Παραδείγματα αναγωγών

Θα δούμε τρία παραδείγματα αναγωγής προβλήματος A σε πρόβλημα B ($A \leq B$). Στο πρώτο παράδειγμα, η αναγωγή μας επιτρέπει να λύσουμε αποδοτικά το πρόβλημα A χρησιμοποιώντας αποδοτικό αλγόριθμο για το πρόβλημα B. Στο τρίτο παράδειγμα, η αναγωγή «μεταφέρει» το κάτω φράγμα που γνωρίζουμε για το πρόβλημα A στο πρόβλημα B. Στο δεύτερο παράδειγμα, παρόλο που το πρόβλημα B είναι ειδική περίπτωση του A (και άρα πιθανώς ευκολότερο), μαθαίνουμε ότι είναι ισοδύναμα.

Ύπαρξη τριγώνου σε γράφο και πολλαπλασιασμός πινάκων.

Δίνεται ο πίνακας γειτνίασης A απλού συνδεδεμένου γράφου $G = (V, E)$. Υπάρχει τριάδα κόμβων που είναι ανά δύο συνδεδεμένοι;

Ο τετριμμένος αλγόριθμος ελέγχει κάθε μία από τις $\binom{n}{3} = \Theta(n^3)$ τριάδες, όπου $n = |V|$. Αφού ο έλεγχος γίνεται σε χρόνο ανεξάρτητο του n, η χρονική πολυπλοκότητα είναι $O(n^3)$. Μπορούμε να εκμεταλλευτούμε μία ιδιότητα του πίνακα γειτνίασης A για να ανάγουμε το πρόβλημα σε αυτό του πολλαπλασιασμού πινάκων.

Πρόταση. *Το στοιχείο στη θέση (u, v) του A^k δίνει τον αριθμό των διαδρομών μήκους k από τον κόμβο u στον κόμβο v.*

Από το παραπάνω προκύπτει άμεσα πως μπορούμε να αποφανθούμε για την ύπαρξη τριγώνου υπολογίζοντας τον A^3. Μπορούμε όμως να τα καταφέρουμε και χρησιμοποιώντας μόνο έναν πολλαπλασιασμό. Είτε από την παραπάνω πρόταση, είτε από τον ορισμό του πολλαπλασιασμού πινάκων ($A^2(i,j) = \sum_{k=1}^{n} A(i,k)A(k,j)$), παρατηρούμε ότι $A^2(i,j) > 0 \iff (\exists k \notin \{i,j\})(A(i,k) = A(k,j) = 1)$. Με άλλα λόγια, $A^2(u,v) > 0$ αν και μόνο αν υπάρχει $w \in V \setminus \{u,v\}$ και υπάρχουν οι ακμές (u,w) και (w,v). Προκύπτει ότι

> «Υπάρχει τρίγωνο στον G αν και μόνο αν υπάρχουν i, j τέτοια που $A(i,j) = 1$ και $A(i,j) > 0$.»

Ο σχετικός αλγόριθμος χρειάζεται $O(n^2)$ χρόνο για τον έλεγχο όλων των ζευγών (i,j) και χρόνο για να υπολογίσει τον A^2. Χρησιμοποιώντας των αλγόριθμο του Strassen για τον πολλαπλασιασμό πινάκων η συνολική χρονική πολυπλοκότητα είναι $O(n^{\log_2 7})$. Αυτή την στιγμή γνωρίζουμε ότι ο πολλαπλασιασμός $n \times n$ πινάκων μπορεί να γίνει σε χρόνο $O(n^{2.3728639})$.

Σχόλιο. Η χρονική πολυπλοκότητα του πολλαπλασιασμού πινάκων είναι ένα σημαντικό ανοιχτό πρόβλημα από το 1968 μέχρι σήμερα. Όπως αναφέρθηκε, γνωρίζουμε ότι η χρονική του πολυπλοκότητα φράσσεται άνω από $O(n^{2.3728639})$. Από την άλλη μεριά, το καλύτερο (γενικό) κάτω φράγμα που γνωρίζουμε είναι το τετριμμένο $\Omega(n^2)$. Το τελευταίο προκύπτει από την παρατήρηση ότι αν ένας αλγόριθμος τερματίζει σε λιγότερο από $2n^2$ βήματα, τότε δεν έχει διαβάσει τουλάχιστον ένα στοιχείο. Δεν είναι δύσκολο να δειχθεί ότι σε αυτή την περίπτωση υπάρχουν πίνακες των οποίων το γινόμενο δεν υπολογίζεται σωστά.

