Περιεχόμενα

ΜΕΡΟΣ ΔΥΟ Δομές δεδομένων 89

ΜΕΡΟΣ ΤΡΙΑ Ταξινόμηση 287

Πρόλογος

ΑΥΤΟ ΤΟ ΒΙΒΛΙΟ, στην πρωτότυπη αμερικανική του έκδοση, είναι το πρώτο από μια σειρά τριών τόμων που έχουν σκοπό τη μελέτη των σημαντικότερων αλγορίθμων που χρησιμοποιούνται σήμερα στους υπολογιστές. Ο πρώτος τόμος (Μέρη 1-4) καλύπτει τις θεμελιώδεις έννοιες (Μέρος 1), τις δομές δεδομένων (Μέρος 2), τους αλγορίθμους ταξινόμησης (Μέρος 3), και τους αλγορίθμους αναζήτησης (Μέρος 4). Ο δεύτερος τόμος (Μέρος 5) καλύπτει τους γράφους και τους αλγορίθμους γράφων, ενώ ο τρίτος τόμος (Μέρη 6-8) — ο οποίος δεν είχε κυκλοφορήσει ακόμη όταν μεταφραζόταν αυτό το βιβλίο — καλύπτει τα αλφαριθμητικά (Μέρος 6), την υπολογιστική γεωμετρία (Μέρος 7), και προηγμένους αλγορίθμους και εφαρμογές (Μέρος 8).

Τα συγκεκριμένα βιβλία είναι χρήσιμα ως συγγράμματα σχετικά νωρίς στο πρόγραμμα μαθημάτων της επιστήμης των υπολογιστών, αφού αποκτήσουν οι σπουδαστές βασικές προγραμματιστικές δεξιότητες και οικειότητα με τα συστήματα υπολογιστών, αλλά πριν αρχίσουν να παρακολουθούν εξειδικευμένα μαθήματα σε προηγμένους τομείς της επιστήμης των υπολογιστών ή των εφαρμογών υπολογιστών. Τα βιβλία είναι επίσης χρήσιμα και ως μέσα αυτοδιδασκαλίας ή ως οδηγοί αναφοράς για τα άτομα που ασχολούνται με την ανάπτυξη συστημάτων υπολογιστών ή προγραμμάτων εφαρμογών, επειδή περιέχουν υλοποιήσεις χρήσιμων αλγορίθμων και λεπτομερείς πληροφορίες για τα χαρακτηριστικά επιδόσεων αυτών των αλγορίθμων. Η ευρεία προοπτική που ακολουθείται κάνει αυτή τη σειρά βιβλίων κατάλληλη ως εισαγωγή στο συγκεκριμένο γνωστικό πεδίο.

Στο σύνολό τους, οι τρεις τόμοι αποτελούν την *Τρίτη Αμερικανική Έκδοση* ενός βιβλίου το οποίο χρησιμοποιείται ευρέως από σπουδαστές και προγραμματιστές σε ολόκληρο τον κόσμο για πολλά χρόνια. Έχω ξαναγράψει από την αρχή το κείμενο αυτής της έκδοσης και έχω προσθέσει χιλιάδες νέες ασκήσεις, εκατοντάδες νέες εικόνες, δεκάδες νέα προγράμματα, και λεπτομερή σχόλια σε όλες τις εικόνες και τα προγράμματα. Στόχος μου με την προσθήκη αυτού του νέου υλικού ήταν να καλύψω νέα θέματα αλλά και να εξηγήσω αναλυτικότερα πολλούς από τους κλασικούς αλγορίθμους. Η έμφαση που δίνεται (σε όλα τα βιβλία) στους αφηρημένους τύπους δεδομένων, κάνει τα προγράμματα χρήσιμα σε ένα μεγαλύτερο εύρος εφαρμογών και κατάλληλα για σύγχρονα αντικειμενοστρεφή περιβάλλοντα προγραμματισμού. Οι αναγνώστες που έχουν διαβάσει τις προηγούμενες εκδόσεις θα βρουν σε αυτή τη νέα έκδοση αρκετές νέες πληροφορίες, και όλοι οι αναγνώστες θα βρουν πλούσιο διδακτικό υλικό που παρέχει αποτελεσματική πρόσβαση στις βασικές έννοιες.

Αυτά τα βιβλία δεν είναι κατάλληλα μόνο για προγραμματιστές και σπουδαστές της επιστήμης των υπολογιστών. Όλοι όσοι χρησιμοποιούν υπολογιστές θέλουν τα προγράμματα που χρησιμοποιούν να εκτελούνται ταχύτερα ή να λύνουν μεγαλύτερα προβλήματα. Οι αλγόριθμοι που παρουσιάζουμε αντιπροσωπεύουν γνώσεις οι οποίες αναπτύχθηκαν τα τελευταία

50 χρόνια και αποτελούν τη βάση για αποδοτική χρήση του υπολογιστή σε μια ευρεία κλίμακα εφαρμογών. Από τα προβλήματα προσομοίωσης του προβλήματος των *N* σωμάτων στη φυσική μέχρι τα προβλήματα των γενετικών ακολουθιών στη μοριακή βιολογία, οι βασικές μέθοδοι που περιγράφονται εδώ έχουν γίνει απαραίτητες στην επιστημονική έρευνα· επίσης, από τα συστήματα βάσεων δεδομένων μέχρι της μηχανές αναζήτησης στο Διαδίκτυο, έχουν γίνει απαραίτητα τμήματα των σύγχρονων συστημάτων λογισμικού. Καθώς το πεδίο των εφαρμογών υπολογιστών γίνεται όλο και μεγαλύτερο, ταυτόχρονα αυξάνεται και η επίδραση των βασικών αλγορίθμων. Στόχος του παρόντος βιβλίου είναι να λειτουργήσει ως πηγή αναφοράς, έτσι ώστε οι σπουδαστές και οι επαγγελματίες να γνωρίσουν αυτούς τους θεμελιώδεις αλγορίθμους προκειμένου να μπορούν να τους χρησιμοποιούν με έξυπνους τρόπους όποτε παρουσιάζεται ανάγκη, σε οποιαδήποτε εφαρμογή υπολογιστή.

Στόχοι του βιβλίου

Το βιβλίο *Αλγόριθμοι σε Java*, *Τρίτη αμερικανική έκδοση*, *Μέρη 1-4*, αποτελείται από 16 κεφάλαια τα οποία έχουν ομαδοποιηθεί σε τέσσερα κύρια μέρη: θεμελιώδεις έννοιες, δομές δεδομένων, ταξινόμηση, και αναζήτηση. Το βιβλίο έχει στόχο να δώσει στους αναγνώστες τη δυνατότητα να κατανοήσουν τις βασικές ιδιότητες όσο το δυνατό περισσότερων θεμελιωδών αλγορίθμων. Οι αλγόριθμοι που περιγράφονται χρησιμοποιούνται ευρέως για πολλά χρόνια, και αντιπροσωπεύουν ένα βασικό τμήμα γνώσης τόσο για τον ασκούμενο προγραμματιστή όσο και για το σπουδαστή της επιστήμης των υπολογιστών. Ο δεύτερος τόμος είναι αφιερωμένος σε αλγορίθμους γράφων, και ο τρίτος αποτελείται από τέσσερα μέρη που καλύπτουν τα αλφαριθμητικά, τη γεωμετρία, και προχωρημένα θέματα. Ο κύριος στόχος μου στη συγγραφή αυτών των βιβλίων ήταν να συγκεντρώσω τις θεμελιώδεις μεθόδους αυτών των τομέων της επιστήμης των υπολογιστών, έτσι ώστε να κάνω προσιτές τις καλύτερες γνωστές μεθόδους για την επίλυση προβλημάτων με υπολογιστή.

Θα εκτιμήσετε ακόμη περισσότερο το υλικό του βιβλίου αν έχετε ήδη παρακολουθήσει ένα ή δύο προηγούμενα μαθήματα της επιστήμης των υπολογιστών ή διαθέτετε αντίστοιχη εμπειρία στον προγραμματισμό: ένα μάθημα στον προγραμματισμό σε γλώσσα υψηλού επιπέδου όπως η Java, η C, ή η C++, και ίσως ένα ακόμη μάθημα στο οποίο διδάσκονται οι θεμελιώδεις έννοιες των συστημάτων προγραμματισμού. Κατά συνέπεια, το βιβλίο απευθύνεται σε όλους εκείνους που έχουν επαφή με κάποια σύγχρονη γλώσσα προγραμματισμού και σε εκείνους που ασχολούνται με τα βασικά χαρακτηριστικά των σύγχρονων συστημάτων υπολογιστών. Επίσης, στο κείμενο θα βρείτε παραπομπές σε βιβλιογραφία που ίσως σας βοηθήσει να καλύψετε κάποια κενά στα θέματα που περιγράφονται.

Επειδή το μεγαλύτερο μέρος της ύλης των μαθηματικών με το οποίο υποστηρίζονται τα αναλυτικά αποτελέσματα είναι αυτοτελές (ή επισημαίνεται ότι ξεφεύγει από τους στόχους του βιβλίου), δεν χρειάζεται πολλή προετοιμασία στα μαθηματικά για το μεγαλύτερο τμήμα του βιβλίου — αν και η μαθηματική ωριμότητα οπωσδήποτε είναι χρήσιμη.

Χρήση του βιβλίου ως διδακτέας ύλης

Υπάρχει μεγάλη ευελιξία σε ό,τι αφορά τον τρόπο με τον οποίο μπορεί να διδαχθεί η ύλη του βιβλίου· εξαρτάται από τον καθηγητή και την προετοιμασία των σπουδαστών του. Γίνεται επαρκής κάλυψη της βασικής ύλης, έτσι ώστε να μπορεί το βιβλίο να χρησιμοποιηθεί για τη διδασκαλία αρχαρίων στις δομές δεδομένων, ενώ ταυτόχρονα παρέχονται επαρκείς λεπτομέρειες και κάλυψη του προχωρημένου υλικού προκειμένου να μπορεί να χρησιμοποιηθεί και στη διδασκαλία της σχεδίασης και ανάλυσης αλγορίθμων σε σπουδαστές πιο προχωρημένου επιπέδου. Μερικοί καθηγητές μπορεί να προτιμήσουν να δώσουν έμφαση στις υλοποιήσεις και στα πρακτικά θέματα, ενώ κάποιοι άλλοι μπορεί να επιθυμούν να δώσουν έμφαση στην ανάλυση και τις θεωρητικές έννοιες.

Σε ένα εισαγωγικό μάθημα στις δομές δεδομένων και τους αλγορίθμους, θα μπορούσε να δοθεί έμφαση στις βασικές δομές δεδομένων του Μέρους 2 και τη χρήση τους στις υλοποιήσεις των Μερών 3 και 4. Σε ένα μάθημα σχεδίασης και ανάλυσης αλγορίθμων θα μπορούσε να δοθεί έμφαση στο θεμελιώδες υλικό του Μέρους 1 και στο Κεφάλαιο 5, και στη συνέχεια να μελετηθούν οι τρόποι με τους οποίους οι αλγόριθμοι των Μερών 3 και 4 επιτυγχάνουν καλές ασυμπτωτικές επιδόσεις. Σε ένα μάθημα τεχνολογίας λογισμικού, θα μπορούσε να παραλειφθεί η ύλη των μαθηματικών καθώς και η προχωρημένη ύλη των αλγορίθμων, και να δοθεί έμφαση στους τρόπους ολοκλήρωσης των υλοποιήσεων που περιλαμβάνονται στο βιβλίο σε μεγάλα προγράμματα ή συστήματα. Τέλος, σε ένα μάθημα με θέμα τους αλγορίθμους, θα μπορούσε να ακολουθηθεί μια γενική προσέγγιση και να εισαχθούν έννοιες από όλους τους παραπάνω τομείς.

