

Πρόλογος

Το βιβλίο αυτό αποτελεί μια εκτενή εισαγωγή στη σύγχρονη μελέτη των υπολογιστικών αλγορίθμων. Πραγματεύεται ένα μεγάλο πλήθος αλγορίθμων, τους οποίους καλύπτει σε αρκετά μεγάλο βάθος, διατηρώντας ταυτόχρονα τη σχεδίαση και την ανάλυσή τους προσιτή σε όλους τους αναγνώστες. Έχουμε προσπαθήσει να διατηρήσουμε τη μελέτη των αλγορίθμων σε στοιχειώδες επίπεδο, χωρίς να θυσιάσουμε την εμβάθυνση ή τη μαθηματική αυστηρότητα.

Σε κάθε κεφάλαιο παρουσιάζεται ένας αλγόριθμος, μια τεχνική σχεδίασης, μια περιοχή εφαρμογών ή ένα σχετικό θέμα. Οι αλγόριθμοι περιγράφονται σε φυσική γλώσσα και μέσω ενός «ψευδοκώδικα» σχεδιασμένου ώστε να είναι κατανοητός από οποιονδήποτε έχει έστω και ελάχιστη εμπειρία στον προγραμματισμό. Το βιβλίο περιλαμβάνει περισσότερα από 230 σχήματα τα οποία αποσαφηνίζουν τη λειτουργία των αλγορίθμων. Δεδομένου ότι δίνουμε έμφαση στη *δραστηκότητα* ως κριτήριο σχεδίασης, έχουμε συμπεριλάβει διεξοδικές αναλύσεις των χρόνων εκτέλεσης των αλγορίθμων μας.

Σκοπός του βιβλίου είναι να χρησιμοποιηθεί ως διδακτικό εγχειρίδιο σε προπτυχιακά ή μεταπτυχιακά μαθήματα αλγορίθμων ή δομών δεδομένων. Δεδομένου ότι εκτός από τις μαθηματικές πλευρές της σχεδίασης αλγορίθμων πραγματεύεται και τα σχετικά τεχνικά ζητήματα, προσφέρεται εξίσου και για κατ' ιδίαν μελέτη από επαγγελματίες με τεχνικό προσανατολισμό.

Σε αυτή τη δεύτερη έκδοση, το βιβλίο έχει αναθεωρηθεί στο σύνολό του. Οι αλλαγές εκτείνονται από την προσθήκη νέων κεφαλαίων μέχρι την αναδιατύπωση μεμονωμένων προτάσεων.

Προς τον διδάσκοντα

Το βιβλίο αυτό έχει σχεδιαστεί έτσι ώστε να είναι αφ' ενός προσαρμόσιμο στις ανάγκες των αναγνωστών και αφ' ετέρου πλήρες. Μπορεί να χρησιμοποιηθεί στη διδασκαλία διαφόρων μαθημάτων, από ένα προπτυχιακό μάθημα δομών δεδομένων μέχρι και ένα μεταπτυχιακό μάθημα αλγορίθμων. Δεδομένου ότι η ύλη που περιλαμβάνει υπερβαίνει σημαντικά αυτήν που μπορεί να καλυφθεί σε ένα τυπικό εξαμηνιαίο μάθημα, μπορεί να θεωρηθεί ως ένα είδος «εκθετηρίου» από το οποίο έχετε τη δυνατότητα να επιλέξετε την ύλη που ταιριάζει περισσότερο στο μάθημα που επιθυμείτε να διδάξετε.

Ευελπιστούμε ότι η δομή του βιβλίου θα σας επιτρέψει να οργανώσετε το μάθημά σας μόνο με βάση τα κεφάλαια που θα θεωρήσετε αναγκαία. Έχουμε φροντίσει τα διάφορα κεφάλαια να είναι σχετικά αυτοτελή, ώστε να μη χρειάζεται να ανησυχείτε για αναπάντεχες και περιττές αλληλεξαρτήσεις μεταξύ τους. Σε κάθε κεφάλαιο,

παρατίθενται πρώτα τα πιο βατά τμήματα της ύλης και ακολουθούν τα πιο απαιτητικά, ενώ τα όρια μεταξύ των ενότητων σηματοδοτούν φυσιολογικά σημεία τερματισμού. Σε ένα προπτυχιακό μάθημα, θα μπορούσαν να χρησιμοποιηθούν μόνο οι αρχικές ενότητες των κεφαλαίων που θα επιλεγούν· σε ένα μεταπτυχιακό μάθημα, θα μπορούσε κανείς να καλύψει στο σύνολό τους τα κεφάλαια που θα επιλέξει.

Το βιβλίο περιλαμβάνει περισσότερες από 920 ασκήσεις και περισσότερα από 140 προβλήματα. Κάθε ενότητα ολοκληρώνεται με τις αντίστοιχες ασκήσεις, και κάθε κεφάλαιο με τα αντίστοιχα προβλήματα. Οι ασκήσεις συνίστανται εν γένει σε σύντομα ερωτήματα που ελέγχουν την αφομοίωση της ύλης σε βασικό επίπεδο. Ορισμένες από τις ασκήσεις αποτελούν απλές νοητικές δοκιμασίες αυτοελέγχου, ενώ άλλες είναι περισσότερο ουσιώδεις και προσφέρονται για ανάθεση κατ' οίκον εργασιών. Τα προβλήματα συνιστούν πιο σύνθετες μελέτες περιπτώσεων στις οποίες συχνά παρουσιάζονται και κάποια συμπληρώματα της θεωρίας, και αποτελούνται κατά κανόνα από διάφορα ερωτήματα τα οποία καθοδηγούν τον σπουδαστή στα βήματα που θα πρέπει να ακολουθήσει για να φτάσει στη λύση.

Οι ενότητες και οι ασκήσεις που είναι περισσότερο κατάλληλες για μεταπτυχιακούς παρά για προπτυχιακούς φοιτητές επισημαίνονται με έναν αστερίσκο (*). Οι επισημασμένες ενότητες δεν είναι κατ' ανάγκη δυσκολότερες από τις μη επισημασμένες, αλλά ενδέχεται να απαιτούν υψηλότερο μαθηματικό υπόβαθρο. Ομοίως, οι επισημασμένες ασκήσεις ενδέχεται να απαιτούν υψηλότερο υπόβαθρο ή αυξημένη επινοητικότητα.

Προς τον σπουδαστή

Ελπίζουμε το βιβλίο αυτό να αποτελέσει για εσάς μια ευχάριστη εισαγωγή στο πεδίο των αλγορίθμων. Έχουμε προσπαθήσει να παρουσιάσουμε όλους τους αλγορίθμους με τρόπο βατό και ενδιαφέροντα. Για να διευκολύνουμε την κατανόηση μη οικείων ή δύσκολων αλγορίθμων, έχουμε ακολουθήσει μια βήμα προς βήμα περιγραφή. Παραθέτουμε επίσης λεπτομερείς αναφορές στα μαθηματικά που απαιτούνται για την κατανόηση της ανάλυσης των αλγορίθμων. Εάν είστε ήδη εξοικειμένοι με κάποιο γνωστικό αντικείμενο, θα διαπιστώσετε ότι τα κεφάλαια είναι δομημένα έτσι ώστε να μπορείτε να διατρέξετε επί τροχάδην τις εισαγωγικές ενότητες και να προχωρήσετε γρήγορα στα πιο απαιτητικά τμήματα της ύλης.

Δεδομένου του μεγάλου όγκου του βιβλίου, το μάθημά σας θα καλύψει πιθανότητα μόνο ένα τμήμα της συνολικής ύλης του. Ωστόσο, έχουμε καταβάλει κάθε προσπάθεια ώστε το βιβλίο αυτό, πέραν της χρήσης του ως διδακτικού εγχειριδίου, να αποτελέσει επίσης χρήσιμο βοήθημα στη μελλοντική σας καριέρα τόσο ως βιβλίο αναφοράς για μαθηματικά ζητήματα όσο και ως τεχνικό εγχειρίδιο.

Ποια είναι τα προαπαιτούμενα για τη μελέτη αυτού του βιβλίου;

- Θα πρέπει να έχετε κάποια εμπειρία στον προγραμματισμό. Συγκεκριμένα, θα πρέπει να έχετε κατανοήσει τις αναδρομικές διαδικασίες και τις απλές δομές δεδομένων, όπως οι συστοιχίες και οι αλυσίδες.
- Θα πρέπει να έχετε κάποια ευχέρεια στις αποδείξεις μέσω μαθηματικής επαγωγής. Λίγα τμήματα του βιβλίου προϋποθέτουν στοιχειώδεις γνώσεις απειροστικού λογισμού. Πέραν αυτών, όλες οι μαθηματικές τεχνικές που θα χρειαστείτε αναπτύσσονται στα Μέρη I και VII.

Προς τον επαγγελματία

Το μεγάλο εύρος των ζητημάτων που καλύπτει το βιβλίο αυτό το καθιστούν ιδανικό εγχειρίδιο αναφοράς για αλγορίθμους. Το γεγονός ότι όλα τα κεφάλαια είναι σχετικά αυτοτελή σας επιτρέπει να εστιάσετε στα ζητήματα που σας ενδιαφέρουν περισσότερο.

Οι περισσότεροι από τους αλγορίθμους που πραγματευόμαστε στο βιβλίο αυτό έχουν μεγάλη πρακτική χρησιμότητα. Για τον λόγο αυτό, η μελέτη τους επεκτείνεται και σε ζητήματα υλοποίησης, καθώς και σε τεχνικά θέματα. Για τους λίγους αλγορίθμους που έχουν πρωτίστως θεωρητικό ενδιαφέρον, συχνά παραθέτουμε πρακτικές εναλλακτικές λύσεις.

Εάν θελήσετε να υλοποιήσετε κάποιους από τους αλγορίθμους, θα διαπιστώσετε ότι η μετάφρασή τους από τον ψευδοκώδικα που χρησιμοποιούμε στην γλώσσα προγραμματισμού της επιλογής σας είναι αρκετά απλή διαδικασία. Ο ψευδοκώδικας είναι σχεδιασμένος έτσι ώστε η παρουσίαση όλων των αλγορίθμων να χαρακτηρίζεται από σαφήνεια και λιτότητα. Ως εκ τούτου, δεν ασχολούμαστε με ζητήματα διαχείρισης σφαλμάτων και τεχνολογίας λογισμικού τα οποία βασίζονται σε συγκεκριμένες παραδοχές για το εκάστοτε προγραμματιστικό περιβάλλον. Στόχος μας είναι να παρουσιάσουμε κάθε αλγόριθμο με απλό και άμεσο τρόπο, και να αποφύγουμε τη συσκοτίση των βασικών του χαρακτηριστικών από τις ιδιομορφίες της κάθε γλώσσας προγραμματισμού.

Προς τους συναδέλφους μας

Έχουμε φροντίσει να παραθέσουμε μια εκτενή βιβλιογραφία καθώς και παραπομπές στη σύγχρονη αρθρογραφία. Στο τέλος κάθε κεφαλαίου, παρατίθενται κάποιες «σημειώσεις» οι οποίες αναφέρονται σε ζητήματα ιστορικού ενδιαφέροντος καθώς και σε βιβλιογραφικές πηγές. Εντούτοις, οι σημειώσεις αυτές δεν εξαντλούν τις βιβλιογραφικές αναφορές στο πεδίο των αλγορίθμων. Αν και, κρίνοντας από τον όγκο του βιβλίου, θα ήταν μάλλον δύσκολο να το πιστέψει κανείς, πολλοί ενδιαφέροντες αλγόριθμοι δεν στάθηκε δυνατόν να συμπεριληφθούν λόγω έλλειψης χώρου.

