

ΘΕΩΡΙΑ - Κεφάλαιο 3 και 9

Να δοθεί ο **ορισμός της δομής δεδομένων**.

Δομή Δεδομένων είναι ένα σύνολο αποθηκευμένων δεδομένων που υφίστανται επεξεργασία από ένα σύνολο λειτουργιών. (σ.56)

Ποιες είναι οι **βασικές πράξεις επί των δομών δεδομένων**;

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

- | | |
|-----------------------|--|
| Εγγραφή | • Προσπέλαση (access), πρόσβαση σε ένα κόμβο με σκοπό να εξετασθεί ή να τροποποιηθεί το περιεχόμενό 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. |
| Αρχείο | • Αναζήτηση (searching), κατά την οποία προσπελαύνονται οι κόμβοι μιας δομής, προκειμένου να εντοπιστούν ένας ή περισσότεροι που έχουν μια δεδομένη ιδιότητα. |
| | • Ταξινόμηση (sorting), όπου οι κόμβοι μιας δομής διατάσσονται κατά αύξουσα ή φθίνουσα σειρά. |
| Βάση δεδομένων | • Αντιγραφή (copying), κατά την οποία όλοι οι κόμβοι ή μερικοί από τους κόμβους μιας 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. (σ.56) |

Ποια είναι η εξάρτηση μεταξύ της **δομής δεδομένων** και του **αλγορίθμου** που επεξεργάζεται τη δομή;

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 κατάλληλης δομής κάθε φορά. Υπάρχει μεγάλη εξάρτηση μεταξύ της δομής δεδομένων και του αλγόριθμου που επεξεργάζεται τη δομή. Μάλιστα, το πρόγραμμα πρέπει να θεωρεί τη δομή δεδομένων και τον αλγόριθμο ως μία αδιάσπαστη ενότητα.

(Αλγόριθμοι + Δομές Δεδομένων = Προγράμματα - Wirth) (σ.57)

Να περιγραφούν οι **δύο κυριότερες κατηγορίες των δομών δεδομένων**.

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

Στατικές : - το ακριβές μέγεθος της απαιτούμενης κύριας μνήμης καθορίζεται κατά τη στιγμή του προγραμματισμού τους, και κατά συνέπεια κατά τη στιγμή της μετάφρασης

- τα στοιχεία αποθηκεύονται σε συνεχόμενες θέσεις μνήμης

Δυναμικές : - δεν έχουν σταθερό μέγεθος, αλλά ο αριθμός των κόμβων τους μεγαλώνει και μικραίνει καθώς στη δομή εισάγονται ή διαγράφονται δεδομένα αντίστοιχα κατά τη διάρκεια της εκτέλεσής τους προγράμματος

- Δεν αποθηκεύονται σε συνεχόμενες θέσεις μνήμης αλλά στηρίζονται στην τεχνική της λεγόμενης **δυναμικής παραχώρησης μνήμης**. (σ.57)

Να περιγραφεί η **δομή του πίνακα** και να δοθεί παράδειγμα χρήσης του.

Πίνακας είναι μια στατική δομή που περιέχει στοιχεία του ίδιου τύπου (δηλαδή ακέραιους, πραγματικούς κ.λπ).

Η χρήση των στοιχείων ενός πίνακα γίνεται με τη χρήση του συμβολικού ονόματος του πίνακα ακολουθούμενου από την τιμή ενός ή περισσότερων δείκτων (indexes) σε παρένθεση ή αγκύλη. (π.χ. πίνακας[γραμμή, στήλη]). Το πλήθος των στοιχείων του πίνακα ορίζεται κατά τη διάρκεια του προγραμματισμού και οι θέσεις μνήμης είναι συνεχόμενες. (σ.58)

Σε ένα δισδιάστατο **πίνακα τι δηλώνει κάθε δείκτης**;

Σε ένα δισδιάστατο πίνακα, ο πρώτος δείκτης δείχνει τη **γραμμή** και ο δεύτερος τη **στήλη**.

Να περιγραφεί η **λειτουργία της αναζήτησης**.

Στην **αναζήτηση** (searching) ενός στοιχείου σε πίνακα θέλουμε να βρούμε τη θέση του στοιχείου στον πίνακα δοθείσης μιας τιμής (key). (σ.63)

Η μέθοδος της αναζήτησης που θα χρησιμοποιηθεί εξαρτάται 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.

Να δοθεί ένα παράδειγμα για τη σειριακή αναζήτηση στοιχείου σε έναν πίνακα. (σ64)

Πότε χρησιμοποιείται η σειριακή αναζήτηση;

Η σειριακή μέθοδος αναζήτησης είναι η πιο απλή, αλλά και η λιγότερη 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 ένα συγκεκριμένο πίνακα γίνεται σπάνια (σ.64)

Να δοθεί ο **ορισμός** της έννοιας της **ταξινόμησης**.

Δοθέντων των στοιχείων $a_1, a_2, \dots, a_n$ η ταξινόμηση συνίσταται στη **μετάθεση** (permutation) της θέσης των στοιχείων, ώστε να τοποθετηθούν σε μία σειρά $a_{k1}, a_{k2}, \dots, a_{kn}$ έτσι ώστε, δοθείσης μίας **συνάρτησης**

διάταξης (ordering function), f , να ισχύει: $f(a_{k1}) \leq f(a_{k2}) \leq \dots \leq f(a_{kn})$ (σ.65)

Να περιγραφεί η **ταξινόμηση ευθείας ανταλλαγής (φυσалиδας)** και να δοθεί ένα παράδειγμα. (σ66,67)

Περιγράψτε τις **δομές δεδομένων σε δευτερεύουσα μνήμη**.

