Apie sveikųjų skaičių daugybos algoritmus

Jevgenijus Kirjackis

2019-05-14

Teorema. Tegul $n = b^m$, $m \in \mathbb{N}_0$

$$\begin{cases} T(n) = a\,T\left(\frac{n}{b}\right) + \Theta(n^d), \\ T(1) = c, \end{cases}$$

čia $a > 0$, $c > 0$, $b > 1$, $d \geq 0$ – konstantos.

Tada

$$T(n) = \begin{cases} \Theta(n^d), & d > \log_b a, \\ \Theta(n^d \log_b n), & d = \log_b a, \\ \Theta(n^{\log_b a}), & d < \log_b a. \end{cases}$$

Pvz. $T(n) = 2\,T\left(\frac{n}{2}\right) + \Theta(n) \Rightarrow T(n) \in \Theta(n \log n)$,

(a: 2 ↑, b: 2 ↖, $d = 1$, $\log_b a = \log_2 2 = 1$)

$T(n) = 3\,T\left(\frac{n}{2}\right) + \Theta(n) \Rightarrow T(n) \in \Theta(n^{\log 3})$.

($d = 1 < \log_2 2 = 1$)

▲ $d < \log_b a$

$$T(bn) = a\,T(n) + \Theta((bn)^d) =$$

$$= a \cdot \Theta(n^{\log_b a}) + \Theta((bn)^d) =$$

$$= \Theta(b^{\log_b a} n^{\log_b a}) + \Theta((bn)^d) =$$

$$= \Theta((bn)^{\log_b a}) + \Theta((bn)^d) \Rightarrow \text{kri}_2$$ ▲

„Ilgoji" aritmetika.

≈ 1956 - A.N. Kolmogorovo hipoteze „n^2".

```
  1 2 3
x 4 5 6
-------
  7 3 8
  6 1 5
4 9 2
-------
5 6 0 8 8
```

		1	2	3
	x	4	5	6
		6	12	18
	5	10	15	
4	8	12		
4	13	28	27	18

5 6 0 8 8

1960 - Seminaras „Matematinės problemos kibernetikoje".
Įgarsinta hip. „n^2".

Anatolijus Karatsuba: „Aš aktyviai pradėjau galvoti apie hipotezę „n^2" ir **lygiai po savaitės** gavau algoritmą sudetingumo $\Theta(n^{\log 3})$.

„Kolmogorovas **labai susijaudino**".
Per kitą seminaro posėdį pats pristatė naują metodą ir uždarė seminarą.

1903 – 1987

A.N. Kolmogorov

Anatolijus Aleksejevičius Karacuba

Gimė: 1937 m. sausio 31 d.
Grozne, TSRS

Pamyre

Mokytojai:

P.S. Aleksandrov
L.S. Pontriagin
L.A. Skorniakov
A.F. Filippov
A.N. Kolmogorov

Carl Friedrich Gauss (1777-1855).

$$(a+bi)(c+di) = \underset{1)}{a\cdot c} - \underset{2)}{b\cdot d} + (\underset{3)}{b\cdot c} + \underset{4)}{a\cdot d})\,i.$$

Bet

$$b\cdot c + a\cdot d = \overset{1)}{(a+b)}\cdot(c+d) - \overset{2)}{a\cdot c} - \overset{3)}{b\cdot d}.$$

Tegul a ir b - $2n$-ženkliai skaičiai.

Tada galima užrašyti:

$$\boxed{a = a_1\cdot w^n + a_2\,,\quad b = b_1 w^n + b_2\,,}$$

čia a_1, a_2, b_1, b_2 - n-ženkliai skaičiai.

(Pvz. $a = \underbrace{6351}_{a_1}\underbrace{1377}_{a_2} = 6351\cdot 10^4 + 1377$).

Tegul $w=10$

$$a\cdot b = (a_1\cdot w^n + a_2)\cdot(b_1 w^n + b_2) =$$

$$= a_1\cdot b_1\cdot w^{2n} + (a_1\cdot b_2 + a_2\cdot b_1)w^n + a_2\cdot b_2.$$

Bet (!)

$$\underbrace{a_1\cdot b_2 + a_2\cdot b_1} = (a_1+a_2)\cdot(b_1+b_2) - a_1\cdot b_1 - a_2\cdot b_2$$

$$= a_1\cdot b_1 + a_2\cdot b_2 - (a_1-a_2)\cdot(b_1-b_2)$$

Gavome 3 sandaugas vietoje 4. !!!