Παρά τα αναρίθμητα αιτήματα που δεχθήκαμε από σπουδαστές για να συμπεριλάβουμε τις λύσεις των προβλημάτων και των ασκήσεων, επιλέξαμε να μην παραθέσουμε αναλυτικές υποδείξεις για τα προβλήματα και τις ασκήσεις, προκειμένου οι σπουδαστές να αποφύγουν τον πειρασμό να ανατρέξουν απλώς στη λύση αντί να τη βρουν οι ίδιοι.

Αλλαγές στη δεύτερη έκδοση

Τι έχει αλλάξει μεταξύ της πρώτης και της δεύτερης έκδοσης αυτού του βιβλίου; Ανάλογα με το πώς το εξετάζει κανείς, είτε ελάχιστα πράγματα είτε πάρα πολλά.

Όπως μπορεί να διαπιστώσει κανείς με μια σύντομη ματιά στον πίνακα περιεχομένων, τα κεφάλαια και οι ενότητες της πρώτης έκδοσης υπάρχουν στην πλειονότητά τους και στη δεύτερη. Έχουμε αφαιρέσει δύο κεφάλαια και λίγες ενότητες, και έχουμε προσθέσει τρία νέα κεφάλαια και τέσσερις νέες ενότητες, πέραν των νέων αυτών κεφαλαίων. Εάν προσπαθήσει κανείς να εκτιμήσει το εύρος των αλλαγών με βάση τον πίνακα των περιεχομένων, θα συμπεράνει μάλλον ότι δεν υπάρχουν σημαντικές μεταβολές.

Εντούτοις, οι αλλαγές στη δεύτερη έκδοση εκτείνονται πολύ πέραν αυτών που φαίνονται στον πίνακα των περιεχομένων. Οι πιο σημαντικές από αυτές, με αυθαίρετη σειρά, είναι οι εξής:

- Στην ομάδα των συγγραφέων έχει προστεθεί και ο Cliff Stein.
- Έχουν διορθωθεί κάποια λάθη. Πόσα; Ας πούμε απλώς αρκετά.
- Υπάρχουν τρία νέα κεφάλαια:
 - Το Κεφάλαιο 1 πραγματεύεται τον ρόλο των αλγορίθμων στις υπολογιστικές διαδικασίες.
 - Το Κεφάλαιο 5 καλύπτει την πιθανοτική ανάλυση και τους τυχαιοκρατικούς αλγορίθμους. Όπως και στην πρώτη έκδοση, τα αντικείμενα αυτά απαντούν σε διάφορα σημεία του βιβλίου.
 - Το Κεφάλαιο 29 είναι αφιερωμένο στον γραμμικό προγραμματισμό.
- Σε κεφάλαια που υπήρχαν και στην πρώτη έκδοση, έχουν προστεθεί νέες ενότητες στα εξής αντικείμενα:
 - στην πλήρη διασπορά (Ενότητα 11.5),
 - σε δύο εφαρμογές δυναμικού προγραμματισμού (Ενότητες 15.1 και 15.5), και
 - σε προσεγγιστικούς αλγορίθμους οι οποίοι βασίζονται σε τυχαιοκρατικές διαδικασίες και σε γραμμικό προγραμματισμό (Ενότητα 35.4).
- Προκειμένου να εισαγάγουμε τον αναγνώστη στην καθαυτό μελέτη των αλγορίθμων νωρίτερα, έχουμε μετακινήσει τρία από τα κεφάλαια που αφορούν το μαθηματικό υπόβαθρο από το Μέρος I στο Παράρτημα, το οποίο αποτελεί το Μέρος VII.
- Υπάρχουν περισσότερα από 40 νέα προβλήματα και περισσότερες από 185 νέες ασκήσεις.
- Έχουμε καθιερώσει την απευθείας χρήση των αναλλοίωτων συνθηκών για την απόδειξη της ορθότητας αλγορίθμων. Η πρώτη αναλλοίωτη συνθήκη εμφανίζεται στο Κεφάλαιο 2, ενώ η μέθοδος χρησιμοποιείται συνολικά μερικές δεκάδες φορές σε όλο το βιβλίο.
- Πολλές από τις πιθανοτικές αναλύσεις έχουν αναθεωρηθεί. Συγκεκριμένα, χρησιμοποιούμε σε περίπου δέκα περιπτώσεις την τεχνική των «δεικτριών τυχαίων μεταβλητών», η οποία απλοποιεί την πιθανοτική ανάλυση, ιδιαίτερα όταν οι δείκτριες μεταβλητές είναι εξαρτημένες.
- Έχουμε επεκτείνει και ανανεώσει τις σημειώσεις των κεφαλαίων και τη βιβλιογραφία. Η βιβλιογραφία έχει αυξηθεί κατά περίπου 50%, και έχουμε προσθέσει αναφορές σε πολλά νέα αποτελέσματα τα οποία δημοσιεύθηκαν μετά την κυκλοφορία της πρώτης έκδοσης.

Έχουμε προχωρήσει επίσης στις ακόλουθες αλλαγές:

- Από το κεφάλαιο που αφορά την επίλυση αναδρομικών σχέσεων έχει απαλειφθεί η επαναληπτική μέθοδος. Αντ' αυτής, στην Ενότητα 4.2, έχουμε «προαγάγει» τα δένδρα αναδρομής σε καθαυτό μέθοδο επίλυσης. Έχουμε διαπιστώσει ότι η μέθοδος της σχεδίασης δένδρων αναδρομής είναι λιγότερο ευάλωτη σε

σφάλματα απ' ό,τι η επαναληπτική επίλυση αναδρομικών σχέσεων. Επισημαίνουμε, ωστόσο, ότι τα δένδρα αναδρομής είναι καλύτερο να χρησιμοποιούνται για τη διατύπωση εικασιών οι οποίες στη συνέχεια επιβεβαιώνονται μέσω της μεθόδου της αντικατάστασης.

- Η μέθοδος διαμέρισης που χρησιμοποιείται στην ταχυταξινόμηση (Ενότητα 7.1) και ο αλγόριθμος διατακτικών στατιστικών γραμμικού αναμενόμενου χρόνου (Ενότητα 9.2) διαφέρουν σε σχέση με την πρώτη έκδοση. Στην έκδοση αυτή, χρησιμοποιούμε τη μέθοδο που έχει αναπτυχθεί από τον Lomuto, η οποία, σε συνδυασμό με τις δείκτριες τυχαίες μεταβλητές, απλοποιεί κάπως την ανάλυση. Η μέθοδος της πρώτης έκδοσης, η οποία οφείλεται στον Hoare, παρατίθεται ως πρόβλημα στο Κεφάλαιο 7.
- Έχουμε τροποποιήσει την περιγραφή της καθολικής διασποράς (Ενότητα 11.3.3) έτσι ώστε να εντάσσεται στη μελέτη της πλήρους διασποράς.
- Η ανάλυση του ύψους ενός τυχαία κατασκευασμένου δυαδικού δένδρου αναζήτησης (Ενότητα 12.4) έχει απλοποιηθεί σημαντικά.
- Η μελέτη των στοιχείων δυναμικού προγραμματισμού (Ενότητα 15.3) έχει επεκταθεί σημαντικά, όπως και η μελέτη των στοιχείων των άπληστων αλγορίθμων (Ενότητα 16.2). Η διερεύνηση του προβλήματος της επιλογής δραστηριοτήτων, με το οποίο ξεκινά το κεφάλαιο των άπληστων αλγορίθμων, διευκολύνει την αποσαφήνιση της σχέσης μεταξύ δυναμικού προγραμματισμού και άπληστων αλγορίθμων.
- Η απόδειξη του χρόνου εκτέλεσης της δομής δεδομένων της ένωσης ξένων συνόλων στην Ενότητα 21.4 έχει αντικατασταθεί. Η νέα απόδειξη βασίζεται στην ενεργειακή μέθοδο για τον προσδιορισμό ενός αυστηρού φράγματος.
- Η απόδειξη της ορθότητας του αλγορίθμου για τις ισχυρά συνδεδεμένες συνιστώσες στην Ενότητα 22.5 έχει γίνει απλούστερη, σαφέστερη και πιο άμεση.
- Το Κεφάλαιο 24, που αναφέρεται στις ομοαφεταιριακές ελαφρύτατες διαδρομές, έχει ανασυγκροτηθεί έτσι ώστε οι αποδείξεις των βασικών ιδιοτήτων να ενταχθούν σε μια ξεχωριστή ενότητα. Η νέα δομή του κεφαλαίου μας επιτρέπει να εισαγάγουμε νωρίτερα τους σχετικούς αλγορίθμους.
- Η Ενότητα 34.5 περιλαμβάνει μια διευρυμένη επισκόπηση της NP-πληρότητας καθώς και νέες αποδείξεις της NP-πληρότητας για τα προβλήματα του χαμιλτονιανού κύκλου και του αθροίσματος υποσυνόλων.

Τέλος, σχεδόν όλες οι ενότητες έχουν τροποποιηθεί προκειμένου να διορθωθούν, να απλοποιηθούν και να αποσαφηνιστούν οι διάφορες εξηγήσεις και αποδείξεις.

Δικτυότοπος

Μια άλλη αλλαγή σε σχέση με την πρώτη έκδοση είναι ότι το βιβλίο διαθέτει πλέον τον δικό του δικτυότοπο: <http://mitpress.mit.edu/algorithms/>. Μέσω αυτού, μπορείτε να αναφέρετε σφάλματα, να προμηθευτείτε έναν κατάλογο των γνωστών σφαλμάτων, ή να κάνετε υποδείξεις· τα σχόλιά σας είναι ευπρόσδεκτα. Θα εκτιμούσαμε ιδιαίτερα τις ιδέες σας για νέες ασκήσεις και προβλήματα, που θα θέλαμε όμως να συνοδεύονται από τις λύσεις τους.

Δυστυχώς, δεν είναι δυνατόν να απαντήσουμε προσωπικά σε όλα τα σχόλια.

Ευχαριστίες για την πρώτη έκδοση

Πολλοί φίλοι και συνάδελφοι έχουν συμβάλει σημαντικά στην ποιότητα αυτού του βιβλίου. Τους ευχαριστούμε όλους για τη συνδρομή τους και την εποικοδομητική τους κριτική.

Το Εργαστήριο Επιστήμης Υπολογιστών του MIT μας εξασφάλισε ένα ιδανικό περιβάλλον εργασίας. Οι συνάδελφοί μας από την Ομάδα Θεωρίας Υπολογισμού του εργαστηρίου προσέφεραν σημαντική υποστήριξη και επέδειξαν εξαιρετική υπομονή στις συνεχείς παρακλήσεις μας για κριτική αξιολόγηση των διαφόρων κεφαλαίων. Θα θέλαμε να ευχαριστήσουμε ιδιαίτερα τους Baruch Awerbuch, Shafi Goldwasser, Λεωνίδα Γκίμπα, Tom Leighton, Albert Meyer, David Shmoys, και Ένα Tardos. Ευχαριστούμε επίσης τους William Ang, Sally Bemus, Ray Hirschfeld, και Mark Reinhold για την εύρυθμη λειτουργία των υπολογιστών μας (DEC Microvax, Apple Macintosh, και Sun Sparcstation) και την επαναμεταγλώττιση του TEX κάθε φορά που υπερβαίναμε κάποιο χρονικό όριο μεταγλώττισης. Η Thinking Machines Corporation προσέφερε την υποστήριξη της στον Charles Leiserson για να ασχοληθεί με αυτό το βιβλίο κατά τη διάρκεια μιας εκπαιδευτικής άδειας από το MIT.