Τα στοιχεία ενός **αρχείου** ονομάζονται **εγγραφές** (records), όπου κάθε εγγραφή αποτελείται από ένα ή περισσότερα **πεδία** (fields). Ένα σύνολο από αρχεία αποτελούν μια βάση δεδομένων. (σ.66)

Τι γνωρίζεται για τα **κλειδιά στα αρχεία**;

Τα πεδία χωρίζονται σε αυτά που ταυτοποιούν την εγγραφή, και σε άλλα που περιγράφουν διάφορα χαρακτηριστικά της εγγραφής.

Το πεδίο που ταυτοποιεί την εγγραφή ονομάζεται **πρωτεύον κλειδί** (primary key) ή απλά κλειδί.

Αν υπάρχει πρωτεύον κλειδί και υπάρχει και άλλο πεδίο που ταυτοποιεί την εγγραφή, αυτό αποκαλείται

δευτερεύον κλειδί (secondary keys). (σ.66)

Ποιες οι **διαφορές** στις **δομές δεδομένων** στην **κύρια και δευτερεύουσα μνήμη**;

Στη δευτερεύουσα μνήμη (δίσκους κλπ) αποθηκεύονται τα δεδομένα έχουμε μεγάλο όγκο δεδομένων και το μέγεθος της κύριας μνήμης δεν επαρκεί για την αποθήκευση τους.

Επίσης τα δεδομένα δεν χάνονται αν διακοπεί η ηλεκτρική παροχή και 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. (σ.66)

Ποια είναι τα **μειονεκτήματα** από τη **χρήση πινάκων**;

Οι πίνακες απαιτούν μνήμη. Κάθε πίνακας δεσμεύει από την αρχή του προγράμματος πολλές θέσεις μνήμης. Σε ένα μεγάλο και σύνθετο πρόγραμμα η άσκοπη χρήση μεγάλων πινάκων μπορεί να οδηγήσει ακόμη και σε αδυναμία εκτέλεσης του προγράμματος.

Οι πίνακες περιορίζουν τις δυνατότητες του προγράμματος. Οι πίνακες είναι στατικές δομές και το μέγεθός τους πρέπει να δηλώνεται στην αρχή του προγράμματος, ενώ παραμένει υποχρεωτικά σταθερό κατά την εκτέλεση του προγράμματος.

Ποιες είναι οι **τυπικές επεξεργασίες σε πίνακες**;

1. Υπολογισμός αθροισμάτων στοιχείων του πίνακα.
2. Εύρεση του μέγιστου ή του ελάχιστου στοιχείου.
3. Ταξινόμηση των στοιχείων του πίνακα.
4. Αναζήτηση ενός στοιχείου του πίνακα.
5. Συγχώνευση δύο πινάκων.

Ενότητα 2. Μέθοδος Διαίρει και Βασίλευε

Τι είναι η μέθοδος σχεδίασης αλγορίθμων «Διαίρει και Βασίλευε» ;

Η «**Διαίρει και Βασίλευε**» (divide and conquer) αποτελεί μια **μέθοδο σχεδίασης** αλγορίθμων στην οποία εντάσσονται οι τεχνικές που υποδιαιρούν ένα πρόβλημα σε μικρότερα υποπροβλήματα, που έχουν την ίδια τυποποίηση με το αρχικό πρόβλημα, αλλά είναι μικρότερα σε μέγεθος. Με όμοιο τρόπο, τα υποπροβλήματα αυτά μπορούν να διαιρεθούν σε ακόμη μικρότερα υποπροβλήματα κ.ο.κ. Έτσι η επίλυση ενός προβλήματος έγκειται στη σταδιακή επίλυση των όσο το δυνατόν μικρότερων υποπροβλημάτων, ώστε τελικά να προκύψει η συνολική λύση του αρχικού ευρύτερου προβλήματος. Η προσέγγιση αυτή ονομάζεται «**από πάνω προς τα κάτω**» (top-down).

Βασιζόμενοι στη μέθοδο «Διαίρει και Βασίλευε», ποιος είναι ο **μέγιστος αριθμός των συγκρίσεων** (επαναλήψεων) που απαιτούνται για την εύρεση ενός στοιχείου σε ένα σύνολο «n» ταξινομημένων στοιχείων;

Ο μέγιστος αριθμός των συγκρίσεων (επαναλήψεων) που απαιτούνται για την εύρεση ενός στοιχείου σε ένα σύνολο «n» ταξινομημένων στοιχείων, συμπεριλαμβανομένης και της περίπτωσης μη ύπαρξης του στοιχείου, δίνεται από το ακέραιο μέρος του $\lceil \log_2(n)+1 \rceil$ (με στρογγυλοποίηση προς τα κάτω),

Επομένως, για την εύρεση του μέγιστου πλήθους των επαναλήψεων θεωρείται γνωστό το $\log_2(n)$.

Για παράδειγμα, σε ένα σύνολο 100 ταξινομημένων στοιχείων ($n=100$), ο μέγιστος αριθμός συγκρίσεων (επαναλήψεων) είναι: $\lceil \log_2(100)+1 \rceil = \lceil 6,643856+1 \rceil = \lceil 7,643856 \rceil = 7$.