Теоретические вопросы информатики, прикладной математики, компьютерных наук и когнитивно-информационных технологий

https://doi.org/10.25559/SITITO.021.202502.158-165
УДК 519.688

Эффективные вычисления

Р. Р. Айдагулов*, В. А. Васенин

ФГБОУ ВО «Московский государственный университет имени М. В. Ломоносова», г. Москва, Российская Федерация

Адрес: 119991, Российская Федерация, г. Москва, ГСП-1, Ленинские горы, д. 1

* a_rust@bk.ru

Оригинальная статья

Аннотация

В статье рассматривается более эффективный метод вычисления, чем вычисления, основанные по принципу «разделяй и властвуй». Ещё Гёте сказал: разделяй и властвуй хороший принцип, однако принцип объединяй и направляй лучше. Более эффективный метод, названный автором градуированным методом вычислений, соответствует этому принципу. Сюда в частности, относится метод быстрого умножения больших чисел используя преобразование Фурье. Алгоритмы, работающие по принципу «разделяй и властвуй», разработаны для сведения задачи с большой размерности к нескольким задачам меньшей размерности. Они описаны во многих учебниках по алгоритмам. Эффективность этого метода обосновывается описанной в этих учебниках мастер теореме. Этот метод обычно демонстрируется на алгоритме сортировки слиянием и на методе умножения Карацубы. Ранее автор на докладах в конференции и своих работах демонстрировал преимущество градуированного метода именно в задачах сортировки и умножении больших чисел.

Ключевые слова: градуированный метод, метод фильтрации, градуированная алгебра, групповая алгебра, бигрупповая алгебра

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Айдагулов Р. Р., Васенин В. А. Эффективные вычисления // Современные информационные технологии и ИТ-образование. 2025. Т. 21, № 2. С. 158-165. https://doi.org/10.25559/SITITO.021.202502.158-165

Theoretical Questions of Computer Science, Computational Mathematics, Computer Science and Cognitive Information Technologies

Efficient Computations

R. R. Aidagulov*, V. A. Vasenin

Lomonosov Moscow State University, Moscow, Russian Federation

Address: 1 Leninskie gory, Moscow 119991, GSP-1, Russian Federation

* a_rust@bk.ru

Original article

Abstract

This article considers a computational method that is more efficient than methods based on the divide-and-conquer principle. Goethe once said that divide and conquer is a good principle, but the principle of unite and guide is better. A more efficient approach, called by the author the graded method of computations, follows this latter principle. It includes, in particular, the method of fast multiplication of large numbers using the Fourier transform. Divide-and-conquer algorithms are designed to reduce a high-dimensional problem to several smaller-dimensional problems. They are described in many algorithm textbooks. The efficiency of this method is justified by the master theorem presented in those textbooks. This approach is usually illustrated by merge sort and Karatsuba multiplication. Earlier, the author demonstrated the advantage of the graded method specifically in sorting problems and large-number multiplication in conference presentations.

Keywords: graded method, filtration method, graded algebra, group algebra, bigroup algebra

Conflict of interests: The authors declares no conflict of interest.

For citation: Aidagulov R.R., Vasenin V.A. Efficient Computations. *Modern Information Technologies and IT-Education.* 2025;21(2):158-165. https://doi.org/10.25559/SITITO.021.202502.158-165

Введение

Вначале 60-х годов академик Колмогоров на семинаре поставил вопрос, можно ли умножать n битные числа быстрее, чем за $O(n^2)$ операций. В ответ Карацуба показал способ вычисления произведения за $O(n^{log_2 3})$ операций[1] [1-4]. По сути, он использовал сведение умножения многочленов большой степени к трем умножениям многочленов вдвое меньшей степени

$$(a+bX)(c+dX) = ac + (ad+bc)X + bdX^2.$$

Еще Гаусс показал, что средний член, соответствующий мнимой части произведения при умножении комплексных чисел может быть получен всего еще одним умножением

$$ad + bc = (a+b)(c+d) - ac - bd.$$

Здесь используется то, что многочлен степени *n-1*, имеющий n коэффициентов, определяется своими значениями в n точках и значения произведения многочленов является произведением значений сомножителей. Здесь младшая степень ac является произведением значений многочленов при $X = 0$, а старшая степень соответствует значению при $X = \infty$, а дополнительное произведение $(a+b)(c+d)$ соответствует значению при $X = 1$. Если бы мы взяли в качестве сомножителей многочлены степени $k-1$, имеющие k коэффициентов, то произведение имело бы $2k-1$ коэффициентов и определялось бы $2k-1$ значениями. Соответственно, алгоритм умножения с рангом умножения $2k-1$, имел бы сложность $O(n^{log_k(2k-1)})$. Случай $k = 3$ называется алгоритмом Тоома-Кука. Дальнейшее увеличение k и уменьшение экспоненты умножения $log_k(2k-1)$ не имело применения из-за роста сложности вычисления значений в целых точках и восстановления коэффициентов многочлена по его значениям.

Уже в 1963 г. появился алгоритм умножения, использующие значения многочленов в корнях из 1. Вычисление n значений многочлена степени меньше n, для всех корней степени n, называется преобразованием Фурье[2]. Вычисление n коэффициентов многочлена по n значениям многочлена называется обратным преобразованием. Сложность этих вычислений без учета роста формата чисел оценивается как $O(n\log(n))$.

Метод, использующий разбиение задачи с большим параметром n на m подзадач размера n/k автор называет методом фильтрации. Алгоритмы, использующие этот метод решаются со сложностью $T(n)$, удовлетворяющей рекуррентному соотношению:

$$T(n) = mT\left(\frac{n}{k}\right) + C(k)n. \qquad (1)$$

Сложность алгоритма, полученная из рекуррентного соотношения (1) оценивается как $O(n^\alpha)$, $\alpha = \log(m)/\log(k)$. Здесь $m = m(k)$ ранг разбиения, α – предельная экспонента. С ростом k коэффициент $C(k)$ обычно растет как $O(k^2)$ и поэтому реальная экспонента сложности оказывается заметно выше предельной $\alpha + \frac{2\log(k)}{\log(n)}$ и не имеют практического применения при больших k.

Метод фильтрации приводит к снижению экспоненту сложности $\alpha + 2/\beta$ только для достаточно больших $\beta = log_k(n)$ $(n \gg 1)$. Причиной тому является отсутствие связи между вычислениями в мелких потоках.

Умножение больших чисел сводится к задаче умножения многочленов:

$$f = \sum_{0\le i<n} a_i x^i,$$

$$g = \sum_{0\le i<n} b_i x^i, \quad \varphi = fg. \qquad (2)$$

Умножение многочленов большой степени от одной переменной можно свести к умножению многочленов от нескольких переменных малых степеней, вводя новые переменные:

$$x_1 = x, x_2 = x_1^{n_1}, \dots, x_k = x_{k-1}^{n_{k-1}},$$
$$n_1 n_2 \dots n_k > \deg(\varphi) \ge 2n - 2.$$

Алгоритм Карацуба по сути сводится к такому разбиению многочленов большой степени на многочлены с малыми степенями с $n_i = 2$. Учитывая, что произведение многочленов по каждой переменной имеет степень 2, т.е. три коэффициента, для вычисления произведения многочленов степени не больше $n = 2^k$, требуется вычислить $3^k = n^{log_2 3}$ значений многочленов и столько же коэффициентов произведения многочленов. В то же время, имеется возможность избежать существенного повышения степени за счет выбора новых переменных и показателей n_i. Пусть n_i взаимно простые числа (например, степени малых простых чисел) и их произведение превышает степень произведения многочленов $N = \prod_i n_i > \deg(\varphi)$. Тогда группу градуировки многочленов Z_N можно представить в виде прямой суммы:

$$Z_N = Z_{n_1} \oplus Z_{n_1} \oplus \dots \oplus Z_{n_k} \qquad (3)$$

с образующими $x_i = x^{N/n_i}$. Так можно избежать от существенного роста степени многочленов и увеличения формата для коэффициентов, сводя вычисление коэффициентов и переносы отдельно по каждой переменной [5].

В этой работе покажем, что градуированный метод более эффективен, чем метод фильтрации не только в задачах умножения больших чисел и многочленов с коммутирующими переменными. Суть градуированного метода заключается в том, что все n^2 попарных операций выполняются в $O(n)$ объединенных операциях и выделении нужных членов исходя из их весов. Далее перенесем

[1] Дасгупта С., Пападимитриу Х., Вазирани У. Алгоритмы ; пер. с англ. А. С. Куликова ; под ред. А. Шеня. 2-е изд., стер. М. : Изд-во МЦНМО, 2019. 318 с.; Knuth D. E. The Art of Computer Programming: Vol. 3, Sorting and Searching. 2nd ed. Addison-Wesley, 1998. 780 p.

[2] Ноден П. Алгебраическая алгоритмика : с упражнениями и решениями / П. Ноден, К. Китте; пер. с франц. В.А. Соколова под ред. Л.С. Казарина. М. : Мир, 1999. 719 с.

градуированный метод на матричные операции и сравним эффективность с имеющимися методами фильтрации.

Методы быстрого умножения квадратных матриц

Стандартное умножение матриц порядка $n \times n$ состоит из n^3 умножений и $n^2(n-1)$ сложений. Первый алгоритм с меньшей сложностью представлен Штрассеном в 1969 г. [6]. Почти все алгоритмы быстрого умножения матриц относятся к методу фильтрации путем уменьшения ранга умножения используя различные методы. Разные методы быстрого умножения матриц описаны в обзорном докладе [7]. В этом обзоре больший акцент сделан на групповые методы [8]. Стоит обратить внимание и на работу [9]. Как будет видно ниже в не коммутативном случае групповые методы менее эффективны. Наиболее эффективным с точки зрения уменьшения экспоненты умножения матриц является метод Винограда Копперсмита [10]. С некоторыми улучшениями этого метода предельная экспонента умножения доведена до $\alpha = 2.3728639$ в [11]. Сам метод с некоторыми улучшениями на русском языке описан в работе [12]. В [13] получены более точные оценки ранга умножения для матриц малых порядков. Современное состояние метода Винограда Копперсмита и обзор тематики матричного умножения на 2025 г. имеется в [14]. Оценка предельной экспоненты на 2025 г равен $\alpha = 2.371339$.

Бигрупповые алгебры и умножение матриц

Умножение элементов в коммутативных алгебрах вычисляется вложением их в конечномерную групповую алгебру. Образующие групповой алгебры являются корнями из 1. Умножение двух элементов

$$A = \sum_{g \in G} a(g)g\,, B = \sum_{g \in G} b(g)g$$

дает их произведение по формуле:

$$C = \sum_{g \in G} c(g)g\,, \qquad c(g) = \sum_{g_1 g_2 = g} a(g_1)b(g_2). \qquad (4)$$

Пусть G^* группа характеров. Каждый характер $\mu \in G^*$ определяет μ значение элементов и их произведения как произведение значений:

$$\mu(A) = \sum_{g \in G} a(g)\mu(g)\,, \qquad \mu(B) = \sum_{g \in G} b(g)\mu(g)\,,$$

$$\mu(C) = \sum_{g_1, g_2} a(g_1)b(g_2)\mu(g_1)\mu(g_2)$$

$$= \mu(A)\mu(B).$$

При этом коэффициенты разложения выделяются стандартным образом:

$$c(g) = \frac{1}{n}\sum_{\mu \epsilon G^*} \mu(g^{-1})\mu(C)\,,$$

$$n = |G| \qquad (5)$$

Отметим, что в каждом вычисленные значения $\mu(C)$ содержится каждая из n^2 произведений коэффициентов $a(g_1)b(g_2)$ ровно один раз с некоторым знаком $\mu(g_1)\mu(g_2)$. Их вычисление проходит как бы параллельно. Формула (5) выделяет нужное значение из вычисленных.

В не коммутативном случае этот способ не работает. Здесь произведение двух градуированных элементов базиса $fg \neq gf$ отличаются при замене порядка расположения множителей. Если учитывая не коммутативность умножения элементы fg, gf считаем разными элементами групповой алгебры, то приходится увеличить размерность группы и ранг умножения в несколько (как минимум в два раза для не единичных элементов) раз и получим менее эффективные алгоритмы. Конечномерные не коммутативные градуированные алгебры являются цветными алгебрами [15], где произведение двух образующих элементов при перестановке множителей отличаются на множитель, являющийся корнем из 1. Группа градуировки при этом остается коммутативной и имеет структуру дискретного симплектического пространства [15]. Пусть G коммутативная группа с конечным числом образующих, через $K[G]$ обозначим групповую алгебру с коэффициентами из G. В случае конечной группы элементы групповой алгебры представляются в виде сумм $f = \sum_{g \in G} a(g)g$, в случае бесконечной группы как функции на группе $a(g)$ со значениями в кольце. Действие элемента g_1 на элемент f групповой алгебры определяется $f \to fg_1 = \sum_{g \in G} a(g)g\,g_1 = \sum_{g \in G} a(gg_1^{-1})g$, т.е. через сдвиг функции $a(g) \to a(gg_1^{-1})$. Элементы группы представляются как степени образующих, а сами степени q представляют градуировку одночлена. Следовательно, элементы групповой алгебры можем представлять как функции $a(g) = f(q)$. Тогда действие умножения на элемент g_1 представляется сдвигом вправо на h (соответствующий элементу g_1) функции $f(q) \to f(q-h) = \exp\left(h\frac{\partial}{\partial q}\right)f(q)$. Элементы характеры из $\mu \in G'$ действуют в $K[G]$ как умножение на корень из 1 – $exp(ipq) = \mu(g)$. Обозначим через $K[G, G']$ алгебру операторов, порожденных действиями элементов группы и характеров. Автор называет эту алгебру бигрупповой алгеброй группы G. Она не коммутативна. Все пары произведений операторов μg образуют базис Сильвестра. Умножение на g в групповой алгебре производит сдвиг функции на группе как и оператор:

$$gf(x) = e^{-h\frac{\partial}{\partial q}}f(q) = f(q-h).$$

Характер умножает на функцию

$$\mu\,(f(q)) = e^{ipq}f(q).$$

Отсюда получается коммутационное соотношение:

$$\mu g = \mu(g)g\mu, \qquad (6)$$

называемое иногда соотношением неопределенности Гейзенберга.

С учетом приведенного представления бигрупповая алгебра является алгеброй линейных псевдодифференциальных операторов на линейном пространстве функций на группе $K[G]$. Бигрупповая алгебра является скрещенным произведением групповых алгебр группы G и группы характеров G^*, элементы которых представляются как линейные операторы на групповой алгебре:

$$\sum_{\mu\in G^*, g\epsilon G} a(\mu,g)\mu g.$$

Для нас важно представление бигрупповой алгебры как алгебры многочленов с не коммутирующими переменными[3] [16]. Пусть конечная группа G представлена как прямая сумма циклических подгрупп. Выберем образующие этих подгрупп $y_1, y_2, \dots, y_k$. Группа характеров имеет ту же структуру и там выберем образующие $x_1, x_2, \dots, x_k$. Эти переменные удовлетворяют условию нормировки

$$x_i^{n_i} = 1 = Id = y_i^{n_i}$$

и коммутационным соотношениям (6), что можно переписать в виде:

$$x_i y_j = \theta_i^{\delta_i^j} y_j x_i.$$

Базис Сильвестра состоит из одночленов

$$\prod_{l\le k} x_l^{i_l} y_l^{j_l}, \qquad 0 \le i_l, j_l < n_l, \qquad \prod_l n_l = n.$$

В дальнейшем для удобства используем краткие тензорные обозначения:

$$\prod_{l\le k} x_l^{i_l} y_l^{j_l} = x^I y^J, I = (i_1, i_2, \dots, i_k),$$
$$J = (j_1, j_2, \dots, j_k). \quad (7)$$

Два элемента базиса Сильвестра коммутируют между собой по формуле:

$$x^{I_1} y^{J_1} x^{I_2} y^{J_2} = \theta^{[(I_1,J_1),(I_2,J_2)]} x^{I_2} y^{J_2} x^{I_1} y^{J_1} \quad (8).$$

Здесь $(I_1,J_1),(I_2,J_2)$ градуировки базисных элементов, а $\theta^{[(I_1,J_1),(I_2,J_2)]} = \prod_l \theta_l^{i_{l1} j_{l2} - i_{l2} j_{l1}}$ развернутая запись коэффициента коммутационных соотношений. Под квадратными скобками определяется кососимметричное скалярное произведение, делающие пространство градуировок симплектическим пространством[4], а образующие x_l, y_l образующими Дарбу. Наряду с $2k$ образующими Дарбу, можно ввести $2k$ образующие Клиффорда z_i, удовлетворяющие коммутационным соотношениям $z_i z_j = \omega_i z_j z_i, i < j$. Так получим представление бигрупповой алгебры как обобщенной Клиффордовой алгебры или алгебры обобщенных спиноров в терминологии физиков. Они обладают некоторыми особыми свойствами[5] [17, 18]. Наряду с базисом Сильвестра удобно пользоваться базисом Вейля Швингера $W(I,J) = \theta^{-(I,J)/2} x^I y^J$.

В случае конечной группы порядка n, не делящейся на характеристику поля, и при наличии корней из 1 (значений характеров) в поле, бигрупповая алгебра совпадает с алгеброй матриц порядка $n \times n$ [16]. В качестве группы G можно взять группу перестановок строк, порожденный элементом y, а в качестве группы характеров, диагональные матрицы, порожденные элементом x, где:

$$x = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & \theta & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \theta^{n-1} \end{pmatrix}, \qquad y = \begin{pmatrix} 0 & 0 & & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ \vdots & & \ddots & \vdots & \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}.$$

Легко проверяется, что соотношение (6,8) выполняется: $xy = \theta yx$. Транспонирование сводится к $x^T = x, y^T = y^{-1}$ (*ортогональность*).

Отметим, что $X_s = \frac{1}{n}\sum_I \theta^{-(I,s)} x^I$ представляет матрицу, у которой на пересечении -ой строки и s-го столбца 1, а все остальные элементы равны нулю. Соответственно, $X_s y^J$ представляет матрицу, у которой на пересечении -ой строки и $s - J$-го столбца 1, а все остальные элементы равны нулю. Это позволяет разложить матрицу $A = (a_{I,J})$ в базисе Сильвестра:

$$A = \sum_{I,J} a_{I,J} X_I\, y^{I-J} = \frac{1}{n}\sum_{I,t,s} a_{I,I-t}\theta^{-(I,s)} x^s y^t$$
$$= \sum_{t,s} a(t,s) x^s y^t. \quad (9)$$

Пусть имеется разложение по базису Сильвестра $A = \sum_{t,s} a(I,J) x^I y^J$. Придание переменной $x = \theta^s$ значения $s = (s_1, s_2, \dots, s_k)$ означает выделение строки с номером s у матрицы. Из (9) получаем

$$A = \sum_{I,J} a(I,J) x^I y^J,$$
$$a(I,J) = \frac{1}{n}\sum_r a_{r,r-J}\theta^{-(r,I)} \quad (10).$$

Следовательно, подстановка $x = \theta^s$ действительно обнуляет все строки, за исключением строки с номером s:

$$A_s = \frac{1}{n}\sum_{r,I,J} a_{r,r-J}\theta^{-(r,I)+(s,I)} y^J = \sum_J a_{s,s-j} X_s y^J.$$

Придание переменной $y = \theta^t$ значения $t = (t_1, t, \dots, t_k)$ означает окрашивание диагоналей, т.е. расставление одинаковых знаков $\theta^{(t,J)}$ при одинаковом смещении J от главной диагонали матрицы:

$$A_t = \frac{1}{n}\sum_{r,I,J} a_{r,r-J}\theta^{-(r,I)+(t,J)} x^I = \sum_{r,J} a_{r,r-J}\,\theta^{(t,J)} X_r y^J.$$

Чтобы действительно выделить диагональ, соответствующий определенному значению смещения J' надо суммировать

$$\frac{1}{n}\sum_t \theta^{-(t,J')} A_t = \sum_r a_{r,r-J'} X_r y^{J'}.$$

[3] Айдагулов Р. Р. Бигрупповые алгебры и их автоморфизмы // Дневник науки. 2019. № 1(25). С. 25. EDN: HNGOYX

[4] Фоменко А. Т. Симплектическая геометрия. Методы и приложения. М. : Изд-во МГУ, 1988. 416 с.

[5] Айдагулов Р. Р. Бигрупповые алгебры и квантовая комбинаторика // Дневник науки. 2019. № 1(25). С. 26. EDN: YWAJVB

Сумма значений при $x = \theta^s, y = \theta^t$ с соответствующими весами дает

$$\frac{1}{n^2}\sum_{t,s}\theta^{(I,s)-(t,J)}A_{t,s} = a(I,J).$$

Не сложно выделить так же некоторые коэффициенты у произведения многочленов $C = AB$:

$$C(I,0) = \frac{1}{n^2}\sum_{t,s}\theta^{(I,s)}A(t,s)B^T(t,s) \qquad (11)$$

$$C(0,J) = \frac{1}{n^2}\sum_{t,s}\theta^{-(t,J)}A^T(t,s)B(t,s) \qquad (12)$$

Для вычисления коэффициентов $C(I,J)$ потребуется комбинировать частями сумм с одинаковыми значениями $\theta^{(I,s)-(t,J)}$ с весовыми коэффициентами. Удобнее вычисления производить в бигрупповой алгебре группы $(Z_2)^k$. Удобство исходит за счет избавления умножений на комплексные константы (они принимают всего два значения +1 и -1), так и от необходимости соблюдать знаки в показателях $(\pm 1)^{-1} = \pm 1,\ (\pm 1)^0 = 1$.

Тогда образующими алгебры будут $2k$ матриц x_i, y_i:

$$x_i = \begin{pmatrix} E_{i-1} & 0 \\ 0 & -E_{i-1} \end{pmatrix} \otimes E_{k-i}, \qquad y_i = \begin{pmatrix} 0 & E_{i-1} \\ E_{i-1} & 0 \end{pmatrix} \otimes E_{k-i}.$$

Здесь E_i- единичная матрица порядка 2^i. Квадраты этих матриц являются единичными матрицами. Коммутационные соотношения упрощаются заменой $\theta = -1$ и имеют вид:

$$x_i y_j = (-1)^{\delta_i^j} y_j x_i.$$

Элемент $x^I y^J$ назовем четным, если $x^I y^J = y^J x^I$, иначе не четным.

Четность матриц в нашем случае соответствует их симметричности. Всего $\frac{n(n+1)}{2}$, $n = 2^k$ элементов базиса Сильвестра четные, оставшиеся $\frac{n(n-1)}{2}$ нечетные. Четность произведения двух элементов равно сумме четностей и их коммутируемости, т.е. если они коммутируют, то произведение двух элементов одинаковой четности является четным, иначе нечетным. Если они не коммутирует, то наоборот, произведение элементов одинаковой четности будет нечетным, произведение разной четности – четным.

Тогда вместо формул (11), (12) можно пользоваться единой формулой:

$$C(I,J) = \frac{1}{n^2}\left(\sum_{(t,J)=(I,s)} A(t,s)B^T(t,s) - (-1)^{(I,J)}\sum_{(t,J)\neq(I,s)} A(t,s)B^T(t,s)\right) \qquad (13)$$

$$\sum_{t,s}\mu(t,s)(A)\mu(t,s)(B^T)(-1)^{(t,J')+(s,I')}, \quad (4)$$

Так можно выделить компоненты произведения матриц и тем самым решить задачу вычисления произведения матриц за $O(n\,ln^2(n))$ операций.

Симметрии и обращение матриц

Для каждого элемента базиса Сильвестра Z отображения матриц

$$S_Z: A \to ZAZ^{-1},\ \ S_{Z'}: A \to ZA^TZ^{-1}$$

являются автоморфизмами алгебры. Первые не меняют порядок произведения, вторые меняют $S_{Z'}(AB) = S_{Z'}(B)S_{Z'}(A)$. Они образуют группу симметрий $S_{Z_1}\left(S_{Z_2}(A)\right) = S_{Z_1Z_2}(A)$. Симметрии со штрихами назовем нечетными. Два раза штрих соответствует случаю без штриха.

Далее рассмотрим эти автоморфизмы только в бигрупповой алгебре группы $(Z_2)^k$. Их порядок равен 2. Обозначим через S_i симметрию на -ой компоненте $S_i(A) = x_i y_i A^T y_i x_i = S_{(x_i y_i)'}(A)$. Легко проверить, что произведение $AS_i(A)$ инвариантно относительно этой симметрии. Образуем цепочку матриц

$$A_0 = A, \qquad A_i = A_{i-1}S_i(A_{i-1}). \quad (14)$$

Тогда матрица A_k инвариантно относительно всех симметрий S_i, следовательно она равна единичной матрице с точностью до константы. Так как симметрии не меняют дискриминант, выполняется соотношение

$$\det(A_i) = \det(A)^{2^i} \to A_k = \det(A)\,E \qquad (15)$$

Так как $A_k = AS_1(A_0)S_2(A_1)\dots S_k(A_{k-1}) = AA_D = \det(A)\,E$,

матрица A_D матрицей алгебраических дополнений и детерминант и обращение матрицы вычисляется через $2k-1, k = \log(n)$ произведений матриц. С учетом того, что произведение вычисляется за $O(n^2\log(n))$ операций получаем, что обращение матриц (при $\det(a) \neq 0$) имеет примерно такую же сложность. Этим несколько уточняется известный факт равенства экспоненты сложности операции умножения и операции обращения матриц[6] [19, 20].

Приведенный подход соответствует вычисление обратного натурального числа a по модулю простого числа по алгоритму $a^{-1}\ mod\ p = a^{p-2}\ mod\ p$. Есть и более короткий алгоритм (аналог алгоритма Евклида), используя аддитивные аналоги симметрий можно вычислить представление в виде произведения диагональной матрицы и ортогональной матрицы.

Заключение

Разработаны алгоритмы для матричных операций со сложностью $O(n^2\log(n))$ сложений вычитаний и $O(n^2)$ умножений и делений на постоянную $n^2 = 2^{2k}$. Они многократно превосходят любые алгоритмы, основанные на методе фильтрации при больших n. Соответственно, позволяют решать задачи линейного программирования даже при значениях n порядка 10^7.

[6] Гантмахер Ф. Р. Теория матриц. 5-е изд. М. : Физматлит, 2010. 559 с.; Хорн Р. Матричный анализ / Р. Хорн, Ч. Джонсон; пер. с англ. Х.Д. Икрамова; под ред. Х.Д. Икрамова. М. : Мир, 1989. 655 с.

References

1. Karatsuba A., Ofman Yu. Multiplication of many-digital numbers by automatic computers. *Dokl. Akad. Nauk SSSR.* 1962;145(2):293-294. (In Russ.)
2. Karatsuba A.A. Comments to my works, written by myself. *Proceedings of the Steklov Institute of Mathematics.* 2013;282(Suppl 1):1-23. https://doi.org/10.1134/S0081543813070018
3. Karatsuba A.A. The Complexity of Computations. *Trudy Matematicheskogo Instituta imeni V.A. Steklova* = Proceedings of the Steklov Institute of Mathematics. 1995;211:169-183.
4. Aidagulov R.R., Glavatsky S.T. Graded Computing. *Modern Information Technologies and IT-Education.* 2019;15(2):274-282. (In Russ., abstract in Eng.) https://doi.org/10.25559/SITITO.15.201902.274-282
5. Gashkov S.B., Sergeev I.S. Multiplication. *Chebyshevskii Sbornik*. 2020;21(1):101-134. (In Russ., abstract in Eng.) https://doi.org/10.22405/2226-8383-2020-21-1-101-134
6. Strassen V. Gaussian elimination is