Οι προηγούμενες εκδόσεις του βιβλίου, οι οποίες βασίζονται σε άλλες γλώσσες προγραμματισμού, έχουν χρησιμοποιηθεί σε πάρα πολλά κολέγια και πανεπιστήμια ως διδακτέα ύλη για το δεύτερο ή τρίτο μάθημα στην επιστήμη των υπολογιστών, και ως βοηθητικό εγχειρίδιο για άλλα μαθήματα. Στο Princeton, έχουμε διαπιστώσει ότι η ύλη που καλύπτει το βιβλίο παρέχει στους σπουδαστές μας μια εισαγωγή στην επιστήμη των υπολογιστών η οποία μπορεί να επεκταθεί σε μεταγενέστερα μαθήματα που αφορούν την ανάλυση αλγορίθμων, τον προγραμματισμό συστημάτων, και τη θεωρητική επιστήμη των υπολογιστών· από την άλλη μεριά, παρέχει στην αυξανόμενη ομάδα των σπουδαστών από άλλους επιστημονικούς κλάδους ένα ευρύ σύνολο τεχνικών τις οποίες μπορούν να χρησιμοποιήσουν αμέσως.

Οι ασκήσεις — από τις οποίες σχεδόν όλες είναι καινούργιες σε αυτή την τρίτη έκδοση — ανήκουν σε διάφορους τύπους. Μερικές έχουν στόχο να ελέγξουν αν έγινε κατανοητή η ύλη του βιβλίου, και απλώς ζητούν από τους αναγνώστες να εργαστούν σε κάποιο παράδειγμα ή να εφαρμόσουν βασικές έννοιες που περιγράφονται στο βιβλίο. Άλλες περιλαμβάνουν την υλοποίηση και τη συναρμολόγηση των αλγορίθμων, ή την εκτέλεση πειραματικών μελετών με στόχο τη σύγκριση διαφόρων παραλλαγών των αλγορίθμων και την καλύτερη κατανόηση των ιδιοτήτων τους. Κάποιες άλλες, πάλι, είναι ένας θησαυρός σημαντικών πληροφοριών, σε επίπεδο λεπτομέρειας το οποίο δεν είναι κατάλληλο για το βιβλίο. Η μελέτη και η διερεύνηση των ασκήσεων θα βοηθήσουν σε μεγάλο βαθμό τον αναγνώστη να κατανοήσει τα θέματα που περιγράφονται στο βιβλίο.

Αλγόριθμοι πρακτικής χρήσης

Οποιοσδήποτε θέλει να χρησιμοποιήσει τον υπολογιστή του αποτελεσματικότερα μπορεί να χρησιμοποιήσει αυτό το βιβλίο ως οδηγό αναφοράς ή για αυτοδιδασκαλία. Όσοι έχουν πείρα στον προγραμματισμό μπορούν να βρουν στα κεφάλαια του βιβλίου πληροφορίες που αφορούν συγκεκριμένα θέματα. Ως ένα μεγάλο βαθμό, μπορείτε να διαβάσετε κάθε κεφάλαιο του βιβλίου ανεξάρτητα από τα υπόλοιπα, παρά το γεγονός ότι, σε μερικές περιπτώσεις, ορισμένοι αλγόριθμοι ενός κεφαλαίου χρησιμοποιούν μεθόδους από κάποιο προηγούμενο κεφάλαιο.

Το βιβλίο είναι προσανατολισμένο στη μελέτη των αλγορίθμων που είναι πιθανό να έχουν πρακτική χρήση. Παρέχονται πληροφορίες σχετικά με τα εργαλεία του εμπορίου, έτσι ώστε οι αναγνώστες να μπορούν με βεβαιότητα να υλοποιήσουν, να αποσφαλματώσουν, και να χρησιμοποιήσουν στην πράξη αλγορίθμους για την επίλυση ενός προβλήματος ή την προσθήκη επιπλέον λειτουργιών σε κάποια εφαρμογή. Περιλαμβάνονται πλήρεις υλοποιήσεις των μεθόδων που εξετάζονται, όπως επίσης και περιγραφές των λειτουργιών αυτών των προγραμμάτων σε ένα συνεπές σύνολο παραδειγμάτων.

Επειδή εργαζόμαστε με πραγματικό κώδικα αντί να γράφουμε ψευδοκώδικα, μπορείτε να χρησιμοποιήσετε αυτά τα προγράμματα στην πράξη πολύ γρήγορα. Τα προγράμματα είναι διαθέσιμα και στην ηλεκτρονική σελίδα του βιβλίου στο Διαδίκτυο. Μπορείτε να χρησιμοποιήσετε αυτά τα προγράμματα με πολλούς τρόπους προκειμένου να σας βοηθήσουν στη μελέτη των αλγορίθμων. Διαβάστε τα για να ελέγξετε πόσο έχετε κατανοήσει τις λεπτομέρειες ενός αλγορίθμου, ή για να δείτε κάποιον τρόπο χειρισμού ανάθεσης αρχικών τιμών, οριακών συνθηκών, και άλλων περίεργων καταστάσεων που αποτελούν συχνά προγραμματιστικές προκλήσεις. Εκτελέστε τα για να δείτε τους αλγορίθμους σε δράση, να μελετήσετε πειραματικά τις επιδόσεις τους, και να ελέγξετε τα αποτελέσματά σας σε σχέση με τους πίνακες που περιέχονται στο βιβλίο, ή να δοκιμάσετε τις δικές σας τροποποιήσεις.

Στο βιβλίο εξετάζονται λεπτομερώς τα χαρακτηριστικά των αλγορίθμων και των περιπτώσεων στις οποίες μπορεί να είναι χρήσιμοι, ενώ παράλληλα γίνονται συνδέσεις με την ανάλυση αλγορίθμων και τη θεωρητική επιστήμη των υπολογιστών. Όποτε κρίνεται απαραίτητο, παρουσιάζονται πειραματικά και αναλυτικά αποτελέσματα προκειμένου να διασαφηνιστεί ο λόγος για τον οποίο προτιμούνται κάποιοι συγκεκριμένοι αλγόριθμοι. Όποτε παρουσιάζει ενδιαφέρον, περιγράφεται και η σχέση των πρακτικών αλγορίθμων που εξετάζονται με αμιγώς θεωρητικά αποτελέσματα. Επίσης, οι ειδικές πληροφορίες σχετικά με τα χαρακτηριστικά επιδόσεων των αλγορίθμων και των υλοποιήσεων συγκεντρώνονται, ενσωματώνονται στο κείμενο, και αναλύονται σε ολόκληρο το βιβλίο.

Γλώσσα προγραμματισμού

Η γλώσσα προγραμματισμού που χρησιμοποιείται για όλες τις υλοποιήσεις των αλγορίθμων είναι η Java. Στα προγράμματα χρησιμοποιείται μια ευρεία κλίμακα καθιερωμένων ιδιωματισμών της Java, και στο βιβλίο περιλαμβάνονται περιεκτικές περιγραφές κάθε κατασκευής.

Ο Mike Schidlowsky και εγώ αναπτύξαμε ένα στυλ προγραμματισμού σε Java το οποίο βασίζεται σε αφηρημένους τύπους δεδομένων· νομίζουμε ότι είναι ένας αποτελεσματικός τρόπος για να παρουσιάζουμε τους αλγορίθμους και τις δομές δεδομένων ως πραγματικά προγράμματα. Κάναμε αρκετή προσπάθεια για να δώσουμε κομψές, συμπαγείς, αποδοτικές, και φορητές υλοποιήσεις. Το στυλ παρουσιάζει συνέπεια όποτε αυτό είναι δυνατό, έτσι ώστε τα προγράμματα που είναι παρόμοια να φαίνονται και παρόμοια.

Στην περίπτωση πολλών από τους αλγορίθμους του βιβλίου, οι ομοιότητες ισχύουν ανεξάρτητα από τη γλώσσα: ο αλγόριθμος ταξινόμησης quicksort είναι πάντοτε ο αλγόριθμος ταξινόμησης quicksort (για να επιλέξουμε ένα εξέχον παράδειγμα), ανεξάρτητα από το αν εκφράζεται σε Ada, Algol-60, Basic, C, C++, Fortran, Java, Mesa, Modula-3, Pascal, Post-

Script, Smalltalk, ή σε οποιαδήποτε από τις αμέτρητες άλλες γλώσσες προγραμματισμού και περιβάλλοντα όπου έχει αποδειχθεί ότι είναι μια αποτελεσματική μέθοδος ταξινόμησης. Από τη μια πλευρά, ο κώδικάς μας παίρνει πληροφορίες από τις εμπειρίες υλοποίησης αλγορίθμων σε αυτές και σε πολλές άλλες γλώσσες προγραμματισμού (υπάρχουν εκδόσεις του βιβλίου και για τις γλώσσες C και C++)· από την άλλη, ορισμένες από τις ιδιότητες μερικών από αυτές τις γλώσσες εμπλουτίζονται από την εμπειρία των σχεδιαστών τους σε μερικούς από τους αλγορίθμους και τις δομές δεδομένων που εξετάζουμε σε αυτό το βιβλίο.

Το Κεφάλαιο 1 αποτελεί λεπτομερές παράδειγμα αυτής της προσέγγισης στην ανάπτυξη αποδοτικών υλοποιήσεων των αλγορίθμων μας σε Java, ενώ στο Κεφάλαιο 2 περιγράφεται η προσέγγιση που ακολουθούμε για την ανάλυσή τους. Τα Κεφάλαια 3 και 4 είναι αφιερωμένα στην περιγραφή και την αιτιολόγηση των βασικών μηχανισμών που χρησιμοποιούμε στις υλοποιήσεις τύπων δεδομένων και αφηρημένων τύπων δεδομένων (ΑΤΔ). Αυτά τα τέσσερα κεφάλαια αποτελούν τα θεμέλια για τα επόμενα κεφάλαια του βιβλίου.

Ευχαριστίες

Η ανταπόκριση πολλών ανθρώπων από τις παλιότερες εκδόσεις αυτού του βιβλίου ήταν ιαδιαίτερα χρήσιμη. Πιο συγκεκριμένα, εκατοντάδες φοιτητές των πανεπιστημίων Princeton και Brown υπέστησαν τις προκαταρκτικές εκδόσεις για αρκετά χρόνια. Θα ήθελα να εκφράσω ιδιαίτερες ευχαριστίες στην Trina Avery και τον Tom Freeman για τη βοήθειά τους στη δημιουργία της πρώτης έκδοσης· στην Janet Incerpi για τη δημιουργικότητα και την εξυπνάδα της που της επέτρεψαν να πειθαναγκάσει το πρωτόγονο υλικό και λογισμικό συγγραφής κειμένου που χρησιμοποιούσαμε προκειμένου να δημιουργήσουν την πρώτη έκδοση· στον Marc Brown για τη συμμετοχή του στην έρευνα οπτικοποίησης των αλγορίθμων, η οποία οδήγησε στη δημιουργία πολλών από τις εικόνες του βιβλίου· και στους Dave Hanson και Andrew Appel για την προθυμία που έδειξαν να απαντήσουν σε όλες τις ερωτήσεις μου σχετικά με τις γλώσσες προγραμματισμού. Θα ήθελα επίσης να ευχαριστήσω όλους τους αναγνώστες που έκαναν σχόλια σχετικά με τις διάφορες εκδόσεις του βιβλίου, μεταξύ των οποίων είναι και οι Guy Almes, Jon Bentley, Marc Brown, Jay Gischer, Allan Heydon, Kennedy Lemke, Udi Manber, Dana Richards, John Reif, M. Rosenfeld, Stephen Seidman, Michael Quinn, και William Ward.

Για να δημιουργήσω αυτή τη νέα έκδοση είχα την ευχαρίστηση να συνεργαστώ με τους Peter Gordon και Helen Goldstein της Addison-Wesley, οι οποίοι και καθοδήγησαν με υπομονή αυτό το έργο καθώς εξελισσόταν. Επίσης, είχα την ευχαρίστηση να συνεργαστώ με πολλά άλλα μέλη του εξειδικευμένου προσωπικού της Addison-Wesley. Η φύση του έργου έκανε το βιβλίο μια, κατά κάποιον τρόπο, ασυνήθιστη πρόκληση για πολλούς από αυτούς, και εκτιμώ ιδιαίτερα την υπομονή τους. Πιο συγκεκριμένα, η Marilyn Rash έκανε εξαιρετική δουλειά στη διαχείριση της έκδοσης αυτού του βιβλίου μέσα σε ένα πολύ συμπιεσμένο χρονοδιάγραμμα.