Πολλαπλασιασμός πινάκων και τετραγωνισμός πίνακα.

Στο προηγούμενο πρόβλημα η αναγωγή είναι ουσιαστικά στο πρόβλημα τετραγωνισμού πίνακα. Είναι εύλογο το ερώτημα: μπορούμε να τετραγωνίσουμε πίνακες γρηγορότερα (ασυμπτοτικά) από ότι να πολλαπλασιάσουμε πίνακες; Θα δούμε ότι η απάντηση είναι αρνητική.

Η ιδέα βρίσκεται στην παρακάτω έκφραση.

$$\begin{bmatrix} 0 & A \\ B & 0 \end{bmatrix}^2 = \begin{bmatrix} AB & 0 \\ 0 & BA \end{bmatrix}$$

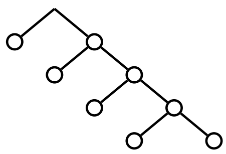
Για να υπολογίσουμε το γινόμενο AB δύο οποιονδήποτε $n \times n$ πινάκων, αρκεί να τετραγωνίσουμε τον παραπάνω $2n \times 2n$ πίνακα. Αν μπορούμε να κάνουμε τον τετραγωνισμό σε χρόνο $T(2n)$, τότε πολλαπλασιάζουμε σε χρόνο $O(T(2n) + n^2)$. (Προσθέσαμε τον χρόνο για την κατασκευή του πίνακα.) Υποθέτοντας ότι $T(2n) = O(T(n))$ (π.χ. αν η $T(n)$ είναι πολυώνυμο), δείξαμε το παρακάτω.

Πρόταση. *Αν υπάρχει αλγόριθμος που υπολογίζει το τετράγωνο $n \times n$ πίνακα σε χρόνο $O(n^\omega)$, τότε υπάρχει αλγόριθμος που υπολογίζει το γινόμενο δύο $n \times n$ πινάκων σε χρόνο $O(n^\omega)$.*

Ταξινόμηση αριθμών και κωδικοποίηση Huffman.

Θυμηθείτε την κωδικοποίηση Huffman που μας επιτρέπει συμπίεση δεδομένων. Δίνονται συχνότητες $f_1, f_2, \ldots, f_n$ εμφάνισης n συμβόλων και αντιστοιχίζεται σε κάθε σύμβολο μία δυαδική ακολουθία κατασκευάζοντας ένα δένδρο (δένδρο Huffman).

Έστω ότι έχουμε να ταξινομήσουμε n θετικούς ακεραίους $x_1, x_2, \ldots, x_n$ που είναι διαφορετικοί ανά δύο. Αν έχουμε έναν αλγόριθμο που κατασκευάζει το δένδρο Huffman, μπορούμε να τον χρησιμοποιήσουμε ως εξής. Έστω $y_i = 2^{x_i}$ και $y = \sum_{i=1}^{n} y_i$. Κατασκευάζουμε το δένδρο που αντιστοιχεί στις συχνότητες $f_i = y_i/y$. Επειδή, για κάθε $k = 1, 2, \ldots$, $2^k > \sum_{i=0}^{k-1} 2^i$, το δένδρο έχει την παρακάτω μορφή.



Έχοντας λοιπόν το δένδρο Huffman μπορούμε να διαβάσουμε τα στοιχεία ταξινομημένα (π.χ. το ελάχιστο στοιχείο θα αντιστοιχεί στην ελάχιστη συχνότητα και θα βρίσκεται μακρύτερα από τη ρίζα).

Παρατηρείστε ότι η αναγωγή δεν δουλεύει για οποιουσδήποτε αριθμούς, αλλά μόνο για διαφορετικούς ανά δύο. Αν ο στόχος μας ήταν να λύσουμε το πρόβλημα ταξινόμησης, αυτό θα ήταν πρόβλημα. Όμως στη συγκεκριμένη περίπτωση μας ενδιαφέρει η παρακάτω πρόταση.

Πρόταση. *Κάθε αλγόριθμος που κατασκευάζει το δένδρο Huffman n συμβόλων χρησιμοποιώντας μόνο συγκρίσεις, χρειάζεται $\Omega(n \log n)$ συγκρίσεις στην χειρότερη περίπτωση.*

Το συμπέρασμα είναι σωστό γιατί το κάτω φράγμα για το πρόβλημα της ταξινόμησης ισχύει ακριβώς στην περίπτωση που τα στοιχεία είναι διαφορετικά ανά δύο.