Πρόλογος

Στόχος αυτών των σημειώσεων είναι η παρουσίαση των βασικών εννοιών και αλγορίθμων της Υπολογιστικής Γεωμετρίας. Από την καθιέρωσή του στις αρχές της δεκαετίας του '70, ο τομέας της Υπολογιστικής Γεωμετρίας έχει παρουσιάσει πολύ μεγάλη ανάπτυξη και έχει αποδώσει ενδιαφέρουσες τεχνικές και αποτελεσματικούς αλγορίθμους (πολλοί από τους οποίους είναι αρκετά σύνθετοι), ώστε να είναι αδύνατον όλα τα διαθέσιμα ερευνητικά αποτελέσματα να καλυφθούν στο πλαίσιο ενός εξαμηνιαίου μαθήματος. Έτσι, έχουν παραλειφθεί σημαντικά θέματα, όπως, για παράδειγμα, ο έλεγχος και ο υπολογισμός της τομής γεωμετρικών αντικειμένων, ο υπολογισμός διαδρομών ελαχίστου μήκους ανάμεσα σε γεωμετρικά αντικείμενα, (στατικές και δυναμικές) γεωμετρικές δομές δεδομένων, randomized και παράλληλοι αλγόριθμοι για γεωμετρικά προβλήματα, αλγοριθμικές τεχνικές όπως fractional cascading, random sampling, parametric search και derandomization, η ευστάθεια γεωμετρικών υπολογισμών, κ.λπ. Ο ενδιαφερόμενος αναγνώστης παραπέμπεται στις βιβλιογραφικές πηγές που παρατίθενται στο τέλος των σημειώσεων.

Στο μεγαλύτερο μέρος τους, οι σημειώσεις αυτές αποτελούν ελεύθερη μετάφραση ενοτήτων του βιβλίου [1], ενώ έχουν επίσης συμπεριληφθεί και ενότητες από το βιβλίο [2].

Λεωνίδας Παληός

Ιωάννινα, Σεπτέμβριος 2005

Περιεχόμενα

Εισαγωγή

Η Υπολογιστική Γεωμετρία

Η Υπολογιστική Γεωμετρία προέκυψε τη δεκαετία του '70 από τον τομέα της σχεδίασης και της ανάλυσης αλγορίθμων και πλέον αποτελεί ανεξάρτητη ερευνητική περιοχή με τα δικά της περιοδικά, συνέδρια και πολυπληθή κοινότητα ερευνητών. Η επιτυχία της μπορεί να αποδοθεί τόσο στην ομορφιά των προβλημάτων που μελετά και των λύσεων που έχουν προταθεί όσο και στους πολλούς τομείς εφαρμογών (γραφικά, γεωγραφικά συστήματα πληροφοριών, ρομποτική, σχεδίαση με υπολογιστή, μοντελοποίηση μορίων, αναγνώριση προτύπων, και άλλους) στους οποίους οι γεωμετρικοί αλγόρθμοι παίζουν σημαντικό ρόλο. Αλλά ας δούμε κάποια παραδείγματα προβλημάτων με τα οποία ασχολείται η Υπολογιστική Γεωμετρία.

Φανταστείτε ότι περπατάτε στην Πανεπιστημιούπολη και δέχεστε μια κλήση στο κινητό σας το οποίο όμως κλείνει στα μισά της συνδιάλεξης καθώς τελειώνει η μπαταρία του. Χρειάζεστε να βρείτε ένα καρτοτηλέφωνο για να ολοκληρώσετε το τηλεφώνημά σας. Στην Πανεπιστημιούπολη υπάρχουν πολλά καρτοτηλέφωνα, αλλά θα θέλατε να πάτε σε αυτό που είναι πιο κοντά σας. Που βρίσκεται αυτό; Θα ξέρατε αμέσως, εάν είχατε έναν χάρτη στον οποίο για κάθε σημείο στην Πανεπιστημιούπολη θα σημειωνόταν το κοντινότερο καρτοτηλέφωνο. Το πιο πιθανό είναι ότι αυτός ο χάρτης θα παρουσίαζε την Πανεπιστημιούπολη χωρισμένη σε περιοχές σημειώνοντας, για καθεμία από αυτές, το καρτοτηλέφωνο το οποίο είναι πιο κοντά σε κάθε σημείο της περιοχής. Με τι θα έμοιαζαν αυτές οι περιοχές; Πως μπορούμε να τις υπολογίσουμε; Το παραπάνω παράδειγμα εισάγει αυτό που ονομάζεται *διάγραμμα Voronoi* και το οποίο μπορεί να χρησιμοποιηθεί για την επίλυση προβλημάτων που έχουν σχέση με οικονομική δραστηριότητα (μεγιστοποίηση πελατείας, ελαχιστοποίηση κόστους μεταφοράς), με πλοήγηση ρομπότ, ή ακόμη με την προσομοίωση της δημιουργίας κρυστάλλων.