Κέρδισα και τρεις νέους μέντορες κατά τη συγγραφή αυτού του βιβλίου, και θα ήθελα να εκφράσω ιδιαίτερα την εκτίμησή μου σε αυτούς. Καταρχάς, ο Steve Summit έλεγξε προσεκτικά τις πρώτες εκδόσεις του βιβλίου σε τεχνικό επίπεδο και έκανε κυριολεκτικά χιλιάδες λεπτομερή σχόλια, ιδιαίτερα για τα προγράμματα. Ο Steve κατάλαβε ξεκάθαρα το στόχο μου να δώσω κομψές, αποδοτικές, και αποτελεσματικές υλοποιήσεις, και τα σχόλιά του, όχι μόνο

με βοήθησαν να προσφέρω ένα μέτρο συνέπειας κατά τις υλοποιήσεις, αλλά και να βελτιώσω σημαντικά πολλές από αυτές. Έπειτα, η Lyn Dupré έκανε και αυτή χιλιάδες λεπτομερή σχόλια, τα οποία ήταν ανεκτίμητης αξίας και με βοήθησαν, όχι μόνο να διορθώσω και να αποφύγω γραμματικά λάθη, αλλά επίσης — που είναι ακόμη σημαντικότερο — να βρω ένα περιεκτικό και συνεπές στυλ συγγραφής το οποίο με βοήθησε να συναρμολογήσω το μεγάλο όγκο του τεχνικού υλικού. Τέλος, ο Chris Van Wyk, μέσα από μια μεγάλη σειρά εμπνευσμένων ανταλλαγών μηνυμάτων ηλεκτρονικού ταχυδρομείου, υπερασπίστηκε με υπομονή τους βασικούς κανόνες του αντικειμενοστρεφούς προγραμματισμού και με βοήθησε να αναπτύξω ένα στυλ συγγραφής κώδικα τέτοιο ώστε οι αλγόριθμοι να παρουσιάζονται με σαφήνεια και ακρίβεια, με ταυτόχρονη εκμετάλλευση των στοιχείων που μπορεί να προσφέρει ο αντικειμενοστρεφής προγραμματισμός. Η βασική προσέγγιση που αναπτύξαμε για την έκδοση του βιβλίου σε C++ επηρέασε σημαντικά τον κώδικα της Java και είναι βέβαιο ότι θα επηρεάσει και τους μελλοντικούς τόμους και στις δύο γλώσσες (όπως επίσης και στη C). Είμαι εξαιρετικά ευγνώμων για την ευκαιρία που μου δόθηκε να μάθω πολλά πράγματα από τους Steve, Lyn, και Chris — η προσφορά τους ήταν ζωτικής σημασίας για την ανάπτυξη αυτού του βιβλίου.

Πολλά από αυτά που γράφονται στο βιβλίο τα έμαθα από τη διδασκαλία και τα γραπτά του Don Knuth, που ήταν ο επιβλέπων καθηγητής μου στο Πανεπιστήμιο Stanford. Παρά το γεγονός ότι ο Don δεν είχε άμεση επίδραση σε αυτή τη δουλειά, η παρουσία του γίνεται αισθητή σε αυτό το βιβλίο επειδή ήταν εκείνος που έθεσε τη μελέτη των αλγορίθμων στην επιστημονική βάση η οποία καθιστά δυνατή μια δουλειά σαν αυτή. Ο φίλος και συνάδελφός μου Philippe Flajolet, ο οποίος αποτέλεσε βασικό παράγοντα για την ανάπτυξη της ανάλυσης αλγορίθμων ως ενός ώριμου τομέα ερευνών, είχε και αυτός παρόμοια επίδραση σε αυτό το έργο.

Θα ήθελα να εκφράσω τις βαθιές μου ευχαριστίες για την υποστήριξη που είχα από το Πανεπιστήμιο Princeton, το Πανεπιστήμιο Brown, και το Εθνικό Ινστιτούτο Έρευνας στην Πληροφορική και τον Αυτοματισμό (Institut National de Recherche en Informatique et Automatique, INRIA), όπου έκανα το μεγαλύτερο μέρος της έρευνας για το βιβλίο· το ίδιο ισχύει και για το Ινστιτούτο Αναλύσεων Άμυνας (Institute for Defense Analyses) καθώς και για το Ερευνητικό Κέντρο της Xerox στο Palo Alto (Xerox Palo Alto Research Center), όπου έκανα ένα μέρος της δουλειάς όσο ήμουν επισκέπτης. Πολλά μέρη του βιβλίου βασίζονται σε έρευνες που υποστηρίχθηκαν γενναιόδωρα από το Εθνικό Ίδρυμα Ερευνών (National Science Foundation) και το Γραφείο Ναυτικών Ερευνών (Office of Naval Research). Τέλος, ευχαριστώ τους Bill Bowen, Aaron Lemonick, και Neil Rudenstine για την υποστήριξή τους στη δημιουργία ακαδημαϊκού περιβάλλοντος στο Princeton, στο οποίο είχα τη δυνατότητα να ετοιμάσω αυτό το βιβλίο παρά τις πολλές άλλες υποχρεώσεις μου.

Robert Sedgewick
Marly-le-Roi, Γαλλία, 1983
Princeton, New Jersey, 1990, 1992
Jamestown, Rhode Island, 1997
Princeton, New Jersey, 1998, 2002

Πρόλογος συμβούλου σε θέματα Java

Την προηγούμενη δεκαετία, η Java έγινε προτιμώμενη γλώσσα για μια ποικιλία εφαρμογών. Όμως, οι προγραμματιστές Java πολλές φορές έπρεπε να ανατρέχουν σε βιβλία όπως το βιβλίο *Αλγόριθμοι σε C* του Sedgewick για την επίλυση διάφορων προβλημάτων προγραμματισμού. Για πολύ καιρό υπήρχε κενός χώρος στη βιβλιοθήκη κάθε αναγνώστη για ένα ισοδύναμο έργο αναφοράς σε Java· αυτό το βιβλίο ήρθε να καλύψει το κενό.

Γράψαμε τα δείγματα προγραμμάτων ως μεθόδους γενικής χρήσης, έτσι ώστε να μπορούν να χρησιμοποιηθούν σε μια ποικιλία συναφών πλαισίων. Γι' αυτόν το λόγο, δεν χρησιμοποιήσαμε το μηχανισμό πακέτων της Java. Για να εστιάσουμε στους αλγορίθμους που εξετάζαμε (και για να αποκαλύψουμε την αλγοριθμική βάση πολλών θεμελιωδών κλάσεων βιβλιοθήκης), αποφύγαμε την καθιερωμένη βιβλιοθήκη της Java ευνοώντας πιο θεμελιώδεις τύπους. Ο κατάλληλος έλεγχος σφαλμάτων και η χρήση άλλων αμυντικών τακτικών θα οδηγούσαν σε σημαντική διόγκωση του κώδικα και θα αποσπούσαν τον αναγνώστη από τον πυρήνα των αλγορίθμων· έτσι, οι ομάδες ανάπτυξης λογισμικού θα πρέπει να προσθέτουν τέτοιου είδους κώδικα όταν χρησιμοποιούν τα προγράμματα σε μεγαλύτερες εφαρμογές.

Παρά το γεγονός ότι οι αλγόριθμοι που παρουσιάζουμε είναι ανεξάρτητοι από τη γλώσσα προγραμματισμού, έχουμε δώσει μεγάλη προσοχή στα θέματα επιδόσεων που αφορούν ειδικά την Java. Οι χρονομετρήσεις που περιλαμβάνονται σε ολόκληρο το βιβλίο παρέχονται ως ένα πλαίσιο για τη σύγκριση διαφορετικών αλγορίθμων, και ποικίλλουν ανάλογα με την εικονική μηχανή. Καθώς εξελίσσονται τα περιβάλλοντα Java, τα προγράμματα θα εκτελούνται το ίδιο γρήγορα όπως και ο εγγενώς μεταγλωττισμένος κώδικας· ωστόσο, αυτού του είδους οι βελτιστοποιήσεις δεν πρόκειται να αλλάξουν τις σχετικές μεταξύ των αλγορίθμων επιδόσεις. Έτσι, παρέχουμε τις μετρήσεις ως χρήσιμη αναφορά για τέτοιου είδους συγκρίσεις.

Θα ήθελα να ευχαριστήσω τον Mike Zamansky, για τη σοφή καθοδήγηση και την αφοσίωσή του στη διδασκαλία της επιστήμης των υπολογιστών, καθώς και τους Daniel Chaskes, Jason Sanders, και James Percy, για την αμέριστη υποστήριξή τους. Θα ήθελα επίσης να ευχαριστήσω την οικογένειά μου για την υποστήριξή της και για τον υπολογιστή που μου πρόσφερε, στον οποίο και δημιούργησα τα πρώτα μου προγράμματα. Ο συγκερασμός της Java με τους κλασικούς αλγορίθμους της επιστήμης των υπολογιστών αποτέλεσε συναρπαστική προσπάθεια για την οποία νιώθω πολύ ευχαριστημένος. Bob, σε ευχαριστώ για την ευκαιρία που μου έδωσες.

Michael Schidlowsky
Oakland Gardens, New York, 2002

5.4 Δένδρα

Τα δένδρα είναι μια μαθηματική αφαίρεση η οποία παίζει κεντρικό ρόλο στη σχεδίαση και την ανάλυση των αλγορίθμων επειδή

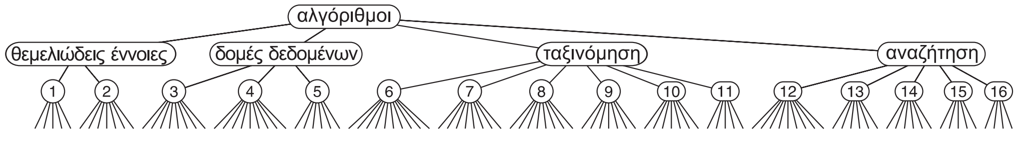
- Χρησιμοποιούμε δένδρα για να περιγράψουμε τις δυναμικές ιδιότητες των αλγορίθμων.
- Κατασκευάζουμε και χρησιμοποιούμε ρητές δομές δεδομένων που αποτελούν συμπαγείς υλοποιήσεις δένδρων.

Έχουμε ήδη συναντήσει παραδείγματα και των δύο αυτών χρήσεων. Στο Κεφάλαιο 1 σχεδιάσαμε για το πρόβλημα της συνδετικότητας αλγορίθμους οι οποίοι βασίζονταν σε δομές δένδρου, και στις Ενότητες 5.2 και 5.3 περιγράψαμε τη δομή κλήσης των αναδρομικών αλγορίθμων με δομές δένδρου.

Δένδρα συναντάμε συχνά στην καθημερινή μας ζωή — η βασική έννοια μας είναι γνωστή. Για παράδειγμα, πολλοί άνθρωποι καταγράφουν τους προγόνους ή τους απογόνους τους σε κάποιο οικογενειακό δένδρο· όπως θα δούμε, μεγάλο μέρος της ορολογίας που χρησιμοποιούμε προέρχεται από αυτή τη χρήση των δένδρων. Ένα άλλο παράδειγμα εντοπίζεται στην οργάνωση αγώνων σε διάφορα αθλήματα· αυτή η χρήση μελετήθηκε, μεταξύ άλλων, από τον Lewis Carroll. Ένα τρίτο παράδειγμα εντοπίζεται στα οργανογράμματα μεγάλων οργανισμών· αυτή η χρήση βασίζεται στην ιεραρχική αποσύνθεση που χαρακτηρίζει τους αλγορίθμους "διαίρει και βασίλευε". Ένα τέταρτο παράδειγμα είναι κάποιο δένδρο συντακτικής ανάλυσης μιας πρότασης στα μέρη που την αποτελούν· τέτοια δένδρα σχετίζονται στενά με την επεξεργασία των γλωσσών υπολογιστών, όπως αναπτύσσεται στο Μέρος 6. Στην Εικόνα 5.19 παρουσιάζεται ένα τυπικό παράδειγμα δένδρου — το οποίο περιγράφει τη δομή αυτού του βιβλίου. Σε ολόκληρο το βιβλίο εξετάζουμε πολλά άλλα παραδείγματα εφαρμογών των δένδρων.