Πολλοί συνάδελφοι οι οποίοι χρησιμοποίησαν προσχέδια του κειμένου αυτού στη διδασκαλία μαθημάτων σε άλλες σχολές μας υπέδειξαν αρκετές διορθώσεις και αναθεωρήσεις. Θα θέλαμε να ευχαριστήσουμε ιδιαίτερα τους Richard Beigel, Andrew Goldberg, Joan Lucas, Mark Overmars, Alan Sherman, και Diane Souvaine.

Πολλοί βοηθοί καθηγητού στα μαθήματά μας συνεισέφεραν σημαντικά στην ανάπτυξη του υλικού αυτού. Ευχαριστούμε ιδιαίτερα τους Alan Baratz, Bonnie Berger, Aditi Dhagat, Burt Kaliski, Arthur Lent, Andrew Moulton, Μάριο Παπαευθυμίου, Cindy Phillips, Mark Reinhold, Phil Rogaway, Flavio Rose, Arie Rudich, Alan Sherman, Cliff Stein, Susmita Sur, Gregory Troxel, και Margaret Tuttle.

Πολλοί άλλοι προσέφεραν πολύτιμη τεχνική υποστήριξη. Η Denise Sergent αφιέρωσε πολλές ώρες στις βιβλιοθήκες του MIT αναζητώντας βιβλιογραφικές πηγές. Η Maria Sensale, η βιβλιοθηκονόμος του αναγνωστηρίου μας, ήταν πάντοτε προσηνής και εξυπηρετική. Η πρόσβαση που είχαμε στην προσωπική βιβλιοθήκη του Albert Meyer μας απάλλαξε από πολλές ώρες απασχόλησης στη βιβλιοθήκη για την προετοιμασία των σημειώσεων των κεφαλαίων. Οι Shlomo Kipnis, Bill Niehaus, και David Wilson διόρθωσαν παλιές ασκήσεις, ανέπτυξαν νέες, και προσέθεσαν σημειώσεις στις λύσεις τους. Οι Μάριος Παπαευθυμίου και Gregory Troxel βοήθησαν στην ευρετηρίαση των όρων. Επί πολλά χρόνια, οι γραμματείς μας Inna Radzihovskiy, Denise Sergent, Gayle Sherman, και ιδιαιτέρως η Be Blackburn παρείχαν απεριόριστη υποστήριξη στην προσπάθεια αυτή. Τις ευχαριστούμε.

Πολλοί φοιτητές μας ανέφεραν σφάλματα στα αρχικά προσχέδια του βιβλίου. Ευχαριστούμε ιδιαίτερα τους Bobby Blumofe, Bonnie Eisenberg, Raymond Johnson, John Keen, Richard Lethin, Mark Lillibridge, Ιωάννη Πεζάρη, Steve Ponzio, και Margaret Tuttle για την προσεκτική τους ανάγνωση.

Επίσης, πολλοί συνάδελφοι συνέταξαν κριτικές ανασκοπήσεις για συγκεκριμένα κεφάλαια, ή προσέφεραν πληροφορίες για συγκεκριμένους αλγόριθμους, και τους ευχαριστούμε θερμά. Ιδιαίτερα θα θέλαμε να ευχαριστήσουμε τους Bill Aiello, Alok Aggarwal, Eric Bach, Vášek Chvátal, Richard Cole, Johan Hastad, Alex Ishii, David Johnson, Joe Kilian, Dina Kravets, Bruce Maggs, Jim Orlin, James Park, Thane Plambeck, Hershel Safer, Jeff Shallit, Cliff Stein, Gil Strang, Bob Tarjan, και Paul Wang. Διάφοροι συνάδελφοί μας είχαν επίσης την καλοσύνη να μας προμηθεύ-

σουν προβλήματα· ευχαριστούμε ιδιαίτερα τους Andrew Goldberg, Danny Sleator, και Umesh Vazirani.

Ήταν χαρά για μας η συνεργασία με την MIT Press και την McGraw-Hill για τη διαμόρφωση του κειμένου αυτού. Ευχαριστούμε ιδιαίτερος τους Frank Satlow, Terry Ehling, Larry Cohen, και Lorrie Lejeune της MIT Press και τον David Shapiro της McGraw-Hill για την ενθάρρυνση, την υποστήριξη και την υπομονή τους, καθώς και τον Larry Cohen για την εξαιρετική επιμέλεια της έκδοσης.

Ευχαριστίες για τη δεύτερη έκδοση

Όταν ζητήσαμε από την Julie Sussman, P.P.A., να αναλάβει την τεχνική επιμέλεια της δεύτερης έκδοσης, δεν μπορούσαμε να φανταστούμε πόσο εύστοχη ήταν η επιλογή μας. Εκτός από την επιμέλεια του κειμένου ως προς το τεχνικό σκέλος, η Julie ανέλαβε επίσης με ιδιαίτερο ζήλο και τη φιλολογική επιμέλεια. Αισθανόμαστε πολύ πιο ταπεινοί όταν σκεπτόμαστε πόσα λάθη εντόπισε η Julie στα αρχικά μας δοκίμια, παρ' ότι αυτό δεν προκαλεί καμία έκπληξη αν αναλογιστούμε πόσα εντόπισε στην πρώτη έκδοση (δυστυχώς, αφότου είχε εκτυπωθεί). Επιπλέον, η Julie υπερέβη το δικό της χρονοδιάγραμμα προκειμένου να προσαρμοστεί στις δικές μας υποχρεώσεις –φτάνοντας στο σημείο να επιμεληθεί κάποια κεφάλαια ακόμη και σε ένα ταξίδι της στις Παρθένους Νήσους! Julie, όσο και αν σε ευχαριστήσουμε για την καταπληκτική σου δουλειά, δεν θα είναι αρκετό.

Η προετοιμασία της δεύτερης έκδοσης πραγματοποιήθηκε ενόσω οι συγγραφείς ήταν μέλη του Τμήματος Επιστήμης Υπολογιστών του Dartmouth College και του Εργαστηρίου Επιστήμης Υπολογιστών του MIT. Και στα δύο αυτά ιδρύματα, το περιβάλλον εργασίας ήταν ιδιαίτερα ενθαρρυντικό, και θα θέλαμε να ευχαριστήσουμε όλους τους συναδέλφους μας για την υποστήριξή τους.

Διάφοροι φίλοι και συνάδελφοι ανά την υφήλιο διατύπωσαν υποδείξεις και απόψεις οι οποίες μας καθοδήγησαν στη συγγραφή του κειμένου. Ευχαριστούμε πολύ τους Sanjeev Arora, Javed Aslam, Guy Blelloch, Avrim Blum, Scot Drysdale, Hany Farid, Hal Gabow, Andrew Goldberg, David Johnson, Yanlin Liu, Nicolas Schabanel, Alexander Schrijver, Sasha Shen, David Shmoys, Dan Spielman, Gerald Jay Sussman, Bob Tarjan, Mikkel Thorup, και Vijay Vazirani.

Πολλοί δάσκαλοι και συνάδελφοι μας δίδαξαν πολλά σχετικά με τους αλγορίθμους. Θα θέλαμε να ευχαριστήσουμε ιδιαίτερα τους δασκάλους μας Jon L. Bentley, Bob Floyd, Don Knuth, Harold Kuhn, H. T. Kung, Richard Lipton, Arnold Ross, Larry Snyder, Michael I. Shamos, David Shmoys, Ken Steiglitz, Tom Szymanski, Éva Tardos, Bob Tarjan, και Jeffrey Ullman.

Θα πρέπει επίσης να ευχαριστήσουμε πολλούς από τους βοηθούς καθηγητού στα μαθήματα της θεωρίας αλγορίθμων στο MIT και το Dartmouth, μεταξύ των οποίων τους Joseph Adler, Craig Barrack, Bobby Blumofe, Roberto De Prisco, Matteo Frigo, Igal Galperin, David Gupta, Raj D. Iyer, Nabil Kahale, Sarfraz Khurshid, Σταύρο Κολλιόπουλο, Alain Leblanc, Yuan Ma, Maria Minkoff, Δημήτρη Μήτσουρα, Alin Popescu, Harald Prokop, Sudipta Sengupta, Donna Slonim, Joshua A. Tauber, Sivan Toledo, Elisheva Werner-Reiss, Lea Wittie, Qiang Wu, και Michael Zhang.

Οι William Ang, Scott Blomquist, και Greg Shomo στο MIT και οι Wayne Cripps, John Konkle, και Tim Tregubov στο Dartmouth μας προσέφεραν υπολογιστική υποστήριξη. Ευχαριστούμε επίσης τους Be Blackburn, Don Dailey, Leigh Deacon,

Irene Sebeda, και Cheryl Patton Wu του MIT και τους Phyllis Bellmore, Kelly Clark, Delia Mauceli, Sammie Travis, Deb Whiting, και Beth Young του Dartmouth για τη διοικητική υποστήριξη. Οι Michael Fromberger, Brian Campbell, Amanda Eubanks, Sung Hoon Kim, και Neha Narula μας προσέφεραν επίσης σημαντική βοήθεια στο Dartmouth.

Πολλοί αναγνώστες είχαν την καλοσύνη να μας αναφέρουν σφάλματα στην πρώτη έκδοση. Θα θέλαμε να ευχαριστήσουμε τους παρακάτω αναγνώστες, καθένας από τους οποίους ανέφερε πρώτος κάποιο σφάλμα στην πρώτη έκδοση: Len Adleman, Selim Akl, Richard Anderson, Juan Andrade-Cetto, Gregory Bachelis, David Barrington, Paul Beame, Richard Beigel, Margrit Betke, Alex Blakemore, Bobby Blumofe, Alexander Brown, Xavier Cazin, Jack Chan, Richard Chang, Chienhua Chen, Ien Cheng, Hoon Choi, Drue Coles, Christian Collberg, George Collins, Eric Conrad, Peter Csaszar, Paul Dietz, Martin Dietzfelbinger, Scot Drysdale, Patricia Ealy, Yaakov Eisenberg, Michael Ernst, Michael Formann, Nedim Fresko, Hal Gabow, Marek Galecki, Igal Galperin, Luisa Gargano, John Gately, Rosario Genario, Mihaly Gereb, Ronald Greenberg, Jerry Grossman, Stephen Guattery, Alexander Hartemik, Anthony Hill, Thomas Hofmeister, Mathew Hostetter, Yih-Chun Hu, Dick Johnsonbaugh, Marcin Jurdzinski, Nabil Kahale, Fumiaki Kamiya, Anand Kanagala, Mark Kantrowitz, Scott Karlin, Dean Kelley, Sanjay Khanna, Haluk Konuk, Dina Kravets, Jon Kroger, Bradley Kuszmaul, Tim Lambert, Hang Lau, Thomas Lengauer, George Madrid, Bruce Maggs, Victor Miller, Joseph Muskat, Tung Nguyen, Michael Orlov, James Park, Seongbin Park, Ιωάννης Πασχαλίδης, Boaz Patt-Shamir, Leonid Peshkin, Patricio Poblete, Ira Pohl, Stephen Ponzio, Kjell Post, Todd Poynor, Colin Prepscious, Sholom Rosen, Dale Russell, Hershel Safer, Karen Seidel, Joel Seiferas, Erik Seligman, Stanley Selkow, Jeffrey Shallit, Greg Shannon, Micha Sharir, Sasha Shen, Norman Shulman, Andrew Singer, Daniel Sleator, Bob Sloan, Michael Sofka, Volker Strumpfen, Lon Sunshine, Julie Sussman, Asterio Tanaka, Clark Thomborson, Nils Thommesen, Homer Tilton, Martin Tompa, Andrei Toom, Felzer Torsten, Hirendu Vaishnav, M. Veldhorst, Luca Venuti, Jian Wang, Michael Wellman, Gerry Wiener, Ronald Williams, David Wolfe, Jeff Wong, Richard Woundy, Neal Young, Huaiyuan Yu, Tian Yuxing, Joe Zachary, Steve Zhang, Florian Zschoke, και Uri Zwick.