Pvz. $a = 63511377$, $b = 81026989$

$a_1 = 6351$, $a_2 = 1377$, $b_1 = 8102$, $b_2 = 6989$

$a_1 - a_2 = 6351 - 1377 = 4974$,

$b_1 - b_2 = 8102 - 6989 = 1113$,

1) $p_1 = a_1 \cdot b_1 = 6351 \cdot 8102 = 51455802$,

2) $p_2 = a_2 \cdot b_2 = 1377 \cdot 6989 = 9623853$,

3) $(a_1 - a_2) \cdot (b_1 - b_2) = 4974 \cdot 1113 = 5536062$,

$p_3 = a_1 b_2 + a_2 b_1 = 51455802 + 9623853 - 5536062 =$

$= 55543593.$

$a \cdot b = \underbrace{51455802}_{p_1}\underbrace{00000000}_{8} + \underbrace{55543593}_{p_3}\underbrace{0000}_{4} +$

$+ \underbrace{9623853}_{p_2} = 5146135645553853.$

OML (Ordinary Multiplication)

$8ž \cdot 8ž \longrightarrow 64$ sandaugos

Kar $\rightarrow 3 \cdot 4ž \cdot 4ž \xrightarrow{OML} 48$ sandaugos.

Kar $\rightarrow 3 \cdot 2ž \cdot 2ž \xrightarrow{OML} 36$ sand.

Kar $\rightarrow 3 \cdot 1 \cdot 1 \xrightarrow{OML} 3^3 = 27$ sand.

KML

Matricų daugyba

Tegul turime dvi matricas

$$A = \begin{pmatrix} a_{11} \dots a_{1m} \\ \dots\dots \\ a_{n1} \dots a_{nm} \end{pmatrix}_{n\times m} \text{ ir } B = \begin{pmatrix} b_{11} \dots b_{1k} \\ \dots\dots \\ b_{m1} \dots b_{mk} \end{pmatrix}_{m\times k}.$$

Tada $C = A \cdot B = \begin{pmatrix} c_{11} \dots c_{1k} \\ \dots\dots \\ c_{n1} \dots c_{nk} \end{pmatrix}_{n\times k}$

$$c_{ij} = \sum_{l=1}^{m} a_{il} \cdot b_{lk}. \leftarrow \text{apibr.}$$

↑ visos elementų $n \cdot k$

↖ čia „$\cdot$" – m; „$+$" – $m-1$ $\Big\}$ $+2m-1$

visos aritm. operacijų – $n \cdot k \cdot (2m-1)$.

Jeigu $k = m = n$ (A ir B kvadr.)

– visos aritm. operacijų $n^2(2n-1) =$

$= 2n^3 - n^2$. (Sudeting. $O(n^3)$.)

Nagrinėkime $n \times n$ matricas A ir B. Tegul $n = 2^k$, $k \in \mathbb{N}$.

Suskaidome mūsų matricas į $\frac{n}{2} \times \frac{n}{2}$ blokus.

$$A = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right), \quad B = \left(\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right).$$

Tada

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix},$$

čia $C_{ij} = \sum_{l=1}^{2} A_{il} \cdot B_{lj}$. (8 „$\cdot$“).

4.2

Jeigu $\frac{n}{2} > 1$ – matricų daugybą atliekame tuo pačiu metodu.

$$\begin{cases} T(1) = 1, \\ T(n) = 8\,T\left(\frac{n}{2}\right) + \Theta(n^2) \end{cases} \Rightarrow$$

$\leftarrow 2n^3 - n^2$

$$T(n) = \Theta\left(n^{\log_2 8}\right) = \Theta(n^3). \; :($$

Nieko nelaimėjome (ir neturėjome, nes nesumažinome daugybų skaičių.).

Štraseno algoritmas

1969 m Štrasenas (V. Strassen) („Gaussian elimination is not optimal", Numer. Math, V14, N3 p 364-366).

publikuoja naują rekursinių matricų daugybos algoritmą.

Skaičiuojamos 7 pagalbinės matricos.

$$
\begin{aligned}
\mathbf{M}_1 &:= (\mathbf{A}_{1,1} + \mathbf{A}_{2,2})(\mathbf{B}_{1,1} + \mathbf{B}_{2,2}) \\
\mathbf{M}_2 &:= (\mathbf{A}_{2,1} + \mathbf{A}_{2,2})\mathbf{B}_{1,1} \\
\mathbf{M}_3 &:= \mathbf{A}_{1,1}(\mathbf{B}_{1,2} - \mathbf{B}_{2,2}) \\
\mathbf{M}_4 &:= \mathbf{A}_{2,2}(\mathbf{B}_{2,1} - \mathbf{B}_{1,1}) \\
\mathbf{M}_5 &:= (\mathbf{A}_{1,1} + \mathbf{A}_{1,2})\mathbf{B}_{2,2} \\
\mathbf{M}_6 &:= (\mathbf{A}_{2,1} - \mathbf{A}_{1,1})(\mathbf{B}_{1,1} + \mathbf{B}_{1,2}) \\
\mathbf{M}_7 &:= (\mathbf{A}_{1,2} - \mathbf{A}_{2,2})(\mathbf{B}_{2,1} + \mathbf{B}_{2,2})
\end{aligned}
$$

} 7-os sandaugos!