Στις εφαρμογές για υπολογιστές, μια από τις γνωστότερες χρήσεις των δομών δένδρου είναι στην οργάνωση του συστήματος αρχείων. Τοποθετούμε τα αρχεία σε *φακέλους* (folders) ή *καταλόγους* (directories) οι οποίοι ορίζονται αναδρομικώς ως ακολουθίες φακέλων και αρχείων. Αυτός ο αναδρομικός ορισμός αντιστοιχεί και πάλι σε μια φυσική αναδρομική αποσύνθεση και είναι όμοιος με τον ορισμό ενός συγκεκριμένου τύπου δένδρου.

Υπάρχουν πολλοί και διάφοροι τύποι δένδρων, και είναι σημαντικό να κατανοήσουμε τη διάκριση ανάμεσα στην αφαίρεση (abstraction) και τη συγκεκριμένη (concrete) αναπαράσταση την οποία χρησιμοποιούμε σε μια δεδομένη εφαρμογή. Αντίστοιχα, θα εξετάσουμε τους διαφορετικούς τύπους δένδρων και τις αναπαραστάσεις τους με λεπτομέρειες.



Εικόνα 5.19
Ένα δένδρο

Σε αυτό το δένδρο παρουσιάζονται τα μέρη, τα κεφάλαια, και οι ενότητες αυτού του βιβλίου. Υπάρχει ένας κόμβος για κάθε οντότητα. Κάθε κόμβος συνδέεται τόσο με τα μέρη από τα οποία αποτελείται, με συνδέσμους που "κατεβαίνουν" προς αυτά, όσο και με το μεγαλύτερο μέρος στο οποίο ανήκει, με ένα σύνδεσμο που "ανεβαίνει" και φτάνει μέχρι το συγκεκριμένο μέρος.

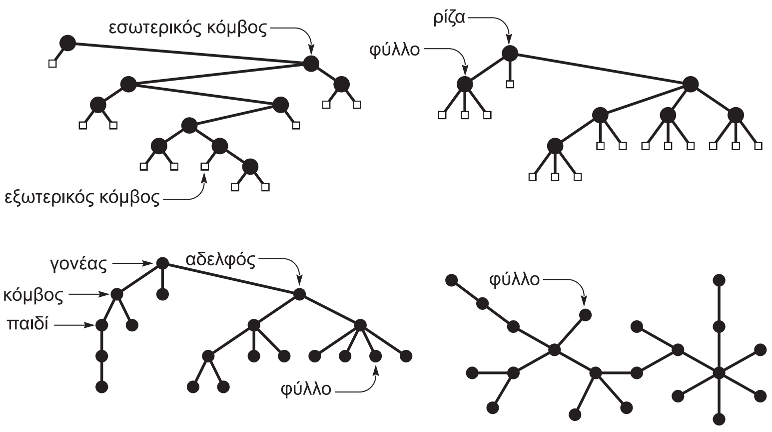
Θα ξεκινήσουμε την ανάλυσή μας ορίζοντας τα δένδρα ως αφηρημένα αντικείμενα και εισάγοντας τους περισσότερους σχετικούς βασικούς όρους. Θα εξετάσουμε ανεπίσημα τους διαφορετικούς τύπους δένδρων που θα μας απασχολήσουν, σε φθίνουσα σειρά γενικότητας:

- Δένδρα
- Δένδρα με ρίζα
- Διατεταγμένα δένδρα
- Μ-αδικά και δυαδικά δένδρα

Αφού αναπτύξουμε κάποιο πλαίσιο με αυτή τη γενική ανάλυση, θα προχωρήσουμε σε επίσημους ορισμούς και θα εξετάσουμε αναπαραστάσεις και εφαρμογές. Στην Εικόνα 5.20 φαίνονται πολλές από τις βασικές έννοιες που θα μας απασχολήσουν.

Ένα *δένδρο* (tree) είναι μια μη κενή συλλογή κορυφών και ακμών που ικανοποιεί συγκεκριμένες απαιτήσεις. Μια *κορυφή* (vertex) είναι ένα απλό αντικείμενο — που αποκαλούμε επίσης *κόμβο* (node) — το οποίο μπορεί να έχει κάποιο όνομα και να περιλαμβάνει και άλλες σχετικές πληροφορίες, ενώ μια *ακμή* (edge) αποτελεί το σύνδεσμο μεταξύ δύο κορυφών. Μια *διαδρομή* (path) δένδρου είναι μια λίστα από διακεκριμένες κορυφές στην οποία οι διαδοχικές κορυφές του δένδρου συνδέονται με ακμές. Η καθοριστική ιδιότητα ενός δένδρου είναι ότι υπάρχει ακριβώς μία διαδρομή που να συνδέει δύο οποιουσδήποτε κόμβους. Αν υπάρχουν περισσότερες από μία διαδρομές μεταξύ ενός ζεύγους κόμβων, ή αν δεν υπάρχει καμία διαδρομή, τότε αυτό που έχουμε είναι γράφος και όχι δένδρο. Ένα σύνολο ξένων μεταξύ τους δένδρων ονομάζεται *δάσος* (forest).

Ένα *δένδρο με ρίζα* (rooted tree) είναι κάποιο δένδρο στο οποίο καθορίζουμε έναν κόμβο ως *ρίζα* (root) του δένδρου. Στην επιστήμη των υπολογιστών, με τον όρο *δένδρο* αναφερόμαστε συνήθως αποκλειστικά σε δένδρα με ρίζα, και χρησιμοποιούμε τον όρο *ελεύθερο δένδρο* (free tree) για να αναφερόμαστε στη γενικότερη δομή που περιγράψαμε στην προηγούμενη παράγραφο. Σε ένα δένδρο με ρίζα, κάθε κόμβος του αποτελεί ρίζα κάποιου *υποδένδρου* (subtree) το οποίο αποτελείται από το συγκεκριμένο κόμβο και τους κόμβους που βρίσκονται κάτω από αυτόν.



Εικόνα 5.20
Τύποι δένδρων

Σε αυτά τα διαγράμματα φαίνονται παραδείγματα ενός δυαδικού δένδρου (επάνω αριστερά), ενός τριαδικού δένδρου (επάνω δεξιά), ενός δένδρου με ρίζα (κάτω αριστερά), και ενός ελεύθερου δένδρου (κάτω δεξιά).

Υπάρχει ακριβώς μία διαδρομή μεταξύ της ρίζας και καθενός από τους υπόλοιπους κόμβους του δένδρου. Ο ορισμός υποθέτει ότι οι ακμές δεν έχουν κατεύθυνση· συνήθως, θεωρούμε ότι όλες οι ακμές δείχνουν "μακριά από τη ρίζα" ή "προς τη ρίζα", ανάλογα με την εκάστοτε εφαρμογή. Τα δένδρα με ρίζα τα σχεδιάζουμε συνήθως με τη ρίζα στην κορυφή (αν και αυτή η συνθήκη φαίνεται αφύσικη με μια πρώτη ματιά) και λέμε ότι ο κόμβος y βρίσκεται *κάτω* από τον κόμβο x (και ο x *επάνω* από τον y) αν ο x βρίσκεται στη διαδρομή από το y προς τη ρίζα (δηλαδή, αν το y βρίσκεται κάτω από το x, όπως φαίνεται στη σελίδα, και συνδέεται με το x μέσω μιας διαδρομής η οποία δεν περνά από τη ρίζα). Κάθε κόμβος (εκτός από τη ρίζα) έχει ακριβώς έναν κόμβο επάνω από αυτόν, ο οποίος ονομάζεται *γονέας* (parent)· οι κόμβοι ακριβώς κάτω από κάποιο κόμβο ονομάζονται *τέκνα* ή *παιδιά* (children). Μερικές φορές, επεκτείνουμε περισσότερο την αναλογία με τα οικογενειακά δένδρα και αναφερόμαστε στον *παππού* (grandparent) ή στον (*αμφιθαλή*) *αδελφό* (sibling) κάποιου κόμβου.

Οι κόμβοι χωρίς παιδιά ονομάζονται *φύλλα* (leaves) ή *τερματικοί* (terminal) κόμβοι. Σε αντιστοιχία προς τη δεύτερη χρήση, οι κόμβοι με τουλάχιστον ένα παιδί ονομάζονται μερικές φορές *μη τερματικοί* (nonterminal) κόμβοι. Είδαμε ένα παράδειγμα της χρησιμότητας της διάκρισης μεταξύ αυτών των τύπων κόμβων σε αυτό το κεφάλαιο. Στα δένδρα που χρησιμοποιούμε για να παρουσιάσουμε τη δομή της κλήσης αναδρομικών αλγορίθμων (δείτε, για παράδειγμα, την Εικόνα 5.14) οι μη τερματικοί κόμβοι (κύκλοι) αναπαριστούν κλήσεις μεθόδων με αναδρομή και οι τερματικοί κόμβοι (τετράγωνα) αναπαριστούν κλήσεις μεθόδων χωρίς αναδρομή.

Σε ορισμένες εφαρμογές, ο τρόπος με τον οποίο είναι διατεταγμένα τα παιδιά κάθε κόμβου είναι σημαντικός· σε κάποιες άλλες εφαρμογές, δεν είναι. *Διατεταγμένο* (ordered) δένδρο είναι ένα δένδρο με ρίζα στο οποίο η σειρά των παιδιών κάθε κόμβου είναι καθορισμένη. Τα διατεταγμένα δένδρα αποτελούν φυσική αναπαράσταση — για παράδειγμα, τοποθετούμε τα παιδιά με κάποια σειρά όταν σχεδιάζουμε ένα οικογενειακό δένδρο. Όπως θα δούμε, αυτή η διάκριση είναι επίσης σημαντική όταν εξετάζουμε την αναπαράσταση των δένδρων σε υπολογιστή.

Αν κάθε κόμβος *πρέπει* να έχει υποχρεωτικά κάποιο καθορισμένο πλήθος παιδιών που να εμφανίζονται με συγκεκριμένη σειρά, τότε έχουμε την περίπτωση ενός M-αδικού (M-ary) δένδρου. Σε ένα τέτοιο δένδρο, συχνά είναι σωστό να ορίζουμε ειδικούς εξωτερικούς κόμβους να μην έχουν παιδιά, έτσι ώστε αυτοί οι εξωτερικοί κόμβοι να μπορούν να συμπεριφέρονται ως ψευδο-κόμβοι στους οποίους να αναφέρονται οι κόμβοι που δεν διαθέτουν το καθορισμένο πλήθος παιδιών. Πιο συγκεκριμένα, ο απλούστερος τύπος M-αδικού δένδρου είναι το δυαδικό δένδρο. *Δυαδικό δένδρο* (binary tree) είναι ένα διατεταγμένο δένδρο που αποτελείται από δύο τύπους κόμβων: εξωτερικούς κόμβους χωρίς παιδιά και εσωτερικούς κόμβους με ακριβώς δύο παιδιά ο καθένας. Αφού τα δύο παιδιά κάθε εσωτερικού κόμβου είναι διατεταγμένα, αναφερόμαστε στο *αριστερό παιδί* (left child) και στο *δεξιό παιδί* (right child) των εσωτερικών κόμβων — κάθε εσωτερικός κόμβος πρέπει να έχει και αριστερό και δεξιό παιδί, παρόλο που ένα από τα δύο ή και τα δύο μπορούν να είναι εξωτερικοί κόμβοι. Ένα *φύλλο* (leaf) σε ένα M-αδικό δένδρο είναι ένας εσωτερικός κόμβος του οποίου τα παιδιά είναι εξωτερικοί κόμβοι.