Πολλοί από τους συναδέλφους μας προσέφεραν ενδελεχείς ανασκοπήσεις ή συμπλήρωσαν μακροσκελή ερωτηματολόγια. Ευχαριστούμε τους Nancy Amato, Jim Aspnes, Kevin Compton, William Evans, Peter Gacs, Michael Goldwasser, Andrzej Proskurowski, Vijaya Ramachandran, και John Reif για την κριτική ανάγνωση κεφαλαίων. Ευχαριστούμε επίσης τους κάτωθι, για τη συμπλήρωση του ερωτηματολογίου: James Abello, Josh Benaloh, Bryan Beresford-Smith, Kenneth Blaha, Hans Bodlaender, Richard Borie, Ted Brown, Domenico Cantone, M. Chen, Robert Cimikowski, William Clocksin, Paul Cull, Rick Decker, Matthew Dickerson, Robert Douglas, Margaret Fleck, Michael Goodrich, Susanne Hambruch, Dean Hendrix, Richard Johnsonbaugh, Κυριάκο Καλορκώτη, Srinivas Kankanahalli, Hikyo Koh, Steven Lindell, Errol Lloyd, Andy Lopez, Dian Rae Lopez, George Lucker, David Maier, Charles Martel, Xiannong Meng, David Mount, Alberto Policriti, Andrzej Proskurowski, Kirk Pruhs, Yves Robert, Guna Seetharaman, Stanley Selkow, Robert Sloan, Charles Steele, Gerard Tel, Murali Varanasi, Bernd Walter, και Alden Wright. Θα ήμασταν ευτυχείς εάν μπορούσαμε να υλοποιήσουμε όλες τις υποδεί-

ξεις τους. Το μόνο πρόβλημα είναι ότι εάν το κάναμε, η δεύτερη έκδοση θα είχε έκταση περίπου 3000 σελίδες!

Η δεύτερη έκδοση συντάχθηκε στο \TeX 2_ε. Ο Michael Downes μετέτρεψε τις μακροεντολές του \TeX από το «κλασικό» \TeX στο \TeX 2_ε, και προσάρμοσε επίσης τα αρχεία κειμένου προκειμένου να χρησιμοποιούν αυτές τις μακροεντολές. Επίσης, προσέφερε τη βοήθειά του και σε γενικότερα ζητήματα του \TeX 2_ε. Τα σχήματα της δεύτερης έκδοσης σχεδιάστηκαν από τους συγγραφείς με το πρόγραμμα MacDraw Pro. Όπως και στην πρώτη έκδοση, για τη σύνταξη του ευρετηρίου χρησιμοποιήθηκε το Windex, ένα πρόγραμμα σε C το οποίο έχουν κατασκευάσει οι συγγραφείς, ενώ η βιβλιογραφία συντάχθηκε μέσω του προγράμματος Bib \TeX . Οι Ayorkor Mills-Tettey και Rob Leatherne βοήθησαν στη μετατροπή των σχημάτων στο πρόγραμμα MacDraw Pro, ενώ ο Ayorkor έλεγξε επίσης τη βιβλιογραφία.

Όπως και στην πρώτη έκδοση, η συνεργασία με την MIT Press και την McGraw-Hill ήταν ιδιαίτερα ευχάριστη. Οι επιμελητές μας, Bob Prior από την MIT Press και Betsy Jones από την McGraw-Hill, ανέχθηκαν τις ιδιορρυθμίες μας και μας παρότρυναν με «καρότο και μαστίγιο».

Τέλος, θα θέλαμε να ευχαριστήσουμε τις συζύγους μας –Nicole Cormen, Gail Rivest, και Rebecca Ivry– τα παιδιά μας –Ricky, William, και Debby Leiserson· Alex και Christopher Rivest· και Molly, Noah, και Benjamin Stein– και τους γονείς μας –Renee και Perry Cormen, Jean και Mark Leiserson, Shirley και Lloyd Rivest, και Irene και Ira Stein– για την αγάπη και συμπαράστασή τους κατά τη διάρκεια της συγγραφής αυτού του βιβλίου. Η ευόδωση της προσπάθειας αυτής οφείλεται στην υπομονή και την ενθάρρυνση των οικογενειών μας. Τους αφιερώνουμε το βιβλίο αυτό με στοργή.

THOMAS H. CORMEN
CHARLES E. LEISERSON
RONALD L. RIVEST
CLIFFORD STEIN

*Hanover, New Hampshire
Cambridge, Massachusetts
Cambridge, Massachusetts
Hanover, New Hampshire*

Μάιος 2001

Εισαγωγή στους αλγορίθμους

I Θεμελιώδεις έννοιες

Εισαγωγή

Σε αυτό το μέρος του βιβλίου παρουσιάζονται οι πρώτες έννοιες της σχεδίασης και ανάλυσης αλγορίθμων. Στόχος του μέρους αυτού είναι να εισαγάγει τον αναγνώστη ομαλά στη μεθοδολογία του ακριβούς καθορισμού των αλγορίθμων, σε ορισμένες από τις στρατηγικές σχεδίασης που χρησιμοποιούνται γενικότερα στο βιβλίο αυτό, και σε πολλές θεμελιώδεις έννοιες της ανάλυσης αλγορίθμων. Η ύλη αυτών των πρώτων κεφαλαίων αποτελεί τη βάση για τα επόμενα μέρη του βιβλίου.

Στο Κεφάλαιο 1 παρουσιάζεται μια γενική επισκόπηση των αλγορίθμων και του ρόλου τους στα σύγχρονα υπολογιστικά συστήματα. Στο κεφάλαιο αυτό ορίζεται η έννοια του αλγορίθμου και παρατίθενται ορισμένα παραδείγματα. Επιπλέον, τεκμηριώνεται η άποψη ότι οι αλγόριθμοι αποτελούν τεχνολογία, ακριβώς όπως το υλισμικό υψηλής ταχύτητας, οι γραφικές επαφές χρήσεως, τα αντικειμενοστρεφή συστήματα και τα δίκτυα.

Στο Κεφάλαιο 2 παρουσιάζονται οι πρώτοι αλγόριθμοι, οι οποίοι επιλύουν το πρόβλημα της ταξινόμησης μιας ακολουθίας n αριθμών. Οι αλγόριθμοι αυτοί είναι γραμμένοι σε ψευδοκώδικα, ο οποίος αν και δεν μπορεί να μεταφραστεί απευθείας σε καμία από τις συμβατικές γλώσσες προγραμματισμού, αποτυπώνει τη δομή του αλγορίθμου με αρκετή σαφήνεια ούτως ώστε ένας έμπειρος προγραμματιστής να μπορεί να τον υλοποιήσει στη γλώσσα της αρεσκείας του. Οι αλγόριθμοι που εξετάζονται είναι αυτοί της ενθετικής ταξινόμησης, η οποία βασίζεται στην αυξητική προσέγγιση, και της συγχωνευτικής ταξινόμησης, η οποία βασίζεται σε μια αναδρομική τεχνική γνωστή ως «διαίρει-και-κυρίευε». Παρ' όλο που ο χρόνος εκτέλεσης και των δύο αυτών αλγορίθμων αυξάνεται καθώς αυξάνεται η τιμή του n , ο ρυθμός αύξησης διαφέρει μεταξύ των δύο αλγορίθμων. Στο Κεφάλαιο 2 προσδιορίζονται αυτοί οι χρόνοι εκτέλεσης, και εισάγεται ένας εύχρηστος συμβολισμός για την περιγραφή τους.

Στο Κεφάλαιο 3 ορίζεται επακριβώς αυτός ο λεγόμενος ασυμπτωτικός συμβολισμός. Σε πρώτη φάση ορίζονται διάφοροι συμβολισμοί αυτού του τύπου, οι οποίοι χρησιμοποιούνται για τη φραγή του χρόνου εκτέλεσης των αλγορίθμων εκ των άνω και/ή εκ των κάτω. Το υπόλοιπο τμήμα του Κεφαλαίου 3 αποτελεί κατά κύριο λόγο μια παρουσίαση μαθηματικών συμβολισμών. Στόχος αυτού του τμήματος δεν είναι

τόσο η παρουσίαση νέων μαθηματικών εννοιών, αλλά κυρίως η αποσαφήνιση των συμβολιστικών συμβάσεων που ακολουθούνται σε αυτό το βιβλίο.

Στο Κεφάλαιο 4 αναλύεται λεπτομερέστερα η μέθοδος διαιρεί-και-κυρίευε η οποία έχει εισαχθεί στο Κεφάλαιο 2. Συγκεκριμένα, στο Κεφάλαιο 4 παρουσιάζονται μέθοδοι επίλυσης αναδρομικών σχέσεων, οι οποίες χρησιμοποιούνται για την περιγραφή του χρόνου εκτέλεσης αναδρομικών αλγορίθμων. Μια ισχυρή τέτοια τεχνική είναι η λεγόμενη «κεντρική μέθοδος», η οποία μπορεί να χρησιμοποιηθεί για την επίλυση αναδρομικών σχέσεων που προκύπτουν από αλγορίθμους τύπου διαιρεί-και-κυρίευε. Μεγάλο μέρος του Κεφαλαίου 4 είναι αφιερωμένο στην απόδειξη της ορθότητας της κεντρικής μεθόδου, αν και η απόδειξη αυτή μπορεί να παραλειφθεί από τον αναγνώστη.

