Υπογραμμικοί Αλγόριθμοι

Μάθημα 9 - 26/11/2019

Επιμέλεια Σημειώσεων: Αργύρης Μουζάκης

Πολλαπλασιασμός Πολυωνύμων

1 Αρχικές Έννοιες

Έστω τα πολυώνυμα:

$$p(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$$

$$q(x) = b_0 + b_1 x + \cdots + b_{n-1} x^{n-1}$$

βαθμού το πολύ $n-1$.
Σε άλλο μάθημα είδαμε πως το $p(x)\,q(x)$ υπολογίζεται σε χρόνο $\mathcal{O}(n \log(n))$ χρησιμοποιώντας τον αλγόριθμο FFT. Υπενθυμίζεται πως το γινόμενο θα είναι ένα πολυώνυμο βαθμού το πολύ $2n-2$ της μορφής:

$$\gamma_0 + \gamma_1 x + \cdots + \gamma_{2n-2} x^{2n-2}$$

με:

$$\begin{aligned}
\gamma_0 &= a_0 b_0 \\
\gamma_1 &= a_1 b_0 + a_0 b_1 \\
\gamma_2 &= a_2 b_0 + a_1 b_1 + a_0 b_2 \\
&\vdots \\
\gamma_i &= \sum_{\substack{j+j'=i \\ 0 \leqslant j, j' \leqslant n-1}} a_j b_{j'}, \forall i
\end{aligned}$$

Αναπαριστώντας τα παραπάνω πολυώνυμα ως διανύσματα του $\mathbb{Z}^{2n-1}$, έχουμε:

$$a = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, b = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \gamma = \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_{n-1} \\ \gamma_n \\ \vdots \\ \gamma_{2n-2} \end{bmatrix}$$

Έτσι, είναι απλά $\gamma = a \star b$, όπου με $\star$ συμβολίζουμε τη συνέλιξη μεταξύ διανυσμάτων. Στόχος μας είναι να βρούμε καλύτερους αλγορίθμους πολλαπλασιασμού πολυωνύμων από τον FFT, ή ισοδύναμα να βρούμε καλύτερους αλγόριθμους για τη συνέλιξη δύο διανυσμάτων.

Κυκλική Συνέλιξη

Το πρώτο βήμα για να μπορέσουμε να κατασκευάσουμε τέτοιους αλγορίθμους είναι να ορίσουμε τι είναι *κυκλική συνέλιξη*. Πρόκειται για μία γενίκευση της απλής συνέλιξης όπου, θεωρώντας $u, v \in \mathbb{Z}^n$, έχουμε:

$$(u \star v)_i = \sum_{\substack{(j+j') \mod n = i \\ 0 \leqslant j, j' \leqslant n-1}} u_j v_{j'}$$

Ως προς τον μετασχηματισμό Φουριέ, η πράξη αυτή είναι δυϊκή του κατά σημείου πολλαπλασιασμού διανυσμάτων. Στο εξής, όταν χρησιμοποιούμε το σύμβολο $\star$, θα εννοούμε κυκλική συνέλιξη. Εν γένει, ο υπολογισμός αυτής της ποσότητας απαιτεί επίσης $\mathcal{O}(n \log(n))$ χρόνο. Τι συμβαίνει, όμως, αν τα u, v είναι *αραιά*, δηλαδή έχουν λίγους μη μηδενικούς συντελεστές (τα $||u||_0, ||v||_0$[1] είναι μικρά); Είναι σαφές πως ο χρόνος που απαιτείται είναι $\mathcal{O}(||u||_0 ||v||_0)$. Πράγματι, αν πάρουμε πχ:

$$\left(1 + x^{100000}\right)\left(1 + x^2 + x^3 + x^4 + x^{100001}\right)$$

μπορούμε να εντοπίσουμε γρήγορα ποιοι είναι οι μη μηδενικοί συντελεστές εκτελώντας τον πολλαπλασιασμό με τον συμβατικό τρόπο.
Από την άλλη, τι θα συνέβαινε αν είχαμε και αρνητικούς συντελεστές; Σε μία περίπτωση όπως πχ αυτή του:

$$x^n - 1 = (x - 1)\left(1 + x + \ldots x^{n-1}\right)$$

[1] Υπενθυμίζεται πως $||u||_0 = \#$ μη μηδενικών συνιστωσών του u.