Αυτή είναι η βασική ορολογία. Στη συνέχεια, θα δούμε τους επίσημους ορισμούς, τις αναπαραστάσεις, και τις εφαρμογές των παρακάτω δομών, σε σειρά αύξουσας γενικότητας:

- Δυαδικά και Μ-αδικά δένδρα
- Διατεταγμένα δένδρα
- Δένδρα με ρίζα
- Ελεύθερα δένδρα

Η παραπάνω σειρά σημαίνει ότι ένα δυαδικό δένδρο αποτελεί ειδικό τύπο διατεταγμένου δένδρου, ένα διατεταγμένο δένδρο αποτελεί ειδικό τύπο δένδρου με ρίζα, και ένα δένδρο με ρίζα αποτελεί ειδικό τύπο ενός ελεύθερου δένδρου. Οι διαφορετικοί τύποι δένδρων εμφανίζονται σε πολλές εφαρμογές, και είναι σημαντικό να γνωρίζουμε τις διαφορές όταν αναζητάμε τρόπους αναπαράστασης δένδρων με συγκεκριμένες δομές δεδομένων. Όπως θα γίνει φανερό, το γεγονός ότι ξεκινάμε από την ειδικότερη αφηρημένη δομή μας επιτρέπει να εξετάσουμε αναλυτικά συγκεκριμένες αναπαραστάσεις της.

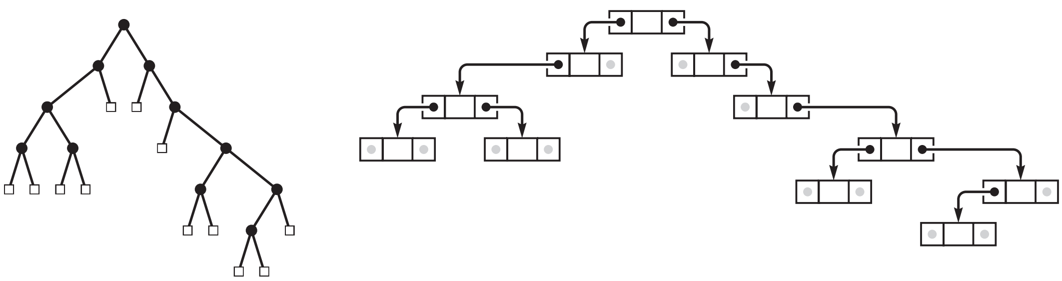
Ορισμός 5.1 *Ένα* ***δυαδικό δένδρο*** *(binary tree) είναι είτε κάποιος εξωτερικός κόμβος είτε κάποιος εσωτερικός κόμβος που συνδέεται με ένα ζεύγος δυαδικών δένδρων, τα οποία ονομάζονται αριστερό και δεξιό υποδένδρο αυτού του κόμβου.*

Αυτός ο ορισμός διασαφηνίζει το γεγονός ότι το ίδιο το δυαδικό δένδρο αποτελεί μια αφηρημένη μαθηματική έννοια. Όταν έχουμε να κάνουμε με κάποια αναπαράσταση στον υπολογιστή, εργαζόμαστε απλώς με μια συγκεκριμένη υλοποίηση αυτού του αφηρημένου τύπου. Η κατάσταση δεν είναι διαφορετική από την αναπαράσταση πραγματικών αριθμών με τύπους `float`, ακεραίων με τύπους `int`, και ούτω καθεξής. Όταν σχεδιάζουμε ένα δένδρο με τον κόμβο στη ρίζα να συνδέεται με ακμές με το αριστερό υποδένδρο στα αριστερά και το δεξιό υποδένδρο στα δεξιά, επιλέγουμε μια βολική και συγκεκριμένη αναπαράσταση. Υπάρχουν πολλοί διαφορετικοί τρόποι να αναπαραστήσουμε δυαδικά δένδρα (δείτε, για παράδειγμα, την Άσκηση 5.62), οι οποίοι μπορεί αρχικά να μας προκαλέσουν έκπληξη· ωστόσο, τέτοιες αναπαραστάσεις πρέπει να τις αναμένουμε, με δεδομένη την αφηρημένη φύση του ορισμού.

Η συγκεκριμένη αναπαράσταση που χρησιμοποιούμε συχνότερα όταν υλοποιούμε προγράμματα τα οποία χρησιμοποιούν και χειρίζονται δυαδικά δένδρα είναι μια δομή με δύο συνδέσμους (έναν αριστερό και ένα δεξιό σύνδεσμο) για τους εσωτερικούς κόμβους (δείτε την Εικόνα 5.21). Αυτές οι δομές είναι παρόμοιες με τις συνδεδεμένες λίστες αλλά έχουν δύο συνδέσμους ανά κόμβο, αντί για έναν. Οι μηδενικοί σύνδεσμοι αντιστοιχούν στους εξωτερικούς κόμβους. Ειδικότερα, προσθέτουμε έναν κόμβο στην καθιερωμένη μας αναπαράσταση συνδεδεμένης λίστας της Ενότητας 3.3, όπως φαίνεται στο επόμενο τμήμα κώδικα:

```
class Node
  { Item item; Node l; Node r;
    Node(Item v, Node l, Node r)
      { this.item = v; this.l = l; this.r = r; }
  }
```

το οποίο δεν είναι τίποτε περισσότερο από τον κώδικα Java για τον Ορισμό 5.1. Ένας κόμβος αποτελείται από ένα στοιχείο και ένα ζεύγος αναφορών σε κόμβους (σύνδεσμοι). Συνεπώς, μπορούμε, για παράδειγμα, να υλοποιήσουμε την αφηρημένη πράξη *μετακίνηση στο αριστερό υποδένδρο* με μια ανάθεση αναφοράς σαν την `x = x.l`.



Εικόνα 5.21
Αναπαράσταση δυαδικού δένδρου
Στην καθιερωμένη αναπαράσταση ενός δυαδικού δένδρου χρησιμοποιούμε κόμβους με δύο συνδέσμους: έναν αριστερό σύνδεσμο με το αριστερό υποδένδρο και ένα δεξιό σύνδεσμο με το δεξιό υποδένδρο. Οι μηδενικοί σύνδεσμοι αντιστοιχούν στους εξωτερικούς κόμβους.

Αυτή η καθιερωμένη αναπαράσταση μας επιτρέπει την αποδοτική υλοποίηση των λειτουργιών που χρησιμοποιούνται για τη μετακίνηση *προς τα κάτω* στο δένδρο ξεκινώντας από τη ρίζα, όχι όμως και για τις λειτουργίες που χρησιμοποιούνται για τη μετακίνηση *προς τα επάνω* στο δένδρο από κάποιο παιδί προς το γονέα του. Για τους αλγορίθμους που απαιτούν τέτοιου είδους λειτουργίες, μπορούμε να προσθέσουμε σε κάθε κόμβο έναν τρίτο σύνδεσμο που να δείχνει στο γονέα του κόμβου. Αυτή η εναλλακτική υλοποίηση είναι ανάλογη με τις διπλά συνδεδεμένες λίστες. Όπως και με τις συνδεδεμένες λίστες (δείτε την Εικόνα 3.8), μπορούμε να τηρούμε τους κόμβους του δένδρου σε κάποιον πίνακα και να χρησιμοποιούμε ως συνδέσμους αριθμοδείκτες αντί για αναφορές, αλλά γενικά δεν ασχολούμαστε με βελτιστοποιήσεις σε τόσο χαμηλό επίπεδο (παρά μόνο για να επισημάνουμε την ύπαρξή τους) επειδή η αποτελεσματικότητά τους εξαρτάται από το εκάστοτε σύστημα. Χρησιμοποιούμε και άλλες αναπαραστάσεις δυαδικών δένδρων για ορισμένους ειδικούς αλγορίθμους, οι οποίοι περιγράφονται κυρίως στο Κεφάλαιο 9.

Εξαιτίας όλων των διαφορετικών δυνατών αναπαραστάσεων, θα μπορούσαμε να αναπτύξουμε έναν αφηρημένο τύπο δεδομένων (ΑΤΔ) δυαδικού δένδρου που να ενθυλακώνει και τις δύο σημαντικές λειτουργίες τις οποίες θέλουμε να εκτελεί, και να διαχωρίζει τη χρήση και την υλοποίηση από αυτές τις λειτουργίες. Ωστόσο, δεν πρόκειται να ακολουθήσουμε αυτή την προσέγγιση στο βιβλίο επειδή

- Τις περισσότερες φορές χρησιμοποιούμε την αναπαράσταση με τους δύο συνδέσμους.
- Χρησιμοποιούμε δένδρα για να υλοποιήσουμε ΑΤΔ υψηλότερου επιπέδου και θέλουμε να ρίξουμε το βάρος μας σε αυτούς.
- Χρησιμοποιούμε αλγορίθμους των οποίων η αποδοτικότητα εξαρτάται από την εκάστοτε αναπαράσταση — ένα γεγονός το οποίο μπορεί να χαθεί σε κάποιον ΑΤΔ.

Αυτοί οι λόγοι είναι οι ίδιοι για τους οποίους χρησιμοποιούμε γνωστές συγκεκριμένες αναπαραστάσεις για πίνακες και συνδεδεμένες λίστες. Η αναπαράσταση του δυαδικού δένδρου που φαίνεται στην Εικόνα 5.21 είναι ένα βασικό εργαλείο το οποίο και προσθέτουμε τώρα σε αυτή τη σύντομη λίστα.

Στην περίπτωση των συνδεδεμένων λιστών, ξεκινήσαμε εξετάζοντας στοιχειώδεις λειτουργίες για την εισαγωγή και τη διαγραφή κόμβων (δείτε τις Εικόνες 3.5 και 3.6). Για την καθιερωμένη αναπαράσταση δυαδικών δένδρων, αυτού του είδους οι λειτουργίες δεν είναι απαραίτητα στοιχειώδεις, λόγω του δεύτερου συνδέσμου. Αν θέλουμε να διαγράψουμε κάποιον κόμβο από ένα δυαδικό δένδρο, πρέπει να διευθετήσουμε το βασικό πρόβλημα ότι μπορεί να έχουμε δύο παιδιά αλλά μόνο ένα γονέα τον οποίο πρέπει να χειριστούμε αφού αφαιρεθεί ο κόμβος. Υπάρχουν τρεις φυσικές λειτουργίες που δεν έχουν αυτή τη δυσκολία: η εισαγωγή ενός νέου κόμβου στο τέλος (αντικατάσταση μηδενικού συνδέσμου με ένα σύνδεσμο προς το νέο κόμβο), η διαγραφή ενός φύλλου (αντικατάσταση του συνδέσμου προς αυτό με έναν κενό σύνδεσμο), και ο συνδυασμός δύο δένδρων με τη δημιουργία μιας νέας ρίζας με έναν αριστερό σύνδεσμο που δείχνει στο ένα δένδρο και ένα δεξιό σύνδεσμο που δείχνει στο άλλο δένδρο. Χρησιμοποιούμε ευρέως αυτές τις λειτουργίες όταν χειριζόμαστε δυαδικά δένδρα.

