόριθμου που επεξεργάζεται τη δομή. Αυτό διατυπώνεται με την εξίσωση

$$\text{Αλγόριθμοι} + \text{δομές δεδομένων} = \text{Προγράμματα}$$

Οι δομές δεδομένων διακρίνονται σε δυο μεγάλες κατηγορίες: τις **στατικές** και τις **δυναμικές**.

Οι **δυναμικές δομές** δεν αποθηκεύονται σε συνεχόμενες θέσεις μνήμης αλλά στηρίζονται στην τεχνική της **δυναμικής παραχώρησης μνήμης**. Οι δομές αυτές δεν έχουν σταθερό μέγεθος, αλλά ο αριθμός των κόμβων τους μεγαλώνει και μικραίνει καθώς στη δομή εισάγονται νέα δεδομένα ή διαγράφονται κάποια δεδομένα. Όλες οι σύγχρονες γλώσσες προγραμματισμού προσφέρουν τη δυνατότητα δυναμικής παραχώρησης μνήμης. Στην ύλη μας ωστόσο μελετάμε μόνο τις στατικές δομές.

Με τον όρο στατική δομή δεδομένων εννοείται ότι το ακριβές μέγεθος μνήμης καθορίζεται κατά τη στιγμή του προγραμματισμού και κατά συνέπεια κατά τη στιγμή της μετάφρασής τους και όχι κατά τη στιγμή της εκτέλεσης του προγράμματος (βλέπε 6<sup>ο</sup> κεφάλαιο). Μια άλλη σημαντική διαφορά σε σχέση με τις δυναμικές δομές δεδομένων είναι ότι τα στοιχεία των στατικών δομών δεδομένων αποθηκεύονται σε συνεχόμενες θέσεις μνήμης.

Στην πράξη οι στατικές δομές δεδομένων υλοποιούνται με πίνακες. Ένας **πίνακας** ορίζεται ως μια δομή που περιέχει στοιχεία του ίδιου τύπου. Η δήλωση των στοιχείων ενός πίνακα και η μέθοδος αναφοράς τους εξαρτάται από τη συγκεκριμένη γλώσσα υψηλού



## Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον Ανασκόπηση Θεωρίας - 3<sup>ο</sup> Κεφάλαιο



επιπέδου που χρησιμοποιείται. Γενικά, η αναφορά στα στοιχεία ενός πίνακα γίνεται με τη χρήση του συμβολικού ονόματος του πίνακα ακολουθούμενου από την τιμή ενός ή περισσότερων δεικτών σε παρένθεση ή αγκύλες.

Ένας πίνακας μπορεί να είναι μονοδιάστατος, διδιάστατος και γενικά  $n$ -διάστατος.

### Στοιίβα

Μια στοιίβα δεδομένων μοιάζει με μια στοιίβα από πιάτα. Τα δεδομένα που βρίσκονται στην κορυφή της στοιίβας λαμβάνονται πρώτα, ενώ αυτά που βρίσκονται στο βάθος της στοιίβας λαμβάνονται τελευταία. Αυτή η μέθοδος επεξεργασίας ονομάζεται *Τελευταίο μέσα, πρώτο έξω (LIFO)*.

Δυο είναι οι κύριες λειτουργίες μιας στοιίβας:

- Η **ώθηση** στοιχείου στην κορυφή της στοιίβας
- Η **απώθηση** ενός στοιχείου από τη στοιίβα

Η διαδικασία της ώθησης πρέπει οπωσδήποτε να ελέγχει αν η στοιίβα είναι γεμάτη, οπότε λέγεται ότι συμβαίνει *υπερχείλιση* της στοιίβας. Αντίστοιχα η διαδικασία της απώθησης ελέγχει αν υπάρχει ένα τουλάχιστον στοιχείο στη στοιίβα, δηλαδή ελέγχει αν γίνεται *υποχείλιση* της στοιίβας.