οι μη μηδενικοί συντελεστές της εισόδου είναι αρκετοί, αλλά απαλείφονται όροι στο γινόμενο με αποτέλεσμα η έξοδος να έχει ελάχιστους μη μηδενικούς όρους.
Από τα προηγούμενα, καταλαβαίνουμε πως το ιδανικό θα ήταν ένας αλγόριθμος να απαιτεί χρόνο $\mathcal{O}\left(||u \star v||_0 + (||u||_0 + ||u||_0)\right)$, όπου ο μεν 1ος όρος είναι το μέγεθος της εξόδου, ο δε 2ος αυτό της εισόδου.
Συνοψίζοντας όσα έχουμε πει μέχρι τώρα, έχουμε ότι, αν θέλαμε να υπολογίσουμε ένα γινόμενο όπως το:

$$(1 + x + \cdots + x^n)^2$$

έχουμε τρεις περιπτώσεις μέχρι στιγμής.

- **Τετριμμένος:** κάνε όλα τα αναπτύγματα $\rightarrow \mathcal{O}\left(n^2\right)$
- **FFT** $\rightarrow \mathcal{O}\left(n \log\left(n\right)\right)$
- **Ιδανικός (και κατά πάσα πιθανότητα ανύπαρκτος):** όλοι οι όροι μη μηδενικοί και οι δυνάμεις πάνε μέχρι $x^{2n} \rightarrow \mathcal{O}\left(n\right)$

Ιστορική Αναδρομή

Εδώ αναφέρουμε τις κυριότερες προσπάθειες που έχουν γίνει για την επιτάχυνση του πολλαπλασιαμού πολυωνύμων.

- Las Vegas πιθανοτικός αλγόριθμος που τρέχει σε $\mathcal{O}\left(k \log^2\left(n\right)\right)$ για $u_i, v_i \geqslant 0, \forall i \in [n]$ μέσω τεχνικών διαχείρισης συμβολοσειρών [Cole, Hariharan, STOC 2002].
- Ντετερμινιστικός αλγόριθμος [Amir, Porat, 2010] για μια ειδική περίπτωση, όπου το ενα από τα δύο διανύσματα κρατιέται σταθερό και το άλλο ανανεώνεται. Στην αρχή ο αλγόριθμος παίρνει αρκετό χρόνο, $O(\|u\|_0, \|v\|_0)$, αλλάμε μετά το γινόμενο υπολογίζεται γρήγορα.
- Ντετερμινιστικός αλγόριθμος σε $\left(k \log\left(n\right)\right)^{1+o(1)}$ όταν δίνεται ένα υποσύνολο T μεγέθους $\mathcal{O}\left(k\right)$ που περιέχει όλα τα i όπου $\left(u \star v\right)_i \neq 0$ [Chan, Lewenshtein, STOC 2015].
- Πιθανοτικός αλγόριθμος σε $\mathcal{O}\left(k \log^4\left(n\right)\right)$ για γενικά u, v [Νάκος, 2019].
- Ντετερμινιστικός για $u_i, v_i \geqslant 0$ σε χρόνο $k 2^{\mathcal{O}\left(\sqrt{\log(k)\log(\log(n))}\right)} poly\left(\log\left(n\right)\right)$ όπου $k = ||u + v||_0$ [Bringmann, Νάκος, 2019].

Σε αυτή τη διάλεξη θα δούμε τον πιθανοτικό αλγόριθμο από το 2019.

2 Πιθανοτικός Αλγόριθμος 2019

Σε ό,τι ακολουθεί ορίζουμε $[n] = \{0, 1, \ldots, n-1\}$. Θεωρούμε ότι οι δείκτες όλων των διανυσμάτων ξεκινάνε από το 0. Έστω ότι έχουμε ακριβώς (ή σε έναν $O(1)$ πολλαπλαστικό παράγοντα) το $||u \star v||_0$. Θα θέλαμε να βρούμε έναν τρόπο να κατακερματίσουμε τα στοιχεία των u, v, ο οποίος να μας επιτρέπει να υπολογίσουμε το $u \star v$, εκμεταλλευόμενοι το ότι ξέρουμε περίπου το πλήθος των μη μηδενικών στοιχείων στο αποτέλεσμα. Για το λόγο αυτό, θέλουμε ένα μετασχηματισμό $\mathbb{Z}^n \ni u \mapsto u' \in \mathbb{Z}^B$, τον οποίο θα τον λέμε αναδίπλωση.