Το Κεφάλαιο 5 αποτελεί μια εισαγωγή στην πιθανοτική ανάλυση και τους τυχαιοκρατικούς αλγορίθμους. Η πιθανοτική ανάλυση χρησιμοποιείται κατά κανόνα για να προσδιοριστεί ο χρόνος εκτέλεσης ενός αλγορίθμου στις περιπτώσεις όπου, λόγω της εγγενούς ύπαρξης μιας κατανομής πιθανότητας, ο χρόνος εκτέλεσης ενδέχεται να διαφέρει για διαφορετικές εισόδους του ίδιου μεγέθους. Σε ορισμένες περιπτώσεις, υποθέτουμε ότι οι εισοδοί ακολουθούν μια γνωστή κατανομή πιθανότητας, και συνεπώς υπολογίζουμε την αναμενόμενη τιμή του χρόνου εκτέλεσης επί όλων των δυνατών εισόδων. Σε άλλες περιπτώσεις, η κατανομή πιθανότητας δεν προέρχεται από τις εισόδους αλλά από τυχαίες επιλογές που πραγματοποιούνται κατά την πορεία εκτέλεσης του αλγορίθμου. Ένας αλγόριθμος του οποίου η συμπεριφορά δεν καθορίζεται μόνο από την είσοδό του αλλά και από τιμές που παράγονται από μια γεννήτρια τυχαίων αριθμών ονομάζεται τυχαιοκρατικός. Οι τυχαιοκρατικοί αλγόριθμοι μπορούν να χρησιμοποιηθούν για να επιβληθεί μια κατανομή πιθανότητας στις εισόδους –ώστε να εξασφαλιστεί ότι καμία συγκεκριμένη είσοδος δεν θα προκαλεί πάντοτε χαμηλή επίδοση–, ή ακόμη και για να φραγεί ο ρυθμός εμφάνισης σφαλμάτων από αλγορίθμους οι οποίοι επιτρέπεται να παράγουν λανθασμένα αποτελέσματα σε κάποιο περιορισμένο ποσοστό.

Τα Παραρτήματα Α'–Γ' καλύπτουν ύλη μαθηματικού χαρακτήρα η οποία θα σας φανεί χρήσιμη κατά την πορεία της μελέτης αυτού του βιβλίου. Κατά πάσα πιθανότητα, μεγάλο μέρος της ύλης αυτής θα σας είναι ήδη γνωστό (παρ' όλο που σε ορισμένες περιπτώσεις οι συγκεκριμένες συμβολιστικές συμβάσεις που ακολουθούμε πιθανόν να διαφέρουν από αυτές που έχετε συναντήσει στο παρελθόν), και επομένως τα παραρτήματα έχουν περισσότερο χαρακτήρα «ύλης αναφοράς». Από την άλλη πλευρά, η ύλη του Μέρους I πιθανότατα θα σας είναι εν πολλοίς άγνωστη. Όλα τα κεφάλαια του Μέρους I καθώς και τα παραρτήματα είναι γραμμένα σε πνεύμα αναλυτικής διδασκαλίας, και όχι επιγραμματικής παρουσίασης.

1 Ο ρόλος των αλγορίθμων στις υπολογιστικές διαδικασίες

Τι είναι οι αλγόριθμοι; Σε τι μας χρησιμεύει η μελέτη τους; Ποιος είναι ο ρόλος τους σε σχέση με τις άλλες τεχνολογίες που χρησιμοποιούνται στους υπολογιστές; Σκοπός μας σε αυτό το κεφάλαιο είναι να δώσουμε απαντήσεις στα παραπάνω ερωτήματα.

1.1 Αλγόριθμοι

Ο όρος *αλγόριθμος* αναφέρεται σε οποιαδήποτε καλά ορισμένη υπολογιστική διαδικασία που δέχεται κάποια τιμή ή κάποιο σύνολο τιμών ως *είσοδο* και δίνει κάποια τιμή ή κάποιο σύνολο τιμών ως *έξοδο*. Συνεπώς, ένας αλγόριθμος είναι μια ακολουθία υπολογιστικών βημάτων που μετασχηματίζει την είσοδο στην έξοδο.

Μπορούμε επίσης να θεωρήσουμε τον αλγόριθμο ως ένα εργαλείο για την επίλυση ενός καλά καθορισμένου *υπολογιστικού προβλήματος*. Η διατύπωση του προβλήματος καθορίζει σε γενικές γραμμές την επιθυμητή σχέση εισόδου-εξόδου. Ο αλγόριθμος περιγράφει μια συγκεκριμένη υπολογιστική διαδικασία για την επίτευξη αυτής της σχέσης εισόδου-εξόδου.

Ας υποθέσουμε, λόγου χάριν, ότι θέλουμε να ταξινομήσουμε μια ακολουθία αριθμών κατ' αύξουσα σειρά. Το συγκεκριμένο πρόβλημα ανακύπτει συχνά στην πράξη και προσφέρεται για την εισαγωγή πολλών καθιερωμένων σχεδιαστικών τεχνικών και εργαλείων ανάλυσης. Το *πρόβλημα της ταξινόμησης* ορίζεται τυπικά ως εξής:

Είσοδος: Μια ακολουθία n αριθμών $\langle a_1, a_2, \dots, a_n \rangle$.

Έξοδος: Μια μετάθεση (αναδιάταξη) $\langle a'_1, a'_2, \dots, a'_n \rangle$ της ακολουθίας εισόδου τέτοια ώστε $a'_1 \leq a'_2 \leq \dots \leq a'_n$.

Επί παραδείγματι, αν η ακολουθία εισόδου είναι $\langle 31, 41, 59, 26, 41, 58 \rangle$, ένας αλγόριθμος ταξινόμησης επιστρέφει ως έξοδο την ακολουθία $\langle 26, 31, 41, 41, 58, 59 \rangle$. Κάθε τέτοια ακολουθία εισόδου ονομάζεται *στιγμιότυπο* του προβλήματος ταξινόμησης. Εν γένει, ο όρος *στιγμιότυπο προβλήματος* δηλώνει την είσοδο η οποία απαιτείται για να υπολογιστεί μια λύση του προβλήματος (εννοείται ότι η είσοδος αυτή θα ικανοποιεί όλους τους περιορισμούς που επιβάλλει η διατύπωση του προβλήματος).

Η ταξινόμηση αποτελεί βασική πράξη στην επιστήμη υπολογιστών (καθώς χρησιμοποιείται ως ενδιάμεσο βήμα σε πολλά προγράμματα), και για τον λόγο αυτό έχουν αναπτυχθεί αρκετοί ικανοποιητικοί αλγόριθμοι ταξινόμησης. Το ποιος αλγόριθμος είναι ο καταλληλότερος για μια δεδομένη εφαρμογή εξαρτάται, μεταξύ

άλλων, από τον αριθμό των ταξινομητέων στοιχείων, τον βαθμό της αρχικής τους ταξινόμησης, τους πιθανούς περιορισμούς στις τιμές των στοιχείων και το είδος της συσκευής αποθήκευσης που θα χρησιμοποιηθεί: κύρια μνήμη, δίσκος ή μαγνητική ταινία.

Ένας αλγόριθμος χαρακτηρίζεται *ορθός* αν για κάθε στιγμιότυπο εισόδου τερματίζει δίνοντας την ορθή έξοδο. Λέμε ότι ένας ορθός αλγόριθμος *επιλύει* το δεδομένο υπολογιστικό πρόβλημα. Ένας μη ορθός αλγόριθμος μπορεί να μην τερματίζει καν σε ορισμένα στιγμιότυπα εισόδου, ή μπορεί να τερματίζει δίνοντας αποτέλεσμα διαφορετικό από το ζητούμενο. Αντίθετα απ' ό,τι θα υπέθετε ίσως κανείς, ακόμη και ένας μη ορθός αλγόριθμος ενδέχεται μερικές φορές να είναι χρήσιμος, αν το ποσοστό των σφαλμάτων του μπορεί να συγκρατηθεί μέσα σε ορισμένα πλαίσια. Μια τέτοια περίπτωση θα δούμε στο Κεφάλαιο 31, όπου θα μελετήσουμε αλγορίθμους για την εύρεση μεγάλων πρώτων αριθμών. Κατά κανόνα, ωστόσο, οι αλγόριθμοι με τους οποίους θα ασχοληθούμε θα είναι ορθοί.

Ένας αλγόριθμος μπορεί να διατυπωθεί σε μια φυσική γλώσσα, λόγου χάριν στα Αγγλικά, όπως συμβαίνει σε ένα υπολογιστικό πρόγραμμα, ή ακόμη και να αποδοθεί μέσω της σχεδίασης του υλισμικού του υπολογιστή. Η μόνη απαίτηση είναι η διατύπωσή του να αποτελεί ακριβή περιγραφή της υπολογιστικής διαδικασίας που πρέπει να ακολουθηθεί.

Τι είδους προβλήματα επιλύονται με αλγορίθμους;

Η ταξινόμηση δεν είναι βέβαια το μόνο υπολογιστικό πρόβλημα για το οποίο έχουν αναπτυχθεί αλγόριθμοι (μάλλον θα το είχατε ήδη υποψιαστεί από τη στιγμή που είδατε το μέγεθος αυτού του βιβλίου.) Οι πρακτικές εφαρμογές των αλγορίθμων είναι πανταχού παρούσες και περιλαμβάνουν μεταξύ άλλων τα ακόλουθα παραδείγματα:

- Το Πρόγραμμα του Ανθρώπινου Γονιδιώματος, οι στόχοι του οποίου είναι ο προσδιορισμός του συνόλου των 100.000 γονιδίων του ανθρώπινου DNA, η εύρεση της αλληλουχίας των 3 δισεκατομμυρίων ζευγών χημικών βάσεων από τις οποίες αποτελείται το ανθρώπινο DNA, η αποθήκευση αυτών των πληροφοριών σε κατάλληλες βάσεις δεδομένων, και η ανάπτυξη εργαλείων για την ανάλυση των δεδομένων. Καθένα από τα παραπάνω βήματα απαιτεί περίτεχνους αλγορίθμους. Αν και οι λύσεις στα διάφορα προβλήματα που υπεισέρχονται στο συγκεκριμένο πρόγραμμα υπερβαίνουν τον ορίζοντα αυτού του βιβλίου, πολλές από τις έννοιες που παρουσιάζονται στα κεφάλαια που ακολουθούν χρησιμοποιούνται για την επίλυση αυτών των προβλημάτων της βιολογίας, δίνοντας στους επιστήμονες τη δυνατότητα να επιτελέσουν το έργο τους αξιοποιώντας αποδοτικά τους πόρους που έχουν στη διάθεσή τους. Με τον τρόπο αυτό εξοικονομείται χρόνος, τόσο ανθρώπινος όσο και υπολογιστικός, και χρήματα, αφού μπορούν να εξαχθούν περισσότερες πληροφορίες από τις διάφορες εργαστηριακές εργασίες.
- Το Διαδίκτυο δίνει τη δυνατότητα σε ανθρώπους από όλο τον κόσμο να προσπελάσουν και να αποκτήσουν γρήγορα μεγάλες ποσότητες πληροφοριών. Η δυνατότητα αυτή έχει επιτευχθεί με τη βοήθεια έξυπνων αλγορίθμων οι οποίοι διαχειρίζονται αυτούς τους μεγάλους όγκους δεδομένων. Τα προβλήματα που υπεισέρχονται σε αυτήν την παγκόσμια επικοινωνία περιλαμβάνουν την εύρεση κατάλληλων διαδρομών για τη μεταφορά των δεδομένων (τεχνικές για την επίλυση τέτοιων προβλημάτων παρουσιάζονται στο Κεφάλαιο 24), και τη χρήση

μηχανών αναζήτησης για την γρήγορη εύρεση των ιστοσελίδων στις οποίες εντοπίζονται συγκεκριμένες πληροφορίες (σχετικές τεχνικές περιγράφονται στα Κεφάλαια 11 και 32).