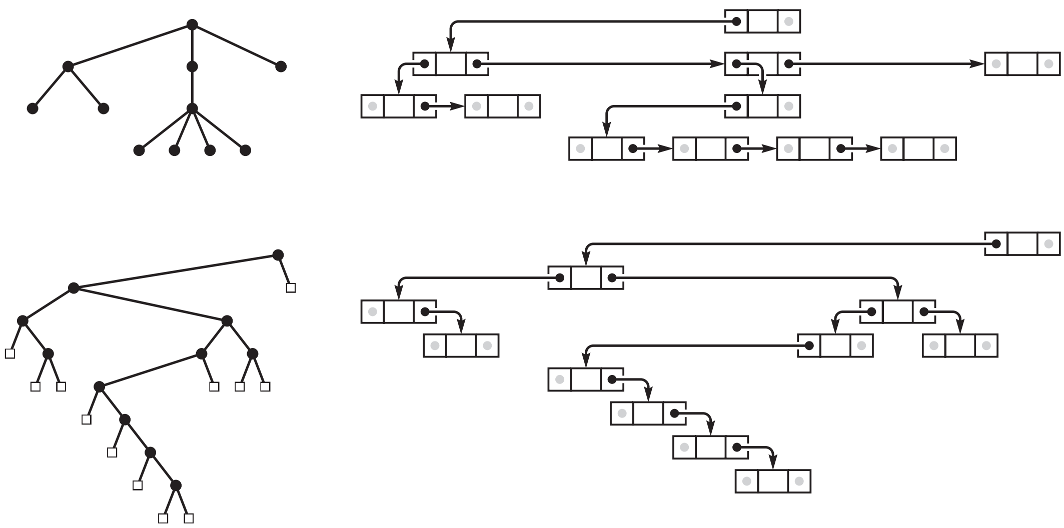
Ορισμός 5.2 *Ένα* **Μ-αδικό δένδρο** *(M-ary tree) είναι είτε ένας εξωτερικός κόμβος είτε ένας εσωτερικός κόμβος που συνδέεται με μια διατεταγμένη ακολουθία M δένδρων τα οποία είναι επίσης M-αδικά δένδρα.*

Συνήθως αναπαριστούμε τους κόμβους των *M*-αδικών δένδρων είτε ως δομές με *M* επώνυμους συνδέσμους (όπως στα δυαδικά δένδρα) είτε ως πίνακες *M* συνδέσμων. Για παράδειγμα, στο Κεφάλαιο 15 θα εξετάσουμε τα *τριαδικά* (3-ary ή ternary) δένδρα στα οποία χρησιμοποιούμε δομές με τρεις επώνυμους συνδέσμους (τον αριστερό, το μεσαίο, και το δεξιό), καθένας από τους οποίους έχει κάποιο συγκεκριμένο νόημα για τους σχετικούς αλγορίθμους. Σε διαφορετική περίπτωση, η χρήση πινάκων για την τήρηση των συνδέσμων είναι κατάλληλη επειδή η τιμή του *M* είναι καθορισμένη — αν και, όπως θα δούμε, όταν χρησιμοποιούμε μια τέτοια αναπαράσταση πρέπει να είμαστε ιδιαίτερα προσεκτικοί με την υπέρμετρη χρήση χώρου αποθήκευσης.

Ορισμός 5.3 *Ένα* **δένδρο** *— που ονομάζεται και* **διατεταγμένο δένδρο** *(ordered tree) — είναι ένας κόμβος (που ονομάζεται ρίζα) ο οποίος συνδέεται με μια ακολουθία ξένων μεταξύ τους δένδρων. Μια τέτοια ακολουθία ονομάζεται* **δάσος** *(forest).*

Η διάκριση μεταξύ διατεταγμένων και *M*-αδικών δένδρων είναι ότι οι κόμβοι των διατεταγμένων δένδρων είναι δυνατό να έχουν οποιοδήποτε πλήθος παιδιών, ενώ οι κόμβοι των *M*-αδικών δένδρων πρέπει να έχουν υποχρεωτικά *M* παιδιά. Μερικές φορές, όταν θέλουμε να διακρίνουμε τα διατεταγμένα από τα *M*-αδικά δένδρα χρησιμοποιούμε τον όρο *γενικό δένδρο* (general tree).

Επειδή κάθε κόμβος ενός διατεταγμένου δένδρου μπορεί να έχει οποιοδήποτε πλήθος συνδέσμων, είναι φυσικό να εξετάσουμε τη χρήση συνδεδεμένης λίστας αντί πίνακα για την αποθήκευση των συνδέσμων προς τα παιδιά ενός κόμβου. Στην Εικόνα 5.22 φαίνεται ένα παράδειγμα μιας τέτοιας αναπαράστασης. Από αυτό το παράδειγμα είναι φανερό ότι κάθε κόμβος περιέχει δύο συνδέσμους — ένα για τη συνδεδεμένη λίστα που τον συνδέει με τα αδέλφια του και έναν άλλο για τη συνδεδεμένη λίστα των παιδιών του.



Εικόνα 5.22
Αναπαράσταση δένδρου

Η αναπαράσταση ενός διατεταγμένου δένδρου με την τήρηση μιας συνδεδεμένης λίστας των παιδιών κάθε κόμβου είναι ισοδύναμη με την αναπαράσταση του δένδρου ως δυαδικού δένδρου. Στο επάνω δεξιά διάγραμμα φαίνεται μια αναπαράσταση με συνδεδεμένη λίστα των παιδιών του δένδρου που υπάρχει στα αριστερά του, με τη λίστα να υλοποιείται στους δεξιούς συνδέσμους των κόμβων και τον αριστερό σύνδεσμος κάθε κόμβου να δείχνει στον πρώτο κόμβο της συνδεδεμένης λίστας που περιέχει τα παιδιά του. Το κάτω δεξιά διάγραμμα αποτελεί μια ελαφρώς τροποποιημένη έκδοση του διαγράμματος που βρίσκεται από επάνω του, και αναπαριστά σαφώς το δυαδικό δένδρο που φαίνεται κάτω αριστερά. Με άλλα λόγια, μπορούμε να θεωρήσουμε ότι το δυαδικό δένδρο αναπαριστά το δένδρο.

Ιδιότητα 5.4 *Υπάρχει αμφιμονοσήμαντη αντιστοιχία (ένα προς ένα) μεταξύ των δυαδικών δένδρων και των διατεταγμένων δασών.*

Η αντιστοιχία φαίνεται στην Εικόνα 5.22. Μπορούμε να αναπαραστήσουμε οποιοδήποτε δάσος ως δυαδικό δένδρο, κάνοντας τον αριστερό σύνδεσμο κάθε κόμβου να δείχνει στο αριστερότερο παιδί του και το δεξιό σύνδεσμο κάθε κόμβου να δείχνει στον αδελφό κόμβο που βρίσκεται στα δεξιά. ■

Ορισμός 5.4 *Ένα* **δένδρο με ρίζα** *(rooted tree) ή* **μη διατεταγμένο δένδρο** *(unordered tree) είναι ένας κόμβος (που ονομάζεται ρίζα) ο οποίος συνδέεται με ένα πολυσύνολο δένδρων με ρίζα. (Ένα τέτοιο πολυσύνολο ονομάζεται μη διατεταγμένο δάσος.)*

Τα δένδρα που συναντήσαμε στο Κεφάλαιο 1, όταν εξετάζαμε το πρόβλημα της συνδετικότητας, ήταν μη διατεταγμένα δένδρα. Τέτοια δένδρα είναι δυνατό να οριστούν ως διατεταγμένα δένδρα στα οποία η σειρά με την οποία εξετάζονται τα παιδιά ενός κόμβου δεν είναι σημαντική. Θα μπορούσαμε επίσης να επιλέξουμε να ορίσουμε τα μη διατεταγμένα δένδρα ως αποτελούμενα από ένα σύνολο σχέσεων γονέα-παιδιού ανάμεσα στους κόμβους. Αυτή η επιλογή μοιάζει να έχει μικρή σχέση με τις αναδρομικές δομές που εξετάζουμε, αλλά ενδεχομένως είναι η συγκεκριμένη αναπαράσταση που είναι η πιο συνεπής με την αφηρημένη έννοια.

Θα μπορούσαμε να επιλέξουμε να αναπαραστήσουμε στον υπολογιστή ένα μη διατεταγμένο δένδρο χρησιμοποιώντας ένα διατεταγμένο δένδρο, αναγνωρίζοντας, όμως, ότι το ίδιο μη διατεταγμένο δένδρο θα μπορούσαν να το αναπαραστήσουν πολλά διαφορετικά διατεταγμένα δένδρα. Στην πραγματικότητα, το αντίστροφο πρόβλημα, του προσδιορισμού κατά πόσο δύο διαφορετικά διατεταγμένα δένδρα αναπαριστούν το ίδιο μη διατεταγμένο δένδρο — το πρόβλημα του *ισομορφισμού δένδρων* (tree-isomorphism) — είναι ένα δυσεπίλυτο πρόβλημα.

Ο γενικότερος τύπος δένδρου είναι εκείνος στον οποίο δεν ξεχωρίζουμε κάποιον κόμβο ως ρίζα. Για παράδειγμα, τα επικαλύπτοντα δένδρα (spanning trees) που προέκυπταν από τους αλγορίθμους συνδετικότητας του Κεφαλαίου 1 έχουν αυτή την ιδιότητα. Για να ορίσουμε κατάλληλα τα *δένδρα χωρίς ρίζα* (unrooted trees), τα *μη διατεταγμένα δένδρα* (unordered trees), ή τα *ελεύθερα δένδρα* (free trees), θα ξεκινήσουμε με έναν ορισμό για τους *γράφους*.

Ορισμός 5.5 **Γράφος** *(graph) είναι ένα σύνολο κόμβων μαζί με ένα σύνολο ακμών οι οποίες συνδέουν ζεύγη διακεκριμένων κόμβων (και υπάρχει το πολύ μία ακμή που συνδέει οποιοδήποτε ζεύγος κόμβων).*

Θα μπορούσαμε να φανταστούμε ότι ξεκινάμε από κάποιον κόμβο και ακολουθούμε μια ακμή προς το συστατικό κόμβο αυτής της ακμής, στη συνέχεια ακολουθούμε μια άλλη ακμή από αυτόν το δεύτερο κόμβο προς κάποιον άλλο κόμβο, και ούτω καθεξής. Μια ακολουθία ακμών που οδηγεί από έναν κόμβο σε κάποιον άλλο με αυτόν τον τρόπο, χωρίς κανένας κόμβος να εμφανίζεται δύο φορές, ονομάζεται *απλή διαδρομή* (simple path). Ένας γράφος είναι *συνδεδεμένος* (connected) αν υπάρχει πάντοτε μια απλή διαδρομή που να συνδέει οποιοδήποτε ζεύγος κόμβων του. Μια απλή διαδρομή της οποίας ο πρώτος και ο τελευταίος κόμβος ταυτίζονται ονομάζεται *κύκλος* (cycle).

Κάθε δένδρο είναι γράφος, αλλά ποιοι γράφοι είναι δένδρα; Θεωρούμε ότι ένας γράφος G είναι δένδρο αν ικανοποιεί οποιαδήποτε από τις επόμενες τέσσερις συνθήκες:

- Ο G έχει $N - 1$ ακμές και κανέναν κύκλο.
- Ο G έχει $N - 1$ ακμές και είναι συνδεδεμένος.
- Μία και μόνο μία απλή διαδρομή συνδέει κάθε ζεύγος κορυφών τού G.
- Ο G είναι συνδεδεμένος, αλλά παύει να είναι συνδεδεμένος αν αφαιρεθεί οποιαδήποτε ακμή.

Οποιαδήποτε από αυτές τις συνθήκες είναι ικανή και αναγκαία για να αποδειχθούν οι υπόλοιπες τρεις. Επισήμως, πρέπει να επιλέξουμε μία από αυτές ως ορισμό του *ελεύθερου δένδρου*· ανεπίσημα, όμως, αφήνουμε όλες τις συνθήκες να λειτουργούν ως ορισμός.

Αναπαριστούμε ένα ελεύθερο δένδρο απλώς ως μια συλλογή ακμών. Αν επιλέξουμε να αναπαραστήσουμε ένα ελεύθερο δένδρο ως μη διατεταγμένο δένδρο, ως διατεταγμένο δένδρο, ή ακόμη και ως δυαδικό δένδρο, πρέπει να αναγνωρίσουμε ότι, γενικά, υπάρχουν πολλοί διαφορετικοί τρόποι να αναπαραστήσουμε κάθε ελεύθερο δένδρο.

Η αφαίρεση του δένδρου εμφανίζεται συχνά, και οι διακρίσεις που εξετάζονται σε αυτή την ενότητα είναι σημαντικές επειδή η γνώση των διαφορετικών αφαιρέσεων δένδρου αποτελεί συχνά το απαραίτητο συστατικό για την εύρεση ενός αποδοτικού αλγορίθμου και της αντίστοιχης δομής δεδομένων για ένα δεδομένο πρόβλημα. Συχνά χειριζόμαστε άμεσα τις συγκεκριμένες αναπαραστάσεις δένδρων, χωρίς να αναφερόμαστε ειδικά σε κάποια αφαίρεση, αλλά συχνά επίσης επωφελούμαστε εργαζόμενοι πρώτα με την κατάλληλη αφαίρεση δένδρου και εξετάζοντας κατόπιν διάφορες συγκεκριμένες αναπαραστάσεις. Θα δούμε πολλά παραδείγματα αυτής της διαδικασίας σε ολόκληρο το βιβλίο.

