1η Σειρά Ασκήσεων

1. Το τετράγωνο ενός πίνακα A είναι ο πίνακας $A^2 = A \cdot A$.

   (a) Δείξτε ότι πέντε πολλαπλασιασμοί αρκούν για τον υπολογισμό του τετραγώνου ενός πίνακα 2×2.

   Παρατηρείστε ότι, $\begin{bmatrix} a & b \\ c & d \end{bmatrix}^2 = \begin{bmatrix} a^2+bc & ab+bd \\ ca+dc & d^2+cb \end{bmatrix} = \begin{bmatrix} a^2+bc & b(a+d) \\ c(a+d) & d^2+cb \end{bmatrix}$

   Έτσι απαιτούνται 5 πολλαπλασιασμοί: a^2, bc, $b(a+d)$, $c(a+d)$ και d^2.

   (b) Θα δείξουμε τώρα ότι ένας τετραγωνισμός πινάκων δεν είναι ευκολότερος από τον πολλαπλασιασμό πινάκων. Συγκεκριμένα, θα δείξουμε ότι αν ο πολλαπλασιασμός $n \times n$ πινάκων μπορεί να γίνει σε χρόνο $S(n)+O(n^c)$, τότε ο πολλαπλασιασμός δύο $n \times n$ πινάκων μπορεί επίσης να γίνει σε χρόνο $O(n^c)$.

   i. Με δεδομένους δύο $n \times n$ πίνακες A και B, δείξτε ότι ο πίνακας $A \cdot B + B \cdot A$ μπορεί να υπολογιστεί σε χρόνο $3S(n) + O(n^2)$.

   Για να υπολογίσουμε το $AB + BA$, μπορούμε να χρησιμοποιήσουμε την ταυτότητα $AB+BA = (A+B)^2 - A^2 - B^2$. Η ταυτότητα μπορεί να υπολογιστεί χρησιμοποιώντας 3 τετραγωνισμούς, μια πρόσθεση και δύο αφαιρέσεις σε $n \times n$ πίνακες. Ο χρόνος για πρόσθεση και αφαίρεση πινάκων είναι $O(n^2)$, οπότε ο υπολογισμός $A \cdot B + B \cdot A$ μπορεί να γίνει σε χρόνο $3S(n) + O(n^2)$.

   ii. Με δεδομένους δύο $n \times n$ πίνακες X και Y, ορίζουμε τους $2n \times 2n$ πίνακες A και B, όπου $A = \begin{pmatrix} x & 0 \\ 0 & 0 \end{pmatrix}$ και $B = \begin{pmatrix} 0 & Y \\ 0 & 0 \end{pmatrix}$. Προσδιορίστε μια κατάλληλη έκφραση για τον πίνακα $A \cdot B + B \cdot A$. δείξτε ότι ο πίνακας $A \cdot B + B \cdot A$ μπορεί να υπολογιστεί σε χρόνο $3S(n) + O(n^2)$.

   $AB + BA = \begin{bmatrix} 0 & XY \\ 0 & 0 \end{bmatrix}$.

   iii. Χρησιμοποιείστε τα ερωτήματα (i) και (ii) για να επιχειρηματολογήσετε ότι το γινόμενο $X \cdot Y$ μπορεί να υπολογιστεί σε χρόνο $3S(2n) + O(n^2)$. Συμπεράνετε ότι ο υπολογισμός $n \times n$ πινάκων μπορεί να γίνει σε χρόνο $O(n^c)$.

   Δεδομένων δύο $n \times n$ πινάκων X και Y, κατασκευασε $2n \times 2n$ πίνακες A και B όπως στην (ii) και χρησιμοποίησε τον αλγόριθμο από την (i) για να υπολογίσεις $C = AB + BA$. Ο πίνακας XY μπορεί να διαβαστεί από το πάνω δεξιά τεταρτημόριο του C. Ο χρόνος που απαιτείται για αυτόν τον αλγόριθμο φράσεται από πάνω από τον χρόνο για να γίνει το (i), που είναι $3S(2n) + O((2n)^2) + O(n^c + n^2)$. Αφού $c \geq 2$, ο χρόνος που απαιτείται είναι $O(n^c)$.