Δεύτερο παράδειγμα. Υποθέστε ότι ξέρετε ποιο είναι το κοντινότερο καρτοτηλέφωνο. Εάν έχετε έναν χάρτη, δεν θα είναι δύσκολο να πάτε μέχρι το καρτοτηλέφωνο ακολουθώντας μια σχετικά σύντομη διαδρομή και αποφεύγοντας τοίχους ή άλλα εμπόδια. Όμως, ο προγραμματισμός ενός ρομπότ για να κάνει το ίδιο είναι μια πολύ πιο δύσκολη εργασία. Και σε αυτήν την περίπτωση, το πρόβλημα είναι καθαρά γεωμετρικό: δίδεται ένα σύνολο γεωμετρικών αντικειμένων-εμποδίων και ζητείται μια σύντομη διαδρομή μεταξύ δύο σημείων που αποφεύγει τα εμπόδια. Η επίλυση αυτού του προβλήματος, το οποίο ονομάζεται πρόβλημα *προγραμματισμού κίνησης*, είναι μεγάλης σημασίας στη ρομποτική.

Τρίτο παράδειγμα. Υποθέστε ότι δεν έχετε έναν χάρτη αλλά δύο: έναν με τα κτήρια αλλά και τα καρτοτηλέφωνα, και έναν με τους δρόμους της Πανεπιστημιούπολης. Για να προγραμματίσουμε την κίνησή μας προς το καρτοτηλέφωνο, χρειάζεται να *υπερθέσουμε* τους δύο χάρτες, δηλαδή, να συνδυάσουμε την πληροφορία που υπάρχει σε αυτούς. Η υπέρθεση χαρτών είναι μια από τις κύριες λειτουργίες των γεωγραφικών συστημάτων πληροφορίας (GIS)· περιλαμβάνει τον εντοπισμό των θέσεων των αντικειμένων του ενός χάρτη στον άλλο, τον υπολογισμό τομών διαφόρων αντικειμένων, κ.λπ.

Αυτά είναι τρία μονο παραδείγματα γεωμετρικών προβλημάτων το οποία απαιτούν προσεκτικά σχεδιασμένους αλγορίθμους για την επίλυσή τους. Η Υπολογιστική Γεωμετρία έχει ώς στόχο την επίλυση τέτοιων προβλημάτων· θα μπορούσε να οριστεί ως η συστηματική μελέτη αλγορίθμων και δομών δεδομένων για γεωμετρικά αντικείμενα με έμφαση σε ακριβείς και ασυμπτωτικά ταχείς αλγορίθμους. Η πορεία από τον ορισμό ενός προβλήματος έως την αποτελεσματική και κομψή επίλυσή του υπήρξε συχνά μακρά, περνώντας από πολλές ενδιάμεσες λύσεις που ήταν πολύπλοκες ή δεν ήταν βέλτιστες. Σήμερα, διαθέτουμε μια πλούσια συλλογή γεωμετρικών αλγορίθμων που είναι αποτελεσματικοί και σχετικά εύκολοι στην κατανόηση και την υλοποίησή τους.

Σχεδίαση Γεωμετρικών Αλγορίθμων

Η σχεδίαση γεωμετρικών αλγορίθμων συχνά περιλαμβάνει τις ακόλουθες τρεις φάσεις.

Στην πρώτη φάση, προσπαθούμε να ασχοληθούμε με το πρόβλημα που μας ενδιαφέρει στη γενική του μορφή αδιαφορώντας για τις ειδικές περιπτώσεις που ενδέχεται να προκύψουν, όπως, για παράδειγμα, συνευθειακά σημεία ή κατακόρυφα ευθύγραμμα τμήματα. Σε μια πρώτη προσπάθεια κατανόησης ή σχεδίασης ενός αλγορίθμου, είναι συχνά καλύτερο να αγνοήσουμε τέτοιες ειδικές περιπτώσεις.