Πριν επιστρέψουμε στους αλγορίθμους και τις υλοποιήσεις, θα μελετήσουμε ορισμένες βασικές μαθηματικές ιδιότητες των δένδρων οι οποίες θα μας φανούν χρήσιμες κατά τη σχεδίαση και την ανάλυση των αλγορίθμων δένδρων.

Ασκήσεις

▷ **5.56** Δώστε αναπαραστάσεις του ελεύθερου δένδρου της Εικόνας 5.20 ως δένδρου με ρίζα και ως δυαδικού δένδρου.

● **5.57** Πόσοι διαφορετικοί τρόποι υπάρχουν για να αναπαρασταθεί το ελεύθερο δένδρο της Εικόνας 5.20 ως διατεταγμένο δένδρο;

▷ **5.58** Σχεδιάστε τρία διατεταγμένα δένδρα που να είναι ισομορφικά με το διατεταγμένο δένδρο της Εικόνας 5.20. Δηλαδή, θα πρέπει να μπορείτε να μετατρέπετε τα τέσσερα δένδρα το ένα στο άλλο αντιμεταθέτοντας παιδιά.

○ **5.59** Θεωρήστε ότι τα δένδρα περιέχουν στοιχεία για τα οποία έχει υλοποιηθεί η μέθοδος ελέγχου ισότητας `equals()`. Γράψτε ένα αναδρομικό πρόγραμμα που να διαγράφει όλα τα φύλλα ενός δυαδικού δένδρου των οποίων τα στοιχεία είναι ίσα με κάποιο δεδομένο στοιχείο (δείτε το Πρόγραμμα 5.5).

○ **5.60** Τροποποιήστε τη μέθοδο "διαίρει και βασίλευε" για την εύρεση του μέγιστου στοιχείου ενός πίνακα (Πρόγραμμα 5.6) έτσι ώστε να διαιρεί τον πίνακα σε k τμήματα τα οποία να διαφέρουν σε μέγεθος το πολύ κατά 1, να βρίσκει αναδρομικά το μέγιστο σε κάθε τμήμα, και να επιστρέφει το μέγιστο όλων των τοπικών μέγιστων.

5.61 Σχεδιάστε τα τριαδικά και τετραδικά δένδρα που αντιστοιχούν σε $k = 3$ και $k = 4$ στην αναδρομική σχέση της Άσκησης 5.60, για έναν πίνακα 11 στοιχείων (δείτε την Εικόνα 5.6).

○ **5.62** Τα δυαδικά δένδρα είναι ισοδύναμα με τα δυαδικά αλφαριθμητικά που έχουν ένα παραπάνω bit με τιμή 0 από τα bit με τιμή 1, με τον πρόσθετο περιορισμό ότι, σε κάθε θέση k, το πλήθος των bit με τιμή 0 που εμφανίζονται αυστηρά στα αριστερά του k δεν είναι μεγαλύτερο από το πλήθος των bit με τιμή 1 που εμφανίζονται αυστηρά στα αριστερά του k. Ένα δυαδικό δένδρο είναι είτε ένα αλφαριθμητικό 0 είτε δύο τέτοια αλφαριθμητικά που συνενώνονται σε ένα, με ένα 1 να προηγείται. Σχεδιάστε το δυαδικό δένδρο που αντιστοιχεί στο ακόλουθο αλφαριθμητικό

```
1 1 1 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0
```

○ **5.63** Τα διατεταγμένα δένδρα είναι ισοδύναμα με ισοσταθμισμένα αλφαριθμητικά παρενθέσεων — ένα διατεταγμένο δένδρο είναι είτε μηδενικό (null) είτε μια ακολουθία διατεταγμένων δένδρων που περικλείεται σε παρενθέσεις. Σχεδιάστε το διατεταγμένο δένδρο που αντιστοιχεί στο επόμενο αλφαριθμητικό

```
( ( ( ) ( ( ) ( ) ) ( ) ) ( ( )( )( ) ) )
```

●● **5.64** Γράψτε ένα πρόγραμμα που να προσδιορίζει αν δύο πίνακες N ακεραίων μεταξύ 0 και $N-1$ αναπαριστούν ισομορφικά μη διατεταγμένα δένδρα, όταν ερμηνεύονται (όπως στο Κεφάλαιο 1) ως σύνδεσμοι γονέα-παιδιού σε ένα δένδρο με κόμβους αριθμημένους από 0 έως $N-1$. Δηλαδή, το πρόγραμμά σας πρέπει να προσδιορίζει κατά πόσο υπάρχει κάποιος τρόπος επαναρίθμησης των κόμβων ενός δένδρου έτσι ώστε η αναπαράσταση πίνακα του ενός δένδρου να είναι όμοια με την αναπαράσταση πίνακα του άλλου δένδρου.

●● **5.65** Γράψτε ένα πρόγραμμα που να προσδιορίζει κατά πόσο δύο δυαδικά δένδρα αναπαριστούν ισομορφικά μη διατεταγμένα δένδρα.

▷ **5.66** Σχεδιάστε όλα τα διατεταγμένα δένδρα με τα οποία θα μπορούσε να αναπαρασταθεί το δένδρο που ορίζεται από το σύνολο ακμών `0-1`, `1-2`, `1-3`, `1-4`, `4-5`.

● **5.67** Αποδείξτε ότι, αν ένας συνδεδεμένος γράφος με N κόμβους έχει την ιδιότητα πως, όταν αφαιρείται μια ακμή να αποσυνδέεται ο γράφος, τότε ο γράφος έχει $N-1$ ακμές και δεν περιέχει κύκλους.

5.5 Μαθηματικές ιδιότητες των δυαδικών δένδρων

Πριν αρχίσουμε να εξετάζουμε αλγορίθμους επεξεργασίας δένδρων, θα συνεχίσουμε σε μαθηματικό ύφος προκειμένου να μελετήσουμε μερικές βασικές ιδιότητες των δένδρων. Θα επικεντρώσουμε την εξέτασή μας στα δυαδικά δένδρα, επειδή τα χρησιμοποιούμε συχνά σε αυτό το βιβλίο. Η κατανόηση των βασικών ιδιοτήτων τους θα δημιουργήσει το υπόβαθρο για να κατανοήσουμε τα χαρακτηριστικά επιδόσεων πολλών αλγορίθμων τους οποίους θα συναντήσουμε — όχι μόνο εκείνους που χρησιμοποιούν δυαδικά δένδρα ως ρητές δομές δεδομένων, αλλά και αναδρομικούς αλγορίθμους "διαίρει και βασίλευε" καθώς και άλλες παρόμοιες εφαρμογές.

Ιδιότητα 5.5 *Ένα δυαδικό δένδρο με N εσωτερικούς κόμβους έχει $N + 1$ εξωτερικούς κόμβους.*

Αποδεικνύουμε αυτή την ιδιότητα με επαγωγή: ένα δυαδικό δένδρο χωρίς εσωτερικούς κόμβους έχει έναν εξωτερικό κόμβο. Συνεπώς, η ιδιότητα ισχύει για $N = 0$. Για $N > 0$, οποιοδήποτε δυαδικό δένδρο με N εσωτερικούς κόμβους έχει k εσωτερικούς κόμβους στο αριστερό του υποδένδρο και $N - 1 - k$ εσωτερικούς κόμβους στο δεξιό του υποδένδρο για κάποιο k μεταξύ 0 και $N - 1$, αφού η ρίζα είναι εσωτερικός κόμβος. Σύμφωνα με την επαγωγική υπόθεση, το αριστερό υποδένδρο έχει $k + 1$ εξωτερικούς κόμβους και το δεξιό υποδένδρο έχει $N - k$ εξωτερικούς κόμβους, οπότε συνολικά υπάρχουν $N + 1$ εξωτερικοί κόμβοι. ■

Ιδιότητα 5.6 *Ένα δυαδικό δένδρο με N εσωτερικούς κόμβους έχει $2N$ συνδέσμους: $N - 1$ συνδέσμους προς εσωτερικούς κόμβους και $N + 1$ συνδέσμους προς εξωτερικούς κόμβους.*

Σε κάθε δένδρο με ρίζα, κάθε κόμβος εκτός από τη ρίζα έχει ένα μοναδικό γονέα, και κάθε ακμή συνδέει έναν κόμβο με το γονέα του. Συνεπώς, υπάρχουν $N - 1$ σύνδεσμοι που συνδέουν εσωτερικούς κόμβους. Παρόμοια, καθένας από τους $N + 1$ εξωτερικούς κόμβους έχει ένα σύνδεσμο με το μοναδικό γονέα του. ■

Τα χαρακτηριστικά επιδόσεων πολλών αλγορίθμων εξαρτώνται, όχι μόνο από το πλήθος των κόμβων των συσχετιζόμενων δένδρων, αλλά και από διάφορες ιδιότητες των δομών.

Ορισμός 5.6 *Το* **επίπεδο** *(level) ενός κόμβου σε κάποιο δένδρο είναι κατά ένα μεγαλύτερο από το επίπεδο του γονέα του (με τη ρίζα να βρίσκεται στο επίπεδο 0). Το* **ύψος** *(height) ενός δένδρου είναι ίσο με το μέγιστο επίπεδο των κόμβων του και το* **μήκος διαδρομής** *(path length) ενός δένδρου είναι ίσο με το άθροισμα των επιπέδων όλων των κόμβων του. Το* **μήκος εσωτερικής διαδρομής** *(internal path length) ενός δυαδικού δένδρου είναι ίσο με το άθροισμα των επιπέδων όλων των εσωτερικών κόμβων του, ενώ το* **μήκος εξωτερικής διαδρομής** *(external path length) ενός δυαδικού δένδρου είναι ίσο με το άθροισμα των επιπέδων όλων των εξωτερικών κόμβων του.*

Ένας βολικός τρόπος για να υπολογίσουμε το μήκος διαδρομής ενός δένδρου είναι να αθροίσουμε, για όλα τα k, το γινόμενο του k επί το πλήθος των κόμβων που βρίσκονται στο επίπεδο k.

Αυτές οι ποσότητες έχουν και απλούς αναδρομικούς ορισμούς οι οποίοι προκύπτουν άμεσα από τους αναδρομικούς ορισμούς των δένδρων και των δυαδικών δένδρων. Για παράδειγμα, το ύψος ενός δένδρου είναι κατά 1 μεγαλύτερο από το μέγιστο ύψος των υποδένδρων της ρίζας του, ενώ το μήκος διαδρομής ενός δένδρου με N κόμβους είναι ίσο με το άθροισμα των μηκών διαδρομής των υποδένδρων της ρίζας του συν $N - 1$. Οι συγκεκριμένες ποσότητες σχετίζονται επίσης άμεσα με την ανάλυση των αναδρομικών αλγορίθμων. Για παράδειγμα, για πολλούς αναδρομικούς υπολογισμούς, το ύψος τού αντίστοιχου δένδρου είναι ακριβώς ίσο με το μέγιστο βάθος της αναδρομής ή με το μέγεθος της στοίβας που απαιτείται για την υποστήριξη αυτού του υπολογισμού.