2. Μας δίνονται n ακέραιοι $a_1, \cdots, a_n$. Μας ζητείται να βρούμε τους συντελεστές $c_0, \cdots, c_{n-1}$ του πολυωνύμου $(x+a_1) \cdot (x+a_2) \cdots (x+a_n) = x^n + c_{n-1}x^{n-1} + \cdots + c_0$. Παρουσιάστε ένα

αλγόριθμο του τύπου διαίρει-και-βασίλευε για το πρόβλημα αυτό με χρονική πολυπλοκότητα $O(n \lg^2 n)$.

Μπορούμε να υποθέσουμε ότι το n είναι δύναμη του 2. (Εάν δεν είναι δύναμη του δύο, μπορούμε να επεκτήνουμε την είσοδο $a_1, \cdots a_n$ με μηδενικά ώς τον πιο κοντινό αριθμό που είναι δύναμη του δυο, να τρέξουμε τον αλγόριθμο, και να επιστρέφουμε τους πρώτους n συντελεστές του πολυώνυμου που προκύπτει.

Για να υπολογίσουμε τους συντελεστές του πολυώνυμου $(x+a_1)\cdots(x+a_n)$, μπορούμε να υπολογίσουμε αναδρομικά τους συντελεστές του $p(x) = (x+a_1)\cdots(x+a_{n/2})$ και τους συντελεστές του $q(x) = (x+a_{n/2+1})\cdots(x+a_n)$ και κατόπιν του γινόμενου $p(x)q(x)$ χρησιμοποιώντας τον FFT αλγόριθμο με πολυωνυμικό πολλαπλασιασμό. Αυτή η παρατήρηση οδηγεί στον παρακάτω αλγόριθμο:

Algorithm coefficients$(a_1, \cdots, a_n)$

if $n = 1$ return $(1, a_1)$

$g_1, \cdots g_{n/2}) := coefficients(a_1, \cdots, a_{n/2)}$

$h_1, \cdots h_{n/2}) := coefficients(a_{n/2+1}, \cdots, a_{n)}$

return $convolve((g_1, \cdots g_{n/2}), (h_1, \cdots h_{n/2}))$

Εδώ, ο αλγόριθμος $convolve(p, q)$ υπολογίζει την συνέλιξη των ακολουθιών p, q. (Εάν τα p, q θεωρούνται ως πολυώνυμα βαθμού n τα οποία αναπαριστούνται από τους συντελεστές τους, τότε η $convolve(p, q)$ είναι το γινόμενο τους.) Από τις διαλέξεις γνωρίζουμε ότι αυτός ο αλγόριθμος απαιτεί χρόνο $O(nlogn)$ πράξεων όταν τα p, q είναι ακολουθίες αριθμών μεγέθους n.

Το πλήθος των πράξεων που εκτελείται για την $coefficients$ ικανοποιεί την αναδρομή $T(n) = 2T(n/2) + O(n \log n)$ (με βάση $T(1) = O(1)$). Αυτή ισούται με $T(n) = O(n \log^2 n)$.

3. Μας δίνονται οι ακέραιοι n, k και οι πιθανότητες $p_1, \cdots, p_n$. Για κάθε i, $1 \leq i \leq n$, p_i είναι η πιθανότητα η ρίψη i ενός νομίσματος να έχει αποτέλεσμα την "κορώνα". Θέλουμε να υπολογίσουμε την πιθανότητα να έχουμε σαν αποτέλεσμα την "κορώνα" k φορές σε n ρίψεις του νομίσματος. Δώστε ένα αλγόριθμο με χρονική πολυπλοκότητα (αριθμό προσθέσεων και πολλαπλασιασμών) $O(n^2)$ για τον υπολογισμό της πιθανότητας.

   Ένας τρόπος να λυθεί το πρόβλημα είναι με δυναμικό προγραμματισμό. Ορισε $L[i, j]$ να είναι η πιθανότητα να πάρουμε ακριβώς j κορώνες στα i πρώτες ρίψεις. Η βασική περίπτωση είναι $L[0, 0] = 1$ και $L[i, j] = 0$ για $j < 0$. Ενημερώνουμε τον πίνακα L διαδοχικά (αύξηση) χρησιμοποιώντας την αναδρομή (με την σειρά $i = 1, 2, \cdots, n$ με εσωτερικό loop $j = 0, 1, \cdots, i$):

$$L[i, j] = p_i L[i-1, j-1] + (1 - p_i) L[i-1, j], \quad j = 0, \cdots, i$$

   Η τελική απάντηση δίνεται από το $L[n, k]$. Για να υπολογίσουμε το $L[i, \cdot]$ από το $L[i-1, \cdot]$ απαιτεί $O(n)$ χρόνο, ο αλγόριθμος απαιτεί $O(n^2)$ χρόνο.