Στη δεύτερη φάση, πρέπει να προσαρμόσουμε τον αλγόριθμο που προέκυψε από την πρώτη φάση ώστε να χειρίζεται σωστά τις ειδικές περιπτώσεις. Κάτι τέτοιο μπορεί να επιτευχθεί με εξαντλητικο έλεγχο περιπτώσεων. Ωστόσο, συχνά μπορούμε να συμπεριλάβουμε τις ειδικές περιπτώσεις στη γενική λύση του προβλήματος εκμεταλλευόμενοι τη γεωμετρία του προβλήματος. Για παράδειγμα, σε μια ταξινόμηση σημείων του επιπέδου ως προς τις x-συντεταγμένες τους, μπορούμε να χειριστούμε την ειδική περίπτωση σημείων με την ίδια x-συντεταγμένη εκτελώντας λεξικογραφική ταξινόμηση, δηλαδή, ταξινομώντας τα σημεία ως προς την x-συντεταγμένη και σε περίπτωση ισότητας ως προς την y-συντεταγμένη. Από την άλλη πλευρά, για πολλούς αλγορίθμους της Υπολογιστικής Γεωμετρίας, οι ερευνητές που τους πρότειναν συχνά παραλείπουν ειδικές περιπτώσεις, υποθέτοντας ότι τα δεδομένα είναι τέτοια ώστε αυτές δεν προκύπτουν. Από θεωρητική άποψη, κάτι τέτοιο συνήθως μπορεί να θεωρηθεί δικαιολογημένο: ο στόχος είναι να εκτιμηθεί η υπολογιστική πολυπλοκότητα του προβλήματος, ενώ μπορούμε σχεδόν πάντοτε να χειριστούμε τις ειδικές περιπτώσεις με κάποιο τρόπο, αν και η λεπτομερειακή ανάλυσή τους συχνά απαιτεί πολλή προσπάθεια. Επιπλέον, υπάρχουν γενικές τεχνικές, που ονομάζονται *σχήματα συμβολικών διαταραχών (symbolic perturbation schemes)*, οι οποίες μας επιτρέπουν να αδιαφορήσουμε για τις ειδικές περιπτώσεις κατά τη σχεδίαση και την υλοποίηση ενός αλγορίθμου ενώ εγγυώνται ότι ο αλγόριθμος θα χειριστεί σωστά και τις εκφυλλιμένες περιπτώσεις.

Η τρίτη φάση είναι η υλοποίηση του αλγορίθμου σε ένα πρόγραμμα. Σε αυτήν, πρέπει να δούμε ποιες βασικές πράξεις χρειαζόμαστε (π.χ., να μπορούμε να ελέγχουμε εάν ένα σημείο είναι στα αριστερά, στα δεξιά, ή επάνω σε μια κατευθυνόμενη ευθεία γραμμή) και πώς θα τις υλοποιήσουμε. Στην καλύτερη περίπτωση, μπορεί να έχουμε στη διάθεσή μας μια βιβλιοθήκη γεωμετρικού λογισμικού που περιέχει αυτές τις πράξεις, αλλιώς θα πρέπει να τις υλοποιήσουμε μόνοι μας. Ένα άλλο θέμα που προκύπτει κατά την υλοποίηση σχετίζεται με την πεπερασμένη ακρίβεια στην αναπαράσταση πραγματικών αριθμών και στην εκτέλεση πράξεων με αυτούς στον υπολογιστή. Η εξασφάλιση αριθμητικής ευστάθειας σε γεωμετρικούς υπολογισμούς δεν είναι εύκολη υπόθεση και συχνα απαιτεί πάρα πολλή προσπάθεια. Μια λύση είναι να χρησιμοποιήσουμε κάποιο από τα πακέτα ακριβούς αριθμητικής (exact arithmetic), τα οποία χρησιμοποιούν ακέραιους, ρητούς, ή και αλγεβρικούς αριθμούς, ανάλογα με το πρόβλημα, αλλά τότε η εκτέλεση του προγράμματος θα είναι αργή. Εναλλακτικά, μπορούμε να φροντίσουμε ώστε ο αλγόριθμος να ελέγχει συνθήκες οι οποίες θα εξασφαλίσουν την ομαλή λειτουργία του προγράμματος. Κάτι τέτοιο προϋποθέτει τον προσεκτικό ορισμό αυτών των συνθηκών καθώς και της συμπεριφοράς του προγράμματος κάθε φορά που κάποια από αυτές τις συνθήκες δεν πληρούται, ενώ συχνά συνεπάγεται ότι το αποτέλεσμα του προγράμματος είναι μόνο κατά προσέγγιση η λύση του προβλήματός μας. Τέλος, είναι δυνατόν να εκτιμήσει κάποιος, με βάση την είσοδο και τους υπολογισμούς, την ακρίβεια στην αναπαράσταση των αριθμών η οποία απαιτείται για την ορθή επίλυση του προβλήματος.

Αλγοριθμικό Πλαίσιο

Στην παράγραφο αυτή συνοψίζουμε βασικά στοιχεία του αλγοριθμικού πλαισίου της Υπολογιστικής Γεωμετρίας. Συγκεκριμένα, αναφερόμαστε στο μοντέλο υπολογισμού και στον τρόπο υπολογισμού και συμβολισμού της πολυπλοκότητας ενός αλγορίθμου.