$$u'_i = \sum_{\substack{j \in [n] \\ j \mod B = i}} u_j$$

$$v'_i = \sum_{\substack{j \in [n] \\ j \mod B = i}} v_j$$

Μπορούμε να υπολογίσουμε το $u' \star v'$ σε χρόνο $\mathcal{O}(B \log)$ με FFT. Στόχος μας είναι από αυτό να ανακτήσουμε το $u \star v$. Προκειμένου, αυτό να είναι εφικτό, αρχικά θα μελετήσουμε τις ιδιότητές της συνέλιξης των αναδιπλωμένων διανυσμάτων.

Πρόταση 1. *Για κάθε $\ell \in [B]$ Ισχύει ότι:*

$$(u' \star v')_l = \sum_{\substack{i \in [n] \\ i \mod B = l}} (u \star v)_i = \left((u \star v)'\right)_i$$

Απόδειξη. Έστω πως έχουμε (j, j') με $j + j' = i$. Γνωρίζουμε πως ο όρος $u_j v_{j'}$ συνεισφέρει στο $(u \star v)_i$. Τι συμβαίνει όμως με το $u' \star v'$; Έχουμε πως τα $u_j, v_{j'}$ συνεισφέρουν στους όρους $u'_{j \mod B}$ και $v'_{j' \mod B}$, αντίστοιχα. Ακόμη, Υπολογίζοντας τη συνέλιξη $u' \star v'$, το γινόμενο $u_j v_{j'}$ εμφανίζεται στον όρο με δείκτη:

$$(j \mod B + j' \mod B) \mod B = (j + j') \mod B = i \mod B$$

Άρα, είτε υπολογίσουμε το $u \star v$ και κατακερματίσουμε εκ των υστέρων, είτε κατακερματίσουμε πρώτα τα u, v και υπολογίσουμε το $u' \star v'$, το γινόμενο $u_j v_{j'}$ θα συνεισφέρει πάντα στη συνιστώσα με δείκτη $i \mod B$. Το ζητούμενο έπεται. ■

Παρόμοιες ιδέες υπάρχουν στα papers των Chan-Lewenshtein και Cole-Hariharan. Η ουσία είναι πως, βασιζόμενοι στο ότι η απεικόνιση modulo είναι ομομορφική ως προς την πράξη της πρόσθεσης ακεραίων, έχουμε δείξει πως η αναδίπλωση $u \mapsto u'$ είναι ομομορφική ως προς την πράξη της κυκλικής συνέλιξης διανυσμάτων, γεγονός

που μας επιτρέπει πρώτα να αναδιπλώσουμε και μετά να υπολογίσουμε την κυκλική συνέλιξη, το οποίο είναι σημαντικό πιο γρήγορο από το να πάρουμε την κυκλιδκή συνέλιξη πρώτα και μετά να αναδιπλώσουμε.
Πάμε τώρα την προηγούμενη ιδέα λίγο παραπέρα.

Πόρισμα 1. *Έστω w ένας αριθμός. Αν ορίσουμε την απεικόνιση $u \mapsto u'$ με:*

$$u'_i = \sum_{\substack{j \in [n] \\ j \mod B = i}} u_j w^j$$

Ισχύει:

$$(u' \star v')_l = \sum_{\substack{i \in [n] \\ i \mod B = l}} (u \star v)_i \, w^i$$

Απόδειξη. Έστω (j, j') με $j + j' = i$. Όπως πριν, το γινόμενο $u_j v_{j'}$ θα εμφανίζεται στο $(u \star v)_{i \mod B}$. Επειδή τώρα έχουμε και τους παράγοντες $w^j, w^{j'}$, είναι:

$$\left(u_j w^j\right)\left(v_{j'} w^{j'}\right) = u_j v_{j'} w^{j+j'} = u_j v_{j'} w^i$$

Το ζητούμενο έπεται.