- Το ηλεκτρονικό εμπόριο δίνει τη δυνατότητα αγοραπωλησίας και διακίνησης αγαθών και υπηρεσιών με ηλεκτρονικά μέσα. Η δυνατότητα ασφαλούς επεξεργασίας εμπιστευτικών πληροφοριών, όπως είναι οι αριθμοί πιστωτικών καρτών, οι συνθηματικοί κωδικοί, και τα αντίγραφα κίνησης τραπεζικών λογαριασμών, αποτελεί ουσιώδη προϋπόθεση για την ευρεία χρήση του ηλεκτρονικού εμπορίου. Μεταξύ των βασικών τεχνικών που χρησιμοποιούνται σε αυτόν τον τομέα είναι η κρυπτογραφία δημόσιου κλειδιού και οι ψηφιακές υπογραφές (τις οποίες θα μελετήσουμε στο Κεφάλαιο 31). Οι συγκεκριμένες τεχνικές βασίζονται σε αριθμητικούς αλγόριθμους και στη θεωρία αριθμών.
- Στη βιομηχανική παραγωγή και σε άλλες εμπορικές δραστηριότητες συχνά είναι ιδιαίτερα σημαντικό να μπορούμε να καταναείμουμε ανεπαρκείς πόρους με τον επωφελέστερο δυνατό τρόπο. Μια εταιρεία πετρελαίου, λόγω χάριν, πιθανόν να επιθυμεί να γνωρίζει πού να εγκαταστήσει τις πετρελαιοπηγές της προκειμένου να μεγιστοποιήσει το αναμενόμενο κέρδος. Ένας υποψήφιος για την προεδρία των ΗΠΑ πιθανόν να θέλει να προσδιορίσει σε ποιες περιοχές της χώρας θα πρέπει να επενδύσει χρήματα για να διαφημίσει την προεκλογική του εκστρατεία ώστε να μεγιστοποιήσει τις πιθανότητές του να κερδίσει τις εκλογές. Μια αεροπορική εταιρεία ίσως να επιθυμεί να καταναείμει τα πληρώματα στις διάφορες πτήσεις με τον λιγότερο δαπανηρό τρόπο, εξασφαλίζοντας ταυτόχρονα ότι όλες οι πτήσεις καλύπτονται επαρκώς και ότι ικανοποιούνται οι πτητικοί κανονισμοί για τον χρονοπρογραμματισμό των πληρωμάτων. Μια εταιρεία παροχής Διαδικτυακών υπηρεσιών πιθανόν να επιθυμεί να προσδιορίσει σε ποια σημεία θα πρέπει να εγκαταστήσει επιπλέον πόρους προκειμένου να ικανοποιήσει αποτελεσματικότερα τις ανάγκες των πελατών της. Όλες οι παραπάνω περιπτώσεις αποτελούν παραδείγματα προβλημάτων που μπορούν να επιλυθούν με τη βοήθεια του γραμμικού προγραμματισμού, τον οποίο θα μελετήσουμε στο Κεφάλαιο 29.

Στο βιβλίο αυτό, αν και δεν θα επεκταθούμε σε λεπτομερή ανάλυση των προηγούμενων παραδειγμάτων, θα μελετήσουμε τις θεμελιώδεις τεχνικές που εφαρμόζονται σε αυτά τα προβλήματα και σε αυτές τις περιοχές προβλημάτων. Θα περιγράψουμε επίσης τη λύση πολλών συγκεκριμένων προβλημάτων, όπως π.χ. τα παρακάτω:

- Μας δίνεται ένας οδικός χάρτης ο οποίος αναγράφει όλες τις αποστάσεις μεταξύ παρακείμενων διασταυρώσεων, και μας ζητείται να προσδιορίσουμε τη βραχύτερη διαδρομή από κάποια διασταύρωση μέχρι κάποια άλλη. Ακόμη και αν αποκλείσουμε τις διαδρομές που τέμνουν τον εαυτό τους, ο αριθμός των δυνατών διαδρομών μπορεί να είναι τεράστιος. Πώς θα βρούμε ποια απ' όλες είναι η βραχύτερη; Στην περίπτωση αυτή, αναπαριστούμε τον οδικό χάρτη (ο οποίος, με τη σειρά του, είναι επίσης μια αναπαράσταση του πραγματικού οδικού δικτύου) με τη μορφή ενός γραφήματος (τα γραφήματα περιγράφονται στο Κεφάλαιο 10 και στο Παράρτημα Β'), και ανάγουμε το πρόβλημα στην εύρεση της βραχύτερης διαδρομής από κάποιον κόμβο του γραφήματος μέχρι κάποιον άλλον. Μια δραστική επίλυση αυτού του προβλήματος παρουσιάζεται στο Κεφάλαιο 24.

- Μας δίνεται μια ακολουθία n πινάκων $\langle A_1, A_2, \dots, A_n \rangle$, και μας ζητείται να υπολογίσουμε το γινόμενο τους $A_1 A_2 \cdots A_n$. Δεδομένου ότι ο πολλαπλασιασμός πινάκων είναι προσεταιριστικός, υπάρχουν διάφορες επιτρεπτές πολλαπλασιαστικές διατάξεις. Παραδείγματος χάριν, θα μπορούσαμε να εκτελέσουμε τους πολλαπλασιασμούς με οποιαδήποτε από τις ακόλουθες ομαδοποιήσεις της ακολουθίας πινάκων: $(A_1(A_2(A_3 A_4)))$, $(A_1((A_2 A_3) A_4))$, $((A_1 A_2)(A_3 A_4))$, $((A_1(A_2 A_3)) A_4)$, ή $((A_1 A_2) A_3) A_4$. Αν όλοι οι πίνακες είναι τετραγωνικοί (και επομένως ισομεγέθεις), η χρονική διάρκεια του πολλαπλασιασμού είναι ανεξάρτητη από την ομαδοποίηση που θα χρησιμοποιήσουμε. Αν, όμως, οι πίνακες έχουν διαφορετικά μεγέθη (τα οποία ωστόσο είναι συμβατά μεταξύ τους όσον αφορά τον πολλαπλασιασμό), τότε η ομαδοποίησή τους μπορεί να έχει σημαντική επίπτωση στον χρόνο εκτέλεσης. Ο αριθμός των δυνατών πολλαπλασιαστικών ομαδοποιήσεων αυξάνεται εκθετικά συναρτήσει του n , και επομένως η δοκιμή όλων των ομαδοποιήσεων μπορεί να είναι εξαιρετικά χρονοβόρα. Όπως θα δούμε στο Κεφάλαιο 15, το πρόβλημα αυτό μπορεί να λυθεί πολύ πιο ικανοποιητικά μέσω μιας γενικής τεχνικής που ονομάζεται δυναμικός προγραμματισμός.
- Μας δίνεται μια εξίσωση $ax \equiv b \pmod{n}$, όπου τα a , b , και n είναι ακέραιοι, και μας ζητείται να βρούμε όλους τους ακεραίους x , modulo n , οι οποίοι την ικανοποιούν. Μια τέτοια εξίσωση πιθανόν να μην έχει καμία λύση, ή να έχει μία ή και περισσότερες λύσεις. Αν και θα μπορούσαμε απλώς να δοκιμάσουμε να θέσουμε διαδοχικά $x = 0, 1, \dots, n - 1$, όπως θα δούμε στο Κεφάλαιο 31 το πρόβλημα μπορεί να λυθεί πιο αποτελεσματικά.
- Μας δίνονται n σημεία στο επίπεδο, και μας ζητείται να βρούμε το κυρτό κάλυμμα των σημείων αυτών, δηλαδή το μικρότερο κυρτό πολύγωνο που περιέχει όλα τα σημεία. Χάριν εποπτείας, μπορούμε να φανταστούμε το κάθε σημείο σαν ένα καρφί που προεξέχει από την επιφάνεια μιας ξύλινης πινακίδας. Στην απεικόνιση αυτή, το κυρτό κάλυμμα θα αναπαρίσταται από μια τεντωμένη ελαστική ταινία που περιβάλλει το σύνολο των καρφιών. Το κάθε καρφί γύρω από το οποίο κάμπτεται η ταινία είναι μια κορυφή του κυρτού καλύμματος (βλ. Σχήμα 33.6, π.χ.). Εν γένει, το σύνολο των κορυφών του καλύμματος μπορεί να είναι οποιοδήποτε από τα 2^n υποσύνολα των σημείων. Επιπλέον, δεν αρκεί να γνωρίζουμε ποια από τα σημεία είναι κορυφές του καλύμματος: θα πρέπει επίσης να γνωρίζουμε τη διάταξη των σημείων αυτών. Επομένως, υπάρχουν πολλές επιλογές για τις κορυφές του κυρτού καλύμματος. Στο Κεφάλαιο 33 θα παρουσιάσουμε δύο καλές μεθόδους για την εύρεση του κυρτού καλύμματος.

Τα παραπάνω παραδείγματα, αν και δεν εξαντλούν επ' ουδενί το σύνολο των σημαντικών περιπτώσεων (όπως και πάλι θα έχετε αντιληφθεί από το βάρος αυτού του βιβλίου), διαθέτουν δύο χαρακτηριστικά που είναι κοινά σε πολλούς ενδιαφέροντες αλγορίθμους.

1. Υπάρχουν πολλές υποψήφιας λύσεις, οι περισσότερες από τις οποίες δεν ανταποκρίνονται στις απαιτήσεις μας. Η εύρεση μιας λύσης που να ικανοποιεί τις απαιτήσεις μας μπορεί να παρουσιάζει αρκετές δυσκολίες.
2. Υπάρχουν πρακτικές εφαρμογές. Το πιο χαρακτηριστικό παράδειγμα μεταξύ των παραπάνω προβλημάτων είναι αυτό των βραχύτερων διαδρομών. Για μια μεταφορική εταιρεία, όπως είναι λόγου χάριν ένας σιδηροδρομικός οργανισμός

ή μια εταιρεία οδικών μεταφορών, η εύρεση των βραχύτερων διαδρομών σε ένα σιδηροδρομικό ή οδικό δίκτυο έχει άμεσο οικονομικό ενδιαφέρον, καθώς η υιοθέτηση των διαδρομών αυτών συνεπάγεται μικρότερες δαπάνες για αμοιβές προσωπικού και καύσιμα. Για έναν κόμβο δρομολόγησης στο Διαδίκτυο, η εύρεση της βραχύτερης διαδρομής στο δίκτυο επιτρέπει την ταχύτερη μεταβίβαση ενός μηνύματος.

Δομές δεδομένων

Στο βιβλίο αυτό θα ασχοληθούμε επίσης με διάφορες δομές δεδομένων. Μια *δομή δεδομένων* είναι ένας τρόπος αποθήκευσης και οργάνωσης των δεδομένων ο οποίος διευκολύνει την προσπέλαση και την τροποποίησή τους. Δεδομένου ότι δεν υπάρχει μία και μοναδική δομή δεδομένων που να καλύπτει ικανοποιητικά όλες τις ανάγκες που εμφανίζονται στις εφαρμογές, θα πρέπει κανείς να γνωρίζει τα πλεονεκτήματα και τους περιορισμούς διαφόρων τέτοιων δομών.