10 „+"

Tada matricos C elementus suskaičiuojame pagal formules:

$$
\begin{aligned}
\mathbf{C}_{1,1} &= \mathbf{M}_1 + \mathbf{M}_4 - \mathbf{M}_5 + \mathbf{M}_7, & \mathbf{C}_{2,1} &= \mathbf{M}_2 + \mathbf{M}_4, \\
\mathbf{C}_{1,2} &= \mathbf{M}_3 + \mathbf{M}_5, & \mathbf{C}_{2,2} &= \mathbf{M}_1 - \mathbf{M}_2 + \mathbf{M}_3 + \mathbf{M}_6,
\end{aligned}
$$

Čia tik sumos (dar 8 „+")

Viso 18 sumavimų (vietoje 4), bet tik 7 daugybos (vietoje 8).

Algoritmo sudėtingumas:

$$T(n) = 7\,T\left(\frac{n}{2}\right) + \Theta(n^2) \Rightarrow$$

$$\Rightarrow T(n) = \Theta\left(n^{\log 7}\right) = \Theta\left(n^{2,81}\right)$$

Realizacijose rekursiją sustabdome anksčiau (apie $n = 40$).

DFT → FFT.

Tegul $A(x) = a_0 x^0 + a_1 x^1 + \dots + a_{n-1} x^{n-1}$.

(Galime laikyti, kad $n = 2^m$.)

Pažymėkime

$$\omega_{n,k} = e^{\frac{2\pi k}{n} i}, \quad k = 0, 1, \dots, n-1.$$

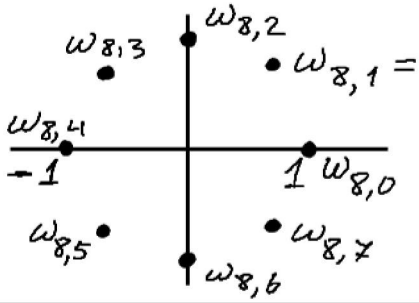
($\omega_{n,k}$ — n-asis vienetinio skritulio taškas)

Tegul $\omega_n := \omega_{n,1} = e^{\frac{2\pi i}{n}}$. Tada

$$\omega_{n,k} = (\omega_n)^k.$$

$$\omega_n^{\frac{n}{2}} = -1$$

Pvz. $n = 8$



$$\omega_8^4 = -1$$

Apibr. Diskrečioji Furje transformacija (DFT) vektoriaus $(a_0, a_1, \dots, a_{n-1})$ yra vektorius

$$\left(A(\omega_{n,0}), A(\omega_{n,1}), \dots, A(\omega_{n,n-1})\right),$$

t.y $DTF(a_0, a_1, \dots, a_{n-1}) = (y_0, y_1, \dots, y_{n-1}) =$

$$= \left(A(\omega_{n,0}), A(\omega_{n,1}), \dots, A(\omega_{n,n-1})\right) =$$

$$= \left(A(\omega_n^0), A(\omega_n^1), \dots, A(\omega_n^{n-1})\right).$$

Analogiškai atvirkštinė diskrečioji Furje transformacija daugianario $A(x)$ reikšmių $(y_0, y_1, \dots, y_{n-1})$ yra vektorius $(a_0, a_1, \dots, a_{n-1})$.

$IDFT(y_0, y_1, \dots, y_{n-1}) = (a_0, a_1, \dots, a_{n-1})$.

(t.y atstatome koeficientus pagal daugianario reikšmės (interpoliacija)).

Pritaikymas DFT daugianarių daugybai.