■

Ο σκοπός που εξυπηρετεί αυτός ο όρος είναι ο εξής: στην ιδανική περίπτωση που έχουμε τέλειο κατακερματισμό (δηλ. κάθε $(u' \star v')_i \neq 0$ είναι μόνο του ή έχει collision με στοιχεία που είναι μηδενικά), θα έχουμε $(u' \star v')_l = (u \star v)_i \cdot w^i$, για κάποιο $i \mod B = l$. Ο στόχος είναι από αυτό να μάθουμε τα $(u \star v)_i$ και i. Αν μάθουμε το 1ο, το 2ο είναι ένα πρόβλημα διακριτού λογαρίθμου (αφού εμείς έχουμε επιλέξει το w). Η επιλογή του w χρειάζεται προσοχή, διότι αν πάρουμε το $w \neq 1$ να είναι οποιοσδήποτε πραγματικός αριθμός ο w^j δυνάται να έχει μέχρι $\Theta(n)$ ψηφία. Για αυτό το λόγο θέτουμε $w = e^{\sqrt{-1}\frac{2\pi}{n}}$ που έχει μοναδιαίο μέτρο, στρογγυλοποιώντας το αποτέλεσμα ανάλογα με το μήκος της αναπαράστασής μας. Έτσι, άμα έχουμε το $(u' \star v')_l$, μπορούμε με τριαδική αναζήτηση να βρούμε τη φάση και να ανακτήσουμε το i, οπότε ο υπολογισμός της ζητούμενης συνιστώσας μετά είναι εύκολος.
Μένει τώρα να εξετάσουμε υπό ποιες προϋποθέσεις το σχήμα που ορίσαμε μπορεί να δουλέψει. Για να είναι το σχήμα "καλό", θα θέλαμε όλα τα μη μηδενικά στοιχεία να είναι "καλά" (δηλ. να μην έχουν συγκρούσεις με άλλα μη μηδενικά στοιχεία). Αυτή είναι πολύ ισχυρή απαίτηση και δύσκολη να επιτευχθεί χωρίς μεγάλο να εκτοξευτεί ο χρόνος. Για την ακρίβεια και σαν γενικό κανόνα, άμα θέλω να κατακερματίστω τέλεια k στοιχεία χρειάζομαι $\Theta(k^2)$ ομάδες, ενώ άμα θέλω να κατακερματίστω τα περισσότερα από αυτά χρειάζομαι $\Theta(k)$ ομάδες, το οποίο αποτελεί τετραγωνική διαφορά. Σε αυτό το σημείο θυμίζουμε το αποτέλεσμα από την Άσκηση 4 της 2ης σειράς:

Πρόταση 2. *Αρκεί να βρούμε* $B = \mathcal{O}\left(||u \star v||_0\right)$ *τέτοιο ώστε τα* $\frac{9}{10}$ *από τα* $0 \leqslant i \leqslant B-1$ *όπου κατακερματίζονται τα μη μηδενικά στοιχεία του* $u \star v$ *να μην έχουν collisions.*

Άρα, χαλαρώνουμε τις απαιτήσεις για το "καλό" σχήμα, αρκούμενοι τα $\frac{9}{10}$ των μη μηδενικών στοιχείων να είναι "καλά". Για να το πετύχουμε αυτό, η διαδικασία είναι η εξής:

1. Πάρε έναν τυχαίο πρώτο B στο $\left[c\tau \log^2(n)\right]$, όπου $\tau = ||u \star v||_0$ και c σταθερά > 1.

2. Υπάρχουν $\geqslant \tau \log(n)$ πρώτοι στο παραπάνω διάστημα.

3. Έχουμε:

$$\mathop{\mathbb{P}}_{B \sim \left[c\tau \log^2(n)\right]} \left[i_1 \mod B = i_2 \mod B\right] = \mathop{\mathbb{P}}_{B \sim \left[c\tau \log^2(n)\right]} \left[B \middle| (i_1 - i_2)\right] =$$

$$= \frac{\#\text{ πρώτων διαιρετών του } i_1 - i_2}{\#\text{ πρώτων στο } \left[c\tau \log^2(n)\right]} = \frac{\log(n)}{c\tau \log(n)} = \frac{1}{c\tau}$$