Η υλοποίηση μιας στοιίβας γίνεται με τη βοήθεια ενός μονοδιάστατου πίνακα. Μια βοηθητική μεταβλητή (με όνομα συνήθως *top*) χρησιμοποιείται για να δείχνει το στοιχείο το οποίο τοποθετήθηκε τελευταίο στην κορυφή της στοιίβας. Κατά την ώθηση αρκεί να αυξηθεί η τιμή της μεταβλητής *top* κατά ένα και στη θέση αυτή να εισέλθει το νέο στοιχείο. Κατά την απώθηση εξέρχεται πρώτα το στοιχείο που βρίσκεται στη θέση που δείχνει η μεταβλητή *top* και στη συνέχεια η μεταβλητή αυτή μειώνεται κατά ένα για να δείχνει τη νέα κορυφή.

### Ουρά

Σε μια ουρά αναμονής εκείνος που στάθηκε πρώτος στην ουρά εξυπηρετείται πρώτος. Αυτή η μέθοδος επεξεργασίας ονομάζεται *Πρώτο μέσα, πρώτο έξω (FIFO)*.

Δυο είναι οι κύριες λειτουργίες που εκτελούνται σε μια ουρά:

- Η **εισαγωγή** στοιχείου στο πίσω άκρο της ουράς
- Η **εξαγωγή** στοιχείου από το εμπρός άκρο της ουράς

Αντίθετα από τη στοιίβα, στην περίπτωση της ουράς απαιτούνται δυο 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 αντίστοιχα.



## Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον Ανασκόπηση Θεωρίας - 3<sup>ο</sup> Κεφάλαιο



Η υλοποίηση μιας ουρά γίνεται με τη βοήθεια ενός μονοδιάστατου πίνακα. Για την εισαγωγή ενός νέου στοιχείου στην ουρά αυξάνεται ο δείκτης rear κατά ένα και στη θέση αυτή τοποθετείται το στοιχείο. Κατά την εξαγωγή, 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 πρέπει να ελέγχεται πριν από οποιαδήποτε ενέργεια αν υπάρχει ελεύθερος χώρος στον πίνακα για την εισαγωγή και αν υπάρχει τουλάχιστον ένα στοιχείο για την εξαγωγή.

### Αναζήτηση

Υπάρχουν αρκετές μέθοδοι αναζήτησης σε πίνακα που εξαρτώνται αν ο πίνακας είναι ταξινομημένος ή όχι. Μια άλλη παράμετρος είναι αν ο πίνακας περιέχει στοιχεία που είναι όλα διάφορα μεταξύ τους ή όχι.

Η πιο απλή μορφή αναζήτησης στοιχείου σε πίνακα είναι η *σειριακή* ή *γραμμική* μέθοδος. Η σειριακή μέθοδος 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  $n \leq 20$ )
- Η αναζήτηση σε έναν συγκεκριμένο πίνακα γίνεται σπάνια

Μια αποτελεσματικότερη μέθοδος αναζήτησης είναι η *δυσδική* αναζήτηση.

### Ταξινόμηση

Η τακτοποίηση των κόμβων μιας δομής με μια ιδιαίτερη σειρά ονομάζεται *ταξινόμηση* ή *διάταξη*. Ο σκοπός της ταξινόμησης είναι να διευκολυνθεί στη συνέχεια η αναζήτηση των στοιχείων του πίνακα.

#### • Ταξινόμηση ευθείας ανταλλαγής

Η μέθοδος της *ταξινόμησης ευθείας ανταλλαγής* βασίζεται στην αρχή της σύγκρισης και ανταλλαγής ζευγών γειτονικών στοιχείων, μέχρις ότου διαταχθούν όλα τα στοιχεία. Σύμφωνα με τη μέθοδο αυτή κάθε φορά γίνονται διαδοχικές προσπελάσεις 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 ως φυσαλίδες σε μια δεξαμενή νερού με βάρη σύμφωνα με την τιμή τους, κάθε προσπέλαση στον πίνακα έχει ως αποτέλεσμα την άνοδο της φυσαλίδας στο κατάλληλο επίπεδο βάρους. Η μέθοδος είναι γνωστή ως *ταξινόμηση φυσαλίδας*. Η ταξινόμηση της φυσαλίδας πραγματοποιείται για έναν πίνακα  $A[n]$  με το επόμενο τμήμα αλγόριθμου.