Τεχνική

Αν και μπορείτε να χρησιμοποιήσετε το βιβλίο αυτό σαν «συνταγολόγιο» έτοιμων αλγορίθμων, είναι πιθανό σε κάποια εφαρμογή να συναντήσετε ένα πρόβλημα για το οποίο να μην μπορείτε να βρείτε εύκολα κάποιον σχετικό αλγόριθμο στα περιεχόμενά του (αυτό ισχύει π.χ. για πολλές από τις ασκήσεις και τα προβλήματα που παρατίθενται στο τέλος των ενότητων!). Στο βιβλίο αυτό, θα μελετήσουμε τεχνικές σχεδίασης και ανάλυσης αλγορίθμων οι οποίες θα σας επιτρέψουν να αναπτύσσετε τους δικούς σας αλγορίθμους, να αποδεικνύετε ότι δίνουν το σωστό αποτέλεσμα και να αξιολογείτε την επίδοσή τους.

Δύσκολα προβλήματα

Το μεγαλύτερο μέρος αυτού του βιβλίου αφορά δραστικούς αλγορίθμους. Ένα συνηθισμένο μέτρο της δραστικότητας είναι η ταχύτητα, δηλαδή το πόσοι χρόνος χρειάζεται για να παραγάγει ένας αλγόριθμος το αποτέλεσμά του. Υπάρχουν ορισμένα προβλήματα, όμως, για τα οποία δεν γνωρίζουμε καμία δραστική λύση. Στο Κεφάλαιο 34 θα μελετήσουμε ένα ενδιαφέρον υποσύνολο αυτών των προβλημάτων, τα λεγόμενα NP-πλήρη προβλήματα.

Τι είναι αυτό που κάνει τα NP-πλήρη προβλήματα ενδιαφέροντα; Κατ' αρχάς, παρ' όλο που δεν έχει βρεθεί ποτέ κάποιος δραστικός αλγόριθμος για ένα τέτοιο πρόβλημα, κανείς δεν έχει αποδείξει ότι δεν μπορεί να υπάρξει τέτοιος αλγόριθμος. Με άλλα λόγια, δεν γνωρίζουμε αν υπάρχει δραστικός αλγόριθμος για NP-πλήρη προβλήματα. Δεύτερον, το σύνολο των προβλημάτων αυτών έχει την αξιοσημείωτη ιδιότητα ότι αν υπάρχει δραστικός αλγόριθμος για οποιοδήποτε από αυτά, τότε υπάρχουν τέτοιοι αλγόριθμοι για όλα. Αυτή η σχέση μεταξύ των NP-πλήρων προβλημάτων κάνει την απουσία δραστικών λύσεων ακόμη πιο προκλητική. Τρίτον, αρκετά NP-πλήρη προβλήματα είναι παρόμοια, αλλά όχι πανομοιότυπα, με προβλήματα για τα οποία γνωρίζουμε δραστικούς αλγορίθμους. Μια μικρή αλλαγή στη διατύπωση του προβλήματος μπορεί να συνεπάγεται τεράστια αλλαγή στη δραστικότητα του καλύτερου γνωστού αλγορίθμου.

Το να γνωρίζει κανείς την ύπαρξη των NP-πλήρων προβλημάτων έχει μεγάλη πρακτική σημασία, διότι ορισμένα από αυτά ανακύπτουν αναπάντεχα συχνά στις

πραγματικές εφαρμογές. Αν έχετε αναλάβει να αναπτύξετε έναν δραστικό αλγόριθμο για ένα NP-πλήρες πρόβλημα, είναι αρκετά πιθανό να σπαταλήσετε αρκετό χρόνο σε μια μάταιη προσπάθεια. Αν, όμως, μπορείτε να αποδείξετε ότι το δεδομένο πρόβλημα είναι NP-πλήρες, μπορείτε να αφιερώσετε τον χρόνο σας στην ανάπτυξη ενός δραστικού αλγορίθμου ο οποίος θα δίνει μια ικανοποιητική, αλλά όχι τη βέλτιστη, λύση.

Ως συγκεκριμένο παράδειγμα, ας εξετάσουμε μια εταιρεία οδικών μεταφορών με μια κεντρική αποθήκη. Κάθε μέρα, η εταιρεία φορτώνει το φορτηγό της στην αποθήκη και καθορίζει ένα δρομολόγιο για την παράδοση των εμπορευμάτων. Στο τέλος της μέρας, το φορτηγό θα πρέπει να έχει επιστρέψει στην αποθήκη ώστε να είναι έτοιμο να φορτώσει τα εμπορεύματα της επομένης. Για να μειώσει τα έξοδά της, η εταιρεία επιδιώκει η σειρά που θα επιλέξει για τα σημεία παράδοσης να δίνει την ελάχιστη ολική απόσταση. Το συγκεκριμένο πρόβλημα είναι το γνωστό «πρόβλημα του περιοδεύοντος πωλητή», και είναι NP-πλήρες. Δεν υπάρχει γνωστός δραστικός αλγόριθμος για την επίλυσή του. Υπό ορισμένες προϋποθέσεις, όμως, υπάρχουν δραστικοί αλγόριθμοι που δίνουν μια ολική απόσταση η οποία δεν υπερβαίνει σημαντικά την ελάχιστη δυνατή. Στο Κεφάλαιο 35 θα μελετήσουμε τέτοιους «προσεγγιστικούς αλγορίθμους».

Ασκήσεις

1.1-1

Αναφέρετε μια περίπτωση της καθημερινής ζωής στην οποία εμφανίζεται ένα από τα παρακάτω υπολογιστικά προβλήματα: ταξινόμηση, προσδιορισμός της καλύτερης ομαδοποίησης για πολλαπλασιασμό πινάκων, εύρεση του κυρτού καλύμματος.

1.1-2

Εκτός από την ταχύτητα, ποια άλλα μέτρα της «δραστικότητας» θα μπορούσε να χρησιμοποιήσει κανείς σε πρακτικά ζητήματα της καθημερινής ζωής;

1.1-3

Επιλέξτε μια δομή δεδομένων που γνωρίζετε ήδη, και αναλύστε τα πλεονεκτήματα και τους περιορισμούς της.

1.1-4

Σε τι μοιάζουν μεταξύ τους τα προβλήματα της βραχύτερης διαδρομής και του περιοδεύοντος πωλητή που αναφέραμε παραπάνω; Σε τι διαφέρουν;

1.1-5

Εντοπίστε ένα πρόβλημα της καθημερινής ζωής για το οποίο η μόνη ικανοποιητική λύση είναι η βέλτιστη. Στη συνέχεια, εντοπίστε ένα πρόβλημα για το οποίο μια λύση που «προσεγγίζει» τη βέλτιστη είναι επαρκής.

1.2 Οι αλγόριθμοι ως τεχνολογία

Ας υποθέσουμε ότι οι υπολογιστές ήταν απείρως ταχείς και η υπολογιστική μνήμη ήταν δωρεάν. Στην υποθετική αυτή περίπτωση, θα υπήρχε άραγε κανένας λόγος να

μελετήσετε τους αλγόριθμους; Η απάντηση είναι ναι, αν μη τι άλλο επειδή ακόμη και σε αυτήν την περίπτωση θα θέλατε να αποδείξετε ότι η μέθοδος επίλυσης που αναπτύξατε τερματίζει δίνοντας το σωστό αποτέλεσμα.

Αν οι υπολογιστές ήταν απείρως ταχείς, οποιαδήποτε ορθή μέθοδος επίλυσης ενός προβλήματος θα αρκούσε. Αν και θα επιδιώκατε ενδεχομένως η υλοποίησή σας να ικανοποιεί τα παραδεκτά πρότυπα της κατασκευής λογισμικού (δηλαδή να είναι καλά σχεδιασμένη και τεκμηριωμένη), στις περισσότερες περιπτώσεις θα χρησιμοποιούσατε οποιαδήποτε μέθοδο θα ήταν ευκολότερο να υλοποιηθεί.

Φυσικά, οι πραγματικοί υπολογιστές είναι μεν ταχείς, αλλά όχι απείρως ταχείς. Και η μνήμη μπορεί να είναι μεν φτηνή, αλλά δεν είναι δωρεάν. Κατά συνέπεια, ο υπολογιστικός χρόνος είναι ένας πεπερασμένος πόρος, όπως και η μνήμη. Οι πόροι αυτοί θα πρέπει να χρησιμοποιούνται με σύνεση, και αυτόν ακριβώς τον σκοπό εξυπηρετούν οι αλγόριθμοι που είναι δραστικοί ως προς τον χρόνο και οικονομικοί ως προς τον χώρο.

Δραστηκότητα

Συχνά για το ίδιο πρόβλημα είναι δυνατόν να αναπτυχθούν αλγόριθμοι οι οποίοι να διαφέρουν συντριπτικά ως προς τη δραστηκότητα. Οι διαφορές αυτές μπορεί να είναι πολύ πιο σημαντικές από εκείνες που οφείλονται στο υλισμικό και το λογισμικό.

Παραδείγματος χάριν, στο Κεφάλαιο 2 θα μελετήσουμε δύο αλγόριθμους ταξινόμησης. Ο πρώτος, η λεγόμενη *ενθετική ταξινόμηση*, απαιτεί χρόνο περίπου ίσο με $c_1 n^2$ για την ταξινόμηση n στοιχείων, όπου c_1 είναι μια σταθερά ανεξάρτητη του n . Δηλαδή, απαιτεί χρόνο περίπου ανάλογο του n^2 . Για τον δεύτερο αλγόριθμο, τη *συγχωνευτική ταξινόμηση*, ο απαιτούμενος χρόνος είναι περίπου ίσος με $c_2 n \lg n$, όπου το σύμβολο $\lg n$ σημαίνει $\log_2 n$ και c_2 είναι μια άλλη σταθερά, επίσης ανεξάρτητη του n . Η ενθετική ταξινόμηση συνήθως έχει μικρότερο σταθερό συντελεστή από τη συγχωνευτική, δηλαδή $c_1 < c_2$. Όπως θα δούμε, οι σταθεροί συντελεστές μπορεί να έχουν πολύ μικρότερη επίπτωση στους σχετικούς χρόνους εκτέλεσης σε σχέση με την εξάρτηση από το μέγεθος n της εισόδου. Αντί του συντελεστή $\lg n$ που εμφανίζεται στον χρόνο εκτέλεσης της συγχωνευτικής ταξινόμησης, η ενθετική έχει συντελεστή n , ο οποίος είναι πολύ μεγαλύτερος. Αν και η ενθετική ταξινόμηση συνήθως είναι ταχύτερη της συγχωνευτικής για μικρό μέγεθος n της εισόδου, από τη στιγμή που το μέγεθος αυτό θα γίνει αρκετά μεγάλο, το πλεονέκτημα του συντελεστή $\lg n$ στη συγχωνευτική ταξινόμηση έναντι του n υπεραντισταθμίζει τη διαφορά των σταθερών συντελεστών. Όσο μικρότερο και αν είναι το c_1 σε σχέση με το c_2 , θα υπάρχει πάντα κάποιο σημείο «χρονικής ισοφάρισης», πέραν του οποίου η συγχωνευτική ταξινόμηση θα είναι ταχύτερη.