Tegul turime du daugianarius: $A(x)$ ir $B(x)$. Kadangi

$$(A \times B)(x) = A(x) \cdot B(x) \Rightarrow$$

$$DFT(A \times B) = DFT(A) .* DFT(B)$$

↖ panariui.

$$A \times B = IDFT(\underbrace{DFT(A) .* DFT(B)}_{O(n) \text{ operacijų.}})$$

Taigi, jeigu pavyktu suskaičiuoti DFT ir IDFT per laiką $O(n \log n)$, tai ir sandaugą daugianarių ($\Rightarrow$ sandaugą ilgų skaičių) galėtume rasti per tą pačią asimptotiką.

$$\left(\begin{array}{l} (2X+1)\cdot(3X+2) = 6X^2 + (2\cdot 2 + 1\cdot 3)X + 2 \\ 21 \cdot 32 = 672 \end{array} \right)$$

$$(3X+5)\cdot(4X+7) = 12X^2 + 41X + 35$$

$$\begin{array}{r} 35 \\ \times\ 47 \\ \hline 21 \quad 35 \\ 12\ 20 \\ \hline 12 \quad 41 \quad 35 \\ \hline 16 \quad 4 \quad 5 \end{array}$$

Greitoji Furje transformacija. (FFT)

(Cooley ir Tukey, 1965).

Tegul $A(x) = a_0 x^0 + a_1 x^1 + \dots + a_{n-1} x^{n-1}$, $n = 2^m$, $n > 1$.

Skaidom jį į du daugianarius, vieną su lyginiais, kitą su nelyginiais koeficientais:

$$A(x) = A_0(x^2) + x A_1(x^2),$$

čia

$$A_0(x) = a_0 x^0 + a_2 x^1 + \dots + a_{n-2} x^{\frac{n}{2}-1},$$

$$A_1(x) = a_1 x^0 + a_3 x^1 + \dots + a_{n-1} x^{\frac{n}{2}-1}.$$

Daugianariai $A_0(x)$ ir $A_1(x)$ turi dvigubai mažesnį laipsnį negu $A(x)$ („skaidyk" žingsnis) ir jeigu mokėsime pagal apskaičiuotus DFT(A_0) ir DFT(A_1) apskaičiuoti DFT(A) per tiesinį laiką („valdyk" žingsnis), tai gausime transformacijos laiką $\mathcal{O}(n \log n)$.

$$T(n) = \underset{a}{2}\, T\left(\frac{n}{\underset{b}{2}}\right) + \mathcal{O}(n^{\overset{d}{1}})$$

$$d = \log_b a = 2$$

Tarkime, turime apskaičiuotus vektorius

$$(y_0^0, y_1^0, \dots, y_{\frac{n}{2}-1}^0) = DFT(A_0)$$

ir

$$(y_0^1, y_1^1, \dots, y_{\frac{n}{2}-1}^1) = DFT(A_1).$$

Raskime

$$(y_0, y_1, \dots, y_{n-1}) = DFT(A) \sim$$

$$(\underbrace{y_0, y_1, \dots, y_{\frac{n}{2}-1}}, \underbrace{y_{\frac{n}{2}}, y_{\frac{n}{2}+1}, \dots, y_{n-1}}) = DFT(A).$$

Iš sąryšio $A(x) = A_0(x^2) + x \cdot A_1(x^2)$ gauname pirmai pusei koef-tų

$$y_k = y_k^0 + \omega_n^k \cdot y_k^1, \quad k = 0, \dots, \frac{n}{2} - 1.$$

$$A(\omega_n^k) = A_0(\omega_n^{2k}) + \omega_n^k \cdot A(\omega_n^{2k})$$

IFFT.

Tegul žinomas vektorius

$$(y_0, y_1, \dots, y_{n-1}) = \left(A(\omega_n^0), A(\omega_n^1), \dots, A(\omega_n^{n-1})\right),$$

$$A(x) = a_0 x^0 + a_1 x^1 + \dots + a_{n-1} x^{n-1}.$$

Reikia pagal vektorių $\{y_k\}_{k=0}^{n-1}$ atstatyti vektorių $\{a_k\}_{k=0}^{n-1}$.

DFT:

$$\begin{matrix} \omega_n^0 \\ \omega_n^1 \\ \omega_n^2 \\ \\ \omega_n^{n-1} \end{matrix} \underbrace{\begin{bmatrix} \omega_n^0 & \omega_n^0 & \omega_n^0 & \dots & \omega_n^0 \\ \omega_n^0 & \omega_n^1 & \omega_n^2 & \dots & \omega_n^{n-1} \\ \omega_n^0 & \omega_n^2 & \omega_n^4 & \dots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \omega_n^0 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \dots & \omega_n^{(n-1)(n-1)} \end{bmatrix}}_{W \leftarrow \text{Vandermondo matrica}} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