4. Έστω $D = \{d_1, \cdots, d_n\}$ ένα σύνολο από νομισματικές αξίες, όπου $d_i \geq 1$, και ακέραιος. Έχουμε ένα μη φραγμένο σύνολο νομισμάτων από κάθε αξία. Κάποιος μας ζητά να δώσουμε νομίσματα για το ποσό των A Euro: Το ποσό αυτό θα προέλθει από τα νομίσματα που έχουμε στη διάθεση μας. Αν μπορούμε να δώσουμε το ποσό αυτό, λέμε ότι το A είναι αναπαραστήσιμο στο D.

(a) Δώστε ένα αλγόριθμο δυναμικού προγραμματισμού ο οποίος, όταν το A είναι αναπαραστήσιμο στο D, βρίσκει την αναπαράσταση του με τον ελάχιστο αριθμό νομισμάτων.

Έστω $M(i)$ ο ελάχιστος αριθμός νομισμάτων που απαιτούνται για την αναπαράσταση του i. Θέλουμε να υπολογίσουμε το $M(A)$. Εάν το i είναι αναπαραστήσιμο και μια νομισματική αξία d_j μπορεί να χρησιμοποιηθεί σε μια ελάχιστη αναπαράσταση του i, τότε $M(i)$ είναι απλώς $M(i - d_j) + 1$. Τυπικά,

$$M(i) = \min_{1 \leq j \leq n} \{M(i - d_j) + 1\}$$

εαν το ελάχιστο υπάρχει και $M(i) = \perp$ (μη αναπαραστήσιμο) διαφορετικά. Οι βασικές περιπτώσεις είναι $M(0) = 0$ και $M(i) = \perp$ εάν $i < 0$. Το μέγεθος του πίνακα είναι γραμμικό στο A και για να γεμίσουμε τον πίνακα χρειαζόμαστε $O(n)$ χρόνο για να ελέγξουμε τις παλαιότερες τιμές. Οπότε ο χρόνος που απαιτείται είναι σίγουρα $O(nA)$.

(b) Δώστε ένα αλγόριθμο δυναμικού προγραμματισμού ο οποίος, με δεδομένα τα A και D, βρίσκει τον αριθμό των αναπαραστάσεων του A στο D.

Έστω $M(i)$ το πλήθος των τρόπων για αναπαραστήσουμε το j χρησιμοποιώντας μόνο τις νομισματικές αξίες $d_1, \cdots, d_i$. Ενδιαφερομαστε να υπολογίσουμε το $M(n, A)$. Μπορούμε να θεωρήσουμε δύο αναπαραστάσεις του j χρησιμοποιώντας το $d_1, \cdots, d_i$: αυτές που χρησιμοποιούν τουλάχιστον ένα νόμισμα αξίας d_i και αυτές που δεν χρησιμοποιούν κανένα νόμισμα αξίας d_i. Αφού αυτά τα σύνολα αναπαράστασης είναι ξένα μεταξύ τους και καλύπτουν όλες τις πςριτπώσεις αναπαράστασης του j χρησιμοποιώντας τα $d_1, \cdots, d_i$, έχουμε τον παρακάτω αναδρομικό ορισμό για το $M(i, j)$:

$$M(i, j) = M(i, j - d_i) + M(i - 1, j))$$

Οι βασικές περιπτώσεις είναι $M(i, 0) = 1$ για όλα τα $i \geq 0$ (υπάρχει μόνο ένας τρόπος να αναπαρασταθει το 0) και $M(0, j) = 0$ για όλα τα $j > 0$ (δεν υπάρχει τρόπος να αναπαραστήσουμε το j χωρίς νομίσματα). Το μέγεθος του πίνακα είναι nA και για να γεμίσουμε μια θέση του χρειάζεται να κάνουμε σταθερό αριθμό από ελέγχους. Οπότε ο χρόνος που απαιτείται είναι $O(nA)$.

Παράδοση: 29 Φεβρουαρίου.