Ας δούμε ένα συγκεκριμένο παράδειγμα: ας βάλουμε έναν ταχύ υπολογιστή (τον υπολογιστή Α) ο οποίος εκτελεί την ενθετική ταξινόμηση να «αναμετρηθεί» με έναν βραδύτερο (τον υπολογιστή Β) ο οποίος εκτελεί τη συγχωνευτική ταξινόμηση. Ο κάθε υπολογιστής θα πρέπει να ταξινομήσει ένα εκατομμύριο αριθμούς. Έστω ότι ο υπολογιστής Α εκτελεί ένα δισεκατομμύριο εντολές το δευτερόλεπτο και ο Β μόνο δέκα εκατομμύρια εντολές το δευτερόλεπτο. Με άλλα λόγια ο Α είναι 100 φορές ταχύτερος του Β. Για να κάνουμε τη διαφορά ακόμη πιο συντριπτική, ας υποθέσουμε ότι ο κώδικας της ενθετικής ταξινόμησης που εκτελεί ο υπολογιστής Α έχει γραφτεί σε γλώσσα μηχανής από τον πιο επιδέξιο προγραμματιστή του κόσμου, και ότι

ο απαιτούμενος αριθμός εντολών για την ταξινόμηση n αριθμών είναι $2n^2$ (δηλαδή, στην περίπτωση μας $c_1 = 2$). Από την άλλη πλευρά, ο κώδικας της συγχωνευτικής ταξινόμησης του υπολογιστή B έχει γραφτεί από έναν κοινό προγραμματιστή σε γλώσσα υψηλού επιπέδου με μέτριο μεταγλωττιστή, με αποτέλεσμα να απαιτεί $50n \lg n$ εντολές (δηλαδή, $c_2 = 50$). Για την ταξινόμηση ενός εκατομμυρίου αριθμών, ο υπολογιστής A χρειάζεται

$$\frac{2 \cdot (10^6)^2 \text{ εντολές}}{10^9 \text{ εντολές/δευτερόλεπτο}} = 2000 \text{ δευτερόλεπτα},$$

ενώ ο B χρειάζεται

$$\frac{50 \cdot 10^6 \lg 10^6 \text{ εντολές}}{10^7 \text{ εντολές/δευτερόλεπτο}} \approx 100 \text{ δευτερόλεπτα}.$$

Χρησιμοποιώντας έναν αλγόριθμο του οποίου ο χρόνος εκτέλεσης αυξάνεται βραδύτερα συναρτήσει του n , ο υπολογιστής B, ακόμη και με μέτριο μεταγλωττιστή, εκτελεί την ίδια εργασία 20 φορές ταχύτερα από τον A! Η υπεροχή της συγχωνευτικής ταξινόμησης γίνεται ακόμη πιο εξόφθαλμη αν επιχειρήσουμε να ταξινομήσουμε δέκα εκατομμύρια αριθμούς: ενώ η ενθετική ταξινόμηση θα διαρκέσει περίπου 2,3 ημέρες, η συγχωνευτική θα ολοκληρωθεί σε λιγότερο από 20 λεπτά. Εν γένει, όσο αυξάνεται το μέγεθος του προβλήματος, τόσο αυξάνεται και το συγκριτικό πλεονέκτημα της συγχωνευτικής ταξινόμησης.

Οι αλγόριθμοι και οι άλλες τεχνολογίες

Το παραπάνω παράδειγμα καταδεικνύει ότι οι αλγόριθμοι είναι ένα είδος *τεχνολογίας*, όπως είναι λ.χ. το υλισμικό του υπολογιστή. Η ολική απόδοση ενός συστήματος εξαρτάται τόσο από την επιλογή υλισμικού υψηλής ταχύτητας όσο και από την επιλογή δραστικών αλγορίθμων. Ο τομέας των αλγορίθμων χαρακτηρίζεται από ταχεία ανάπτυξη, όπως συμβαίνει και με τις υπόλοιπες τεχνολογίες των υπολογιστών.

Πιθανόν να αναρωτιέστε αν οι αλγόριθμοι είναι πράγματι τόσο σημαντικοί για τη λειτουργία των σύγχρονων υπολογιστών, τη στιγμή που υπάρχουν άλλες προηγμένες τεχνολογίες, όπως λόγου χάριν

- υλισμικό με υψηλή συχνότητα ρολογιού, ομοχειριακή αρχιτεκτονική, και υπερταυτοχρονική αρχιτεκτονική,
- εύχρηστες, παραστατικές γραφικές επαφές χρήσεως (Graphical User Interface - GUI),
- αντικειμενοστρεφή συστήματα, και
- τοπικά δίκτυα και δίκτυα ευρείας περιοχής.

Η απάντηση είναι ναι. Αν και υπάρχουν ορισμένες εφαρμογές που δεν απαιτούν ρητά κάποια αλγοριθμική συνιστώσα στο επίπεδο της εφαρμογής (λ.χ. μερικές απλές «ιστοκεντρικές» εφαρμογές), οι περισσότερες εμπεριέχουν καθ' εαυτές κάποιο αλγοριθμικό σκέλος. Ας εξετάσουμε, παραδείγματος χάριν, μια ιστοκεντρική υπηρεσία που παρέχει πληροφορίες για το πώς μπορείτε να ταξιδέψετε από κάποια τοποθεσία σε κάποια άλλη. (Τη στιγμή που γραφόταν αυτό το βιβλίο υπήρχαν διάφορες τέτοιες υπηρεσίες.) Η υλοποίησή της θα βασιζόταν σε υλισμικό υψηλής ταχύτητας,

σε μια γραφική επαφή χρήσεως, στην ύπαρξη κάποιου δικτύου ευρείας περιοχής, και ενδεχομένως σε κάποιο αντικειμενοστρεφές σύστημα. Εντούτοις, η συγκεκριμένη υπηρεσία θα απαιτούσε επίσης κάποιους αλγόριθμους για ορισμένες λειτουργίες όπως είναι λόγου χάριν η εύρεση διαδρομών (πιθανότατα μέσω κάποιου αλγορίθμου βραχύτερης διαδρομής), η απεικόνιση χαρτών και η παρεμβολική εύρεση διευθύνσεων.

Εξάλλου, ακόμη και μια εφαρμογή που δεν απαιτεί καμία αλγοριθμική συνιστώσα στο επίπεδο της εφαρμογής εξαρτάται καθοριστικά από αλγόριθμους. Δεν βασίζεται η εφαρμογή σε υλισμικό υψηλής ταχύτητας; Η σχεδίαση του υλισμικού έγινε με τη βοήθεια αλγορίθμων. Δεν βασίζεται η εφαρμογή σε γραφικές επαφές χρήσεως; Η σχεδίαση οποιασδήποτε τέτοιας επαφής βασίζεται σε αλγόριθμους. Δεν βασίζεται η εφαρμογή σε κάποιο δίκτυο; Η δρομολόγηση στα δίκτυα βασίζεται καθοριστικά σε αλγόριθμους. Επιπλέον, αν η εν λόγω εφαρμογή γράφτηκε σε κάποια γλώσσα διαφορετική από τον κώδικα μηχανής, τότε υπέστη επεξεργασία από κάποιον μεταγλωττιστή, διερμηνέα ή συμβολομεταφραστή, προγράμματα στα οποία χρησιμοποιούνται ευρύτατα αλγόριθμοι. Οι αλγόριθμοι βρίσκονται στον πυρήνα των περισσότερων τεχνολογιών οι οποίες χρησιμοποιούνται στους σύγχρονους υπολογιστές.

Επιπλέον, καθώς οι υπολογιστές αναβαθμίζονται διαρκώς όσον αφορά τις λειτουργικές τους δυνατότητες, χρησιμοποιούνται για την επίλυση ολοένα και μεγαλύτερων προβλημάτων. Όπως είδαμε παραπάνω στη σύγκριση ενθετικής και συγχωνευτικής ταξινόμησης, οι διαφορές στην αποτελεσματικότητα των αλγορίθμων γίνονται ιδιαίτερα εμφανείς ακριβώς όταν το μέγεθος των προβλημάτων είναι μεγάλο.

Η καλή γνώση των αλγορίθμων και των αλγοριθμικών τεχνικών είναι ένα από τα χαρακτηριστικά που διακρίνουν τους πραγματικά έμπειρους προγραμματιστές από τους αρχάριους. Αν και με τη σύγχρονη τεχνολογία των υπολογιστών μπορείτε να διεκπεραιώσετε κάποιες εργασίες χωρίς να έχετε ιδιαίτερες γνώσεις αλγορίθμων, ωστόσο ένα καλό γνωστικό υπόβαθρο στους αλγόριθμους θα σας επιτρέψει να επιτύχετε πολύ περισσότερα.

Ασκήσεις

1.2-1

Αναφέρετε ένα παράδειγμα εφαρμογής η οποία απαιτεί αλγοριθμική συνιστώσα στο επίπεδο της εφαρμογής, και αναλύστε τη λειτουργία των αλγορίθμων που υπεισέρχονται σε αυτήν.

1.2-2

Υποθέστε ότι συγκρίνουμε την υλοποίηση της ενθετικής και της συγχωνευτικής ταξινόμησης στον ίδιο υπολογιστή. Για εισόδους μεγέθους n , η ενθετική ταξινόμηση απαιτεί $8n^2$ βήματα, ενώ η συγχωνευτική $64n \lg n$. Για ποιες τιμές του n η ενθετική ταξινόμηση υπερτερεί της συγχωνευτικής;

1.2-3

Ποια είναι η μικρότερη τιμή του n για την οποία ένας αλγόριθμος με χρόνο εκτέλεσης $100n^2$ εκτελείται ταχύτερα από έναν άλλο με χρόνο εκτέλεσης 2^n στον ίδιο υπολογιστή;

Προβλήματα

1-1 Σύγκριση χρόνων εκτέλεσης

Για κάθε συνάρτηση $f(n)$ και για κάθε χρονικό διάστημα t που παρατίθεται στον παρακάτω πίνακα, προσδιορίστε το μέγιστο μέγεθος n του προβλήματος που μπορεί να επιλυθεί σε χρόνο t , υποθέτοντας ότι ο αλγόριθμος που επιλύει το πρόβλημα απαιτεί χρόνο $f(n)$ μικροδευτερόλεπτα.

	1 δευτερόλεπτο	1 λεπτό	1 ώρα	1 ημέρα	1 μήνας	1 έτος	1 αιώνας
$\lg n$							
\sqrt{n}							
n							
$n \lg n$							
n^2							
n^3							
2^n							
$n!$							

Σημειώσεις κεφαλαίου

Υπάρχουν πολλά εξαιρετικά εγχειρίδια γενικού χαρακτήρα για αλγορίθμους, μεταξύ των οποίων αυτά των Aho, Hopcroft, και Ullman [5, 6], Baase και Van Gelder [26], Brassard και Bratley [46, 47], Goodrich και Tamassia [128], Horowitz, Sahni, και Rajasekaran [158], Kingston [179], Knuth [182, 183, 185], Kozen [193], Manber [210], Mehlhorn [217, 218, 219], Purdom και Brown [252], Reingold, Nievergelt, και Deo [257], Sedgewick [269], Skiena [280], και Wilf [315]. Τα βιβλία των Bentley [39, 40] και Gonnet [126] αναλύουν ορισμένες πιο πρακτικές πλευρές της σχεδίασης αλγορίθμων. Για μια γενική επισκόπηση του πεδίου των αλγορίθμων, μπορεί κανείς να συμβουλευθεί το Handbook of Theoretical Computer Science, Volume A [302] και το CRC Handbook on Algorithms and Theory of Computation [24]. Μια συνοπτική περιγραφή των αλγορίθμων που χρησιμοποιούνται στην υπολογιστική βιολογία παρατίθεται στα εγχειρίδια των Gusfield [136], Pevzner [240], Setubal και Meidanis [272], και Waterman [309].