$$\bar{a} = W^{-1} \bar{y}$$

$\exists$

$$\det W = \prod_{0 \le i < j \le n-1} (\omega_n^i - \omega_n^j) \neq 0.$$

$$W^{-1} = \frac{1}{n} \begin{bmatrix} \omega_n^0 & \omega_n^0 & \omega_n^0 & \dots & \omega_n^0 \\ \omega_n^0 & \omega_n^{-1} & \omega_n^{-2} & \dots & \omega_n^{-(n-1)} \\ \omega_n^0 & \omega_n^{-2} & \omega_n^{-4} & \dots & \omega_n^{-2(n-1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \omega_n^0 & \omega_n^{-(n-1)} & \omega_n^{-2(n-1)} & \dots & \omega_n^{-(n-1)(n-1)} \end{bmatrix}$$

$$\bar{a} = W^{-1} \bar{y}$$

$$a_k = \frac{1}{n} \sum_{j=0}^{n-1} \omega_n^{-kj} \cdot y_j \qquad IDFT.$$

$$\left(y_k = \sum_{j=0}^{n-1} \omega_n^{kj} a_k \quad - \quad DFT \right).$$

Beveik tas pats. Izgi galime naudoti „skaldyk ir valdyk" metodą iz IDFT izgi galime suskaičiuoti per laiką $\Theta(n \log n)$.

$$A \times B = IDFT(\underbrace{DFT(A) .* DFT(B)}_{\Theta(n) \text{ operacijų.}})$$

2 sandaugai skaičių.

$$\omega^8 \equiv 1 \pmod{2^{16}+1}$$

$\omega^0 = 1$ —//—

$\omega^1 = 16$ —//—

$\omega^2 = 256$ —//—

$\omega^3 = 4096$ —//—

$\omega^4 = 65536 = -1 \pmod{2^{16}+1}$ (65537)

$\omega^5 = 65521 = -16$ —//—

$\omega^6 = -256$ —//—

$\omega^7 = -4096$ —//—

$\omega^8 = 1$ —//—

SEMINARAS

2019 gegužės 14 d. 09:00, SRL-I 420

Jevgenijus Kirjackis

Apie sveikųjų skaičių daugybos algoritmus

Daugindami du *n*-skaitmenų skaičius įprastu mokykliniu stulpelio metodu naudojame n^2 daugybos operacijų. 1956 m. A. N. Kolmogorovas iškėlė hipotezę, jog bet kurio *n*-skaitmenų skaičių daugybos metodo apatinis sudėtingumo įvertis yra $\Omega(n^2)$, t.y greičiau sudauginti nepavyks ir "standartinis" metodas yra asimptotiškai optimalus. 1960 m A.N. Kolmogorovo vadovaujamame seminare "Matematinės problemos kibernetikoje", šita hipotezė buvo dar kartą įgarsinta. 23-jų metų amžiaus aspirantas Anatolijus Karacuba, panaudojus *skaldyk ir valdyk* metodą, atrado elegantišką sveikųjų skaičių daugybos algoritmą sudėtingumo $O\left(n^{\log_2 3}\right) \approx O\left(n^{1,59}\right)$ ir A.N. Kolmogorovo hipotezė buvo paneigta. Kai paaiškėjo, kad "natūralioji" asimptotika $\Omega(n^2)$ nėra apatinis įvertis dviejų *n*-skaitmenų skaičių sandaugai, matematikai pradėjo ieškoti dar greitesnių algoritmų. 1971 m Arnold Schönhage ir Volker Strassen publikuoja straipsnį, kuriame pritaiko jau tuo metu žinoma (Cooley ir Tukey, 1965) greitąją Furje transformaciją, kurios sudėtingumas $O\left(n\log n\right)$, sveikųjų skaičių daugybai ir pasiekia greičio $O\left(n\log n\cdot\log\log n\right)$. Be to jie suformulavo naują hipotezę apie tai, kad neįmanoma rasti algoritmo *n*-skaitmenų skaičių daugybai greitesnio už $O\left(n\log n\right)$. Martin Fürer (2007) dar šiek tiek priartino skaičių daugybos algoritmo greitį prie, galimai, apatinės ribos $O\left(n\log n\right)$. Jo algoritmas yra $O\left(n\log n\cdot 2^{O\left(\log^* n\right)}\right)$ sudėtingumo. Covanov and Thomé (2016) pristatė algoritmą sudėtingumo $O\left(n\log n\cdot 2^{2\log^* n}\right)$.

2019-03-18 sveikųjų skaičių daugybos algoritmas sudėtingumo $O\left(n\log n\right)$ matematikų bendruomenės vertinimui pateikė David Harvey and Joris van der Hoeven (https://hal.archives-ouvertes.fr/hal-02070778/document).

Kviečiame dalyvauti.
Seminaro sekretorius A. Bugajev