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον  
Ανασκόπηση Θεωρίας - 3<sup>ο</sup> Κεφάλαιο



Για  $i$  από 2 μέχρι  $n$

Για  $j$  από  $n$  μέχρι  $i$  με\_βήμα  $-1$

Αν  $A[j - 1] > A[j]$  τότε

$temp \leftarrow A[j - 1]$

$A[j - 1] \leftarrow A[j]$

$A[j] \leftarrow temp$

τέλος\_αν

τέλος\_επανάληψης

τέλος\_επανάληψης



Ο παραπάνω αλγόριθμος ταξινομεί τον πίνακα  $A[n]$  κατά αύξουσα σειρά, δηλαδή στην πρώτη θέση του πίνακα  $A$  τοποθετείται το μικρότερο στοιχείο και στην τελευταία θέση το μεγαλύτερο στοιχείο. Αν επιθυμούμε το αντίθετο, δηλαδή στην πρώτη θέση να τοποθετηθεί το μεγαλύτερο στοιχείο και στην τελευταία θέση το μικρότερο στοιχείο, θα πρέπει να αντικαταστήσουμε την εντολή

Αν  $A[j - 1] > A[j]$  τότε

με την εντολή

Αν  $A[j - 1] < A[j]$  τότε

Η μόνη διαφορά μεταξύ των δυο εντολών είναι το ανισοτικό σύμβολο.



Υπάρχουν περιπτώσεις όπου πρέπει να ταξινομήσουμε ταυτόχρονα δυο πίνακες. Θεωρήστε, για παράδειγμα, ότι έχουμε δυο μονοδιάστατους πίνακες  $B[n]$  και  $A[n]$  με τα στοιχεία του  $B$  να περιέχουν τα ονόματα  $n$  υπαλλήλων και τα στοιχεία του  $A$  τις μηνιαίες αποδοχές τους. Το στοιχείο  $B[1]$  περιέχει το όνομα του πρώτου υπαλλήλου και το στοιχείο  $A[1]$  τις αντίστοιχες αποδοχές του, το στοιχείο  $B[2]$  περιέχει το όνομα του δεύτερου υπαλλήλου και το στοιχείο  $A[2]$  τις δικές του αποδοχές κ.ο.κ. Αν μας ζητηθεί να εμφανίσουμε τα ονόματα των υπαλλήλων κατά αύξουσα ή φθίνουσα σειρά μηνιαίων αποδοχών, πρέπει να ταξινομήσουμε κατάλληλα τον πίνακα  $A[n]$  των μηνιαίων αποδοχών και ταυτόχρονα τον πίνακα  $B[n]$  των ονομάτων, ώστε να εξακολουθεί να υπάρχει αντιστοιχία μεταξύ των στοιχείων των δυο πινάκων ( το στοιχείο  $B[1]$  να περιέχει το όνομα ενός υπαλλήλου και το στοιχείο  $A[1]$  τις αντίστοιχες αποδοχές του, το στοιχείο  $B[2]$  να περιέχει το όνομα ενός δεύτερου υπαλλήλου και το στοιχείο  $A[2]$  τις δικές του αποδοχές κ.ο.κ.). Η

Επιμέλεια Θεωρίας Πιερράτος



Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον  
Ανασκόπηση Θεωρίας - 3<sup>ο</sup> Κεφάλαιο



ταυτόχρονη ταξινόμηση των δυο πινάκων επιτυγχάνεται αν τροποποιήσουμε τον παραπάνω αλγόριθμο ως εξής:

```
Για i από 2 μέχρι n
  Για j από n μέχρι i με_βήμα -1
    Αν A[j - 1] > A[j] τότε
      temp ← A[j - 1]
      A[j - 1] ← A[j]
      A[j] ← temp
      temp2 ← B[j - 1]
      B[j - 1] ← B[j]
      B[j] ← temp2
    τέλος_αν
  τέλος_επανάληψης
τέλος_επανάληψης
```

Ας σημειωθεί ότι η ταξινόμηση φυσαλίδας είναι ο πιο απλός αλλά και ο πιο αργός αλγόριθμος ταξινόμησης. Άλλοι σχετικά απλοί αλγόριθμοι ταξινόμησης είναι η ταξινόμηση με επιλογή και η ταξινόμηση με παρεμβολή. Ο πιο γρήγορος αλγόριθμος ταξινόμησης είναι η «γρήγορη ταξινόμηση».