4. Από φράγμα ένωσης:

$$\mathop{\mathbb{P}}_{B \sim \left[c\tau \log^2(n)\right]} \left[\exists i_j, j \geqslant 2 : i_1 \mod B = i_j \mod B\right] \leqslant \frac{\tau - 1}{c\tau} < \frac{1}{c}$$

5. Αν η $\mathcal{E}_i$ γίνεται 1 όταν πραγματοποιείται το κακό ενδεχόμενο για το $0 \leqslant i \leqslant B-1$, έχουμε:

$$\mathop{\mathbb{E}}_{B \sim \left[c\tau \log^2(n)\right]} \left[\sum_{\substack{i_k \\ k \in [\tau]}} \mathcal{E}_{i_k}\right] = \frac{\tau}{c} \implies \mathop{\mathbb{P}}_{B \sim \left[c\tau \log^2(n)\right]} \left[\sum_{\substack{i_k \\ k \in [\tau]}} \mathcal{E}_{i_k} \geqslant \frac{\tau}{10}\right] \leqslant \frac{10}{c}$$

Άρα, με κατάλληλη επιλογή του c, ρίχνουμε την προηγούμενη πιθανότητα όσο θέλουμε.

3 Απαλλαγή από την υπόθεση της γνώσης του $\|u * v\|_0$

Το μόνο που απομενει είναι να αποβάλλουμε τον περιορισμό ότι γνωρίζουμε το $\|u * v\|_0$ μέχρι ένα $O(1)$ πολλαπλασιαστικό παράγοντα. Για να το κάνουμε αυτό ξεκινάμε από την τιμή 1 (ή οποιαδήποτε σταθερή τιμή) και διπλάσιάζούμε, παίρνοντας $2, 2^2, \ldots, 2^j, \ldots$, μέχρι να βρούμε σωστά το $u * v$; λεπτομέρεις ακολουθούν. Για κάθε j τρέχουμε το σχήμα της προηγούμενης ενότητας, με την υπόθεση ότι $\|u * v\|_0 = \Theta(2^j)$. Άμα το τρέξουμε με j τέτοιο ώστε $2^j > \|u * v\|_0$, τότε έχουμε σταθερή πιθανότητα επιτυχίας (η οποία μπορεί να γίνει $1 - \delta$ επαναλάμβοντας το σχήμα $R = \Theta(\log(1/\delta))$ φορές και κρατώντας το διάνυσμα που εμφανίζεται τουλάχιστον $R/2$ φορές. Αυτό που μένει είναι να βρούμε το σωστό τέτοιο j. Θεωρώντας τα j σε αύξουσα σειρά, αυτό που χρειαζόμαστε είναι μια ρουνία η οποία παίρνει 3 πολυώνυμα p, q, r_j και αποφαίζει αν $p(x) \cdot q(x) = r_j(x)$. Εν προκειμένη, τα p, q είναι τα πολυώνυμα των οποίων θέλουμε να υπολογίσουμε το γινόμενο, ενώ το r_j είναι το πολυώνυμο που μας δίνει η j-οστή κληση στον αλγόριθμο της προηγούμενης ενότητας, ο οποίος υποθέσαμε ότι γνωρίζει το πλήθος των συντελεστών του $p(x) \cdot q(x)$. Αυτό μπορεί να γίνει με ένα κλασικό τέχνασμα: πάρε έναν πρώτο p λιγο μεγαλύτερο από το n, ας πούμε περίπου $10n$, έναν τυχαιο αριθμό $x' \in \mathbb{Z}_p$, και έλεγξε αν $p(x') \cdot q(x') = r_j(x')$. Αν $r_j(x) = p(x)q(x)$ τότε ο έλεγχος θα είναι πάντα θετικός, ενώ αν $r_j(x) = p(x)q(x)$ η πιθανότητα να βγει θετικός ο έλεγχος είναι $\frac{n}{p} \approx \frac{1}{10}$, διότι το πολυώνυμο $p(x)q(x) - r_j(x)$ έχει βαθμό το πολύ n και κατά συνέπεια το πολύ n ρίζες στο $\mathbb{Z}_p$. Με λίγη προσοχή, μπορούμε να τοποθετήσουμε όλα τα κομμάτια του παζλ μαζί και να πάρουμε το ζητούμενο αποτέλεσμα.