Μοντέλο Υπολογισμού

Ως μοντέλο υπολογισμού για την εκτίμηση της πολυπλοκότητας των αλγορίθμων, χρησιμοποιούμε τη **μηχανή τυχαίας προσπέλασης** που χρησιμοποιεί **πραγματικούς αριθμούς** *(real RAM)* και για την οποία θεωρούμε οτι:

- ▷ αποθηκεύει οποιονδήποτε πραγματικό αριθμό με απόλυτη ακρίβεια και σε μοναδιαίο χώρο,
- ▷ έχει άμεση πρόσβαση σε κάθε στοιχείο της μνήμης σε μοναδιαίο χρόνο και
- ▷ εκτελεί τις 4 βασικές αριθμητικές πράξεις καθώς και συγκρίσεις μεταξύ πραγματικών με απόλυτη ακρίβεια και σε μοναδιαίο χρόνο.

Σε ειδικές περιπτώσεις και ανάλογα με τις απαιτήσεις του προβλήματος, ενδέχεται να θεωρηθεί ότι η μηχανή εκτελεί επίσης με απόλυτη ακρίβεια και σε μοναδιαίο χρόνο κάποιες ή και όλες τις εξής πράξεις: ύψωση σε δύναμη, υπολογισμός τετραγωνικής ρίζας, λογαρίθμου, και τις τριγωνομετρικές πράξεις. Για ακεραίους, ορισμένες φορές περιλαμβάνουμε στο υπολογιστικό μας μοντέλο τις πράξεις υπολογισμού ακέραιου μέρους και υπολογισμού του υπολοίπου κατά τη διαίρεση μεταξύ ακεραίων.

Πολυπλοκότητα Αλγορίθμων

Η ανάλυση των αλγορίθμων που θα παρουσιάσουμε βασίζεται στην εκτίμηση της **ασυμπτωτικής πολυπλοκότητάς** τους συνήθως στη **χείριστη περίπτωση**, που περιγράφεται ως μια συνάρτηση του μεγέθους των δεδομένων του προβλήματος. Ενδιαφερόμαστε για την **πολυπλοκότητα χρόνου** και την **πολυπλοκότητα χώρου** ενός αλγορίθμου: η πολυπλοκότητα χρόνου (χώρου, αντίστοιχα) ενός αλγορίθμου για ένα πρόβλημα μεγέθους n είναι ένα (ασυμπτωτικό) άνω φράγμα στον χρόνο (χώρο, αντίστοιχα) που απαιτεί ο αλγόριθμος όταν εκτελεστεί για οποιοδήποτε σύνολο δεδομένων μεγέθους n. Σε κάποιες περιπτώσεις, μελετάμε και την πολυπλοκότητα στη μέση περίπτωση, κυρίως όταν πρόκειται για αλγορίθμους που χρησιμοποιούν τυχαίους αριθμούς ή που υποθέτουν κάποια κατανομή στις τιμές των δεδομένων. Τέλος, ενδέχεται η πολυπλοκότητα ενός αλγορίθμου να εκφράζεται ως συνάρτηση και του μεγέθους της εξόδου· τότε, μιλάμε για *output-sensitive* αλγορίθμους.

Για την περιγραφή της πολυπλοκότητας αλγορίθμων χρησιμοποιούμε τους συμβολισμούς $O(\,)$, $\Omega(\,)$, $\Theta(\,)$, και $o(\,)$, οι οποίοι ορίζονται ως εξής:

$$
\begin{aligned}
f(n) = O(g(n)) &\iff \exists c \in \mathbb{R}, n_0 \in \mathbb{N} : \forall n \geq n_0, |f(n)| \leq c\,|g(n)| \\
f(n) = \Omega(g(n)) &\iff \exists c \in \mathbb{R}, n_0 \in \mathbb{N} : \forall n \geq n_0, |f(n)| \geq c\,|g(n)| \\
f(n) = \Theta(g(n)) &\iff \exists c_1, c_2 \in \mathbb{R}, n_0 \in \mathbb{N} : \forall n \geq n_0, c_1\,|g(n)| \leq |f(n)| \leq c_2\,|g(n)| \\
f(n) = o(g(n)) &\iff \forall c \in \mathbb{R}, \exists n_0 \in \mathbb{N} : \forall n \geq n_0, |f(n)| < c\,|g(n)|.
\end{aligned}
$$

Για τους συμβολισμούς $O(\,)$, $\Omega(\,)$, και $\Theta(\,)$, είναι εύκολο να αποδειχθεί ότι ισχύουν οι ακόλουθες ισοδυναμίες:

$$
\begin{aligned}
f(n) = \Omega(g(n)) &\iff g(n) = O(f(n)) \\
f(n) = \Theta(g(n)) &\iff f(n) = O(g(n)) \text{ και } g(n) = O(f(n))
\end{aligned}
$$

ενώ ένας εναλλακτικός (και ίσως πιο εύχρηστος) ορισμός για τον συμβολισμό $o(\,)$ είναι ο εξής:

$$f(n) = o(g(n)) \iff \lim_{n \to +\infty} \frac{f(n)}{g(n)} = 0.$$