Ιδιότητα 5.7 *Το μήκος της εξωτερικής διαδρομής κάθε δυαδικού δένδρου με N εσωτερικούς κόμβους είναι κατά $2N$ μεγαλύτερο από το μήκος της εσωτερικής του διαδρομής.*

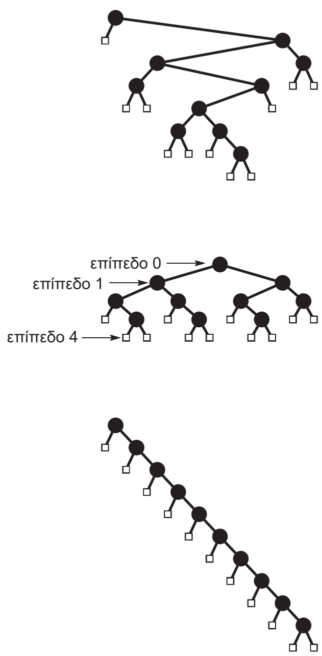
Θα μπορούσαμε να αποδείξουμε αυτή την ιδιότητα επαγωγικά, αλλά μια εναλλακτική απόδειξη (η οποία έχει εφαρμογή και στην Ιδιότητα 5.6) έχει διδακτικό χαρακτήρα. Κάθε δυαδικό δένδρο μπορεί να κατασκευαστεί σύμφωνα με την εξής διαδικασία: ξεκινάμε με το δυαδικό δένδρο που αποτελείται από έναν εξωτερικό κόμβο. Στη συνέχεια, επαναλαμβάνουμε την επόμενη διαδικασία N φορές: επιλέγουμε έναν εξωτερικό κόμβο και τον αντικαθιστούμε με ένα νέο εσωτερικό κόμβο ο οποίος έχει δύο εξωτερικούς κόμβους ως παιδιά. Αν ο εξωτερικός κόμβος που επιλέξαμε βρίσκεται στο επίπεδο k, το μήκος της εσωτερικής διαδρομής αυξάνεται κατά k, αλλά το μήκος της εξωτερικής διαδρομής αυξάνεται κατά $k + 2$ (αφαιρείται ένας εξωτερικός κόμβος στο επίπεδο k, αλλά προστίθενται δύο εξωτερικοί κόμβοι στο επίπεδο $k + 1$). Η διαδικασία ξεκινά με κάποιο δένδρο το οποίο έχει μήκος εσωτερικής και εξωτερικής διαδρομής ίσα με 0 και, σε καθένα από N βήματα, το μήκος της εξωτερικής του διαδρομής αυξάνεται κατά 2 μονάδες περισσότερο απ' ό,τι αυξάνεται το μήκος της εσωτερικής του διαδρομής. ■

Ιδιότητα 5.8 *Το ύψος ενός δυαδικού δένδρου με N εσωτερικούς κόμβους είναι τουλάχιστον ίσο με $\lg N$, και το πολύ ίσο με $N - 1$.*

Η χειρότερη περίπτωση είναι κάποιο εκφυλισμένο δένδρο με ένα μόνο φύλλο και $N - 1$ συνδέσμους από τη ρίζα προς το φύλλο (δείτε την Εικόνα 5.23). Η καλύτερη περίπτωση είναι κάποιο ισοσταθμισμένο δένδρο με 2^i εσωτερικούς κόμβους σε κάθε επίπεδο i εκτός από το κατώτατο (δείτε την Εικόνα 5.23). Αν το ύψος είναι h, τότε έχουμε

$$2^{h-1} < N + 1 \leq 2^h$$

αφού υπάρχουν $N + 1$ εξωτερικοί κόμβοι. Αυτή η ανισότητα συνεπάγεται την ιδιότητα που προσπαθούμε να αποδείξουμε: το ύψος του δένδρου στην καλύτερη περίπτωση είναι ακριβώς ίσο με $\lg N$ στρογγυλοποιημένο στον πλησιέστερο ακέραιο. ■



Εικόνα 5.23
Τρία δυαδικά δένδρα με 10 εσωτερικούς κόμβους

Το επάνω δυαδικό δένδρο έχει ύψος 7, μήκος εσωτερικής διαδρομής 31, και μήκος εξωτερικής διαδρομής 51. Ένα πλήρως ισοσταθμισμένο δυαδικό δένδρο (κέντρο) με 10 εσωτερικούς κόμβους έχει ύψος 4, μήκος εσωτερικής διαδρομής 19, και μήκος εξωτερικής διαδρομής 39 (κανένα δυαδικό δένδρο με 10 κόμβους δεν έχει μικρότερες τιμές για οποιαδήποτε από αυτές τις ποσότητες). Ένα εκφυλισμένο δυαδικό δένδρο (κάτω) με 10 εσωτερικούς κόμβους έχει ύψος 10, μήκος εσωτερικής διαδρομής 45, και μήκος εξωτερικής διαδρομής 65 (κανένα δυαδικό δένδρο με 10 κόμβους δεν έχει μεγαλύτερες τιμές για οποιαδήποτε από αυτές τις ποσότητες).

Ιδιότητα 5.9 *Το μήκος της εσωτερικής διαδρομής ενός δυαδικού δένδρου με N εσωτερικούς κόμβους είναι τουλάχιστον ίσο με $N \lg (N/4)$ και το πολύ ίσο με $N(N-1)/2$.*

Η χειρότερη περίπτωση και η καλύτερη περίπτωση προκύπτουν για τα ίδια δένδρα στα οποία αναφερθήκαμε στην εξέταση της Ιδιότητας 5.8 και παρουσιάζονται στην Εικόνα 5.23. Το μήκος της εσωτερικής διαδρομής για το δένδρο της δυσμενέστερης περίπτωσης είναι $0 + 1 + 2 + \ldots + (N-1) = N(N-1)/2$. Το δένδρο της καλύτερης περίπτωσης έχει $N+1$ εξωτερικούς κόμβους σε ύψος όχι μεγαλύτερο από $\lfloor \lg N \rfloor$. Πολλαπλασιάζοντας αυτές τις ποσότητες και εφαρμόζοντας την Ιδιότητα 5.7, παίρνουμε το φράγμα $(N+1)\lfloor \lg N \rfloor - 2N < N \lg (N/4)$. ∎

Όπως θα δούμε, τα δυαδικά δένδρα εμφανίζονται συχνά σε εφαρμογές της επιστήμης των υπολογιστών, και η απόδοση είναι βέλτιστη όταν είναι πλήρως ισοσταθμισμένα (ή σχεδόν ισοσταθμισμένα). Για παράδειγμα, τα δένδρα που χρησιμοποιούμε για να περιγράψουμε τους αλγορίθμους "διαίρει και βασίλευε", όπως στις περιπτώσεις της δυαδικής αναζήτησης ή της ταξινόμησης με συγχώνευση, είναι πλήρως ισοσταθμισμένα (δείτε την Άσκηση 5.74). Στα Κεφάλαια 9 και 13 θα εξετάσουμε ρητές δομές δεδομένων που βασίζονται σε ισοσταθμισμένα δένδρα.

Αυτές οι βασικές ιδιότητες των δένδρων παρέχουν τις πληροφορίες που χρειαζόμαστε για να αναπτύξουμε αποδοτικούς αλγορίθμους για πολλά πρακτικά προβλήματα. Για να κάνουμε λεπτομερέστερες αναλύσεις για πολλούς από τους συγκεκριμένους αλγορίθμους που θα συναντήσουμε απαιτείται μαθηματική ανάλυση υψηλότερου επιπέδου, αν και συχνά μπορούμε να πάρουμε εύκολα χρήσιμες εκτιμήσεις με επαγωγικά επιχειρήματα σαν αυτά που χρησιμοποιήσαμε στην τρέχουσα ενότητα. Στα επόμενα κεφάλαια θα εξετάσουμε επιπλέον μαθηματικές ιδιότητες των δένδρων, όποτε χρειάζεται. Σε αυτό το σημείο, είμαστε έτοιμοι να επιστρέψουμε σε αλγοριθμικά θέματα.

Ασκήσεις

▷ **5.68** Πόσοι εξωτερικοί κόμβοι υπάρχουν σε ένα *M*-αδικό δένδρο με *N* εσωτερικούς κόμβους; Χρησιμοποιήστε την απάντησή σας για να βρείτε το χώρο μνήμης που απαιτείται για την αναπαράσταση ενός τέτοιου δένδρου, υποθέτοντας ότι καθένας από τους συνδέσμους και τα στοιχεία απαιτεί μία λέξη μνήμης.

5.69 Δώστε το άνω και το κάτω φράγμα του ύψους ενός *M*-αδικού δένδρου με *N* εσωτερικούς κόμβους.

○ **5.70** Δώστε το άνω και το κάτω φράγμα του μήκους εσωτερικής διαδρομής ενός *M*-αδικού δένδρου με *N* εσωτερικούς κόμβους.

5.71 Δώστε το άνω και το κάτω φράγμα του πλήθους των φύλλων ενός δυαδικού δένδρου με *N* κόμβους.

● **5.72** Αποδείξτε ότι, αν τα επίπεδα των εξωτερικών κόμβων ενός δυαδικού δένδρου διαφέρουν κατά μια σταθερά, τότε το ύψος του δένδρου είναι $O(\log N)$.

○ **5.73** Ένα *δένδρο Fibonacci* ύψους $n > 2$ είναι ένα δυαδικό δένδρο, με ένα δένδρο Fibonacci ύψους $n - 1$ στο ένα υποδένδρο και ένα δένδρο Fibonacci ύψους $n - 2$ στο άλλο υποδένδρο. Ένα δένδρο Fibonacci ύψους 0 είναι ένας απλός εξωτερικός κόμβος, και ένα δένδρο Fibonacci ύψους 1 είναι ένας απλός εσωτερικός κόμβος με δύο εξωτερικά παιδιά (δείτε την Εικόνα 5.14). Δώστε το ύψος και το μήκος της εξωτερικής διαδρομής ενός δένδρου Fibonacci ύψους *n* ως συνάρτηση του *N*, που είναι το πλήθος των κόμβων του δένδρου.

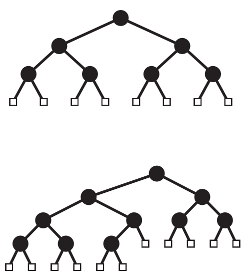
5.74 Ένα *δένδρο "διαίρει και βασίλευε" N* κόμβων είναι ένα δυαδικό δένδρο με μια ρίζα που έχει την ετικέτα *N*, ένα δένδρο "διαίρει και βασίλευε" με $\lfloor N/2 \rfloor$ κόμβους στο ένα υποδένδρο, και ένα δένδρο "διαίρει και βασίλευε" με $\lceil N/2 \rceil$ κόμβους στο άλλο υποδένδρο. (Μπορείτε να δείτε ένα δένδρο "διαίρει και βασίλευε" στην Εικόνα 5.6.) Σχεδιάστε ένα δένδρο "διαίρει και βασίλευε" με 11, 15, 16, και 23 κόμβους.

○ **5.75** Αποδείξτε με επαγωγή ότι το μήκος της εσωτερικής διαδρομής ενός δένδρου "διαίρει και βασίλευε" είναι μεταξύ $N \lg N$ και $N \lg N + N$.

5.76 Ένα *δένδρο "συνδύαζε και βασίλευε" N* κόμβων είναι ένα δυαδικό δένδρο με μια ρίζα που έχει την ετικέτα *N*, ένα δένδρο "συνδύαζε και βασίλευε" με $\lfloor N/2 \rfloor$ κόμβους στο ένα υποδένδρο, και ένα δένδρο "συνδύαζε και βασίλευε" με $\lceil N/2 \rceil$ κόμβους στο άλλο υποδένδρο (δείτε την Άσκηση 5.18). Σχεδιάστε ένα δένδρο "συνδύαζε και βασίλευε" με 11, 15, 16, και 23 κόμβους.

5.77 Αποδείξτε επαγωγικά ότι το μήκος της εσωτερικής διαδρομής ενός δένδρου "συνδύαζε και βασίλευε" είναι μεταξύ $N \lg N$ και $N \lg N + N$.

5.78 Ένα *πλήρες* (complete) δυαδικό δένδρο είναι ένα δένδρο με συμπληρωμένα όλα τα επίπεδά του εκτός ίσως από το τελευταίο, το οποίο συμπληρώνεται από τα αριστερά προς τα δεξιά, όπως φαίνεται στην Εικόνα 5.24. Αποδείξτε ότι το μήκος της εσωτερικής διαδρομής ενός πλήρους δένδρου με *N* κόμβους είναι μεταξύ $N \lg N$ και $N \lg N + N$.



Εικόνα 5.24
Πλήρη δυαδικά δένδρα με 7 και 10 εσωτερικούς κόμβους

Όταν το πλήθος των εξωτερικών κόμβων είναι μια δύναμη του 2 (επάνω), οι εξωτερικοί κόμβοι ενός πλήρους δυαδικού δένδρου βρίσκονται όλοι στο ίδιο επίπεδο. Σε διαφορετική περίπτωση (κάτω), οι εξωτερικοί κόμβοι εμφανίζονται σε δύο επίπεδα, με τους εσωτερικούς κόμβους που βρίσκονται στα αριστερά των εξωτερικών κόμβων να ανήκουν στο προτελευταίο επίπεδο (το προηγούμενο του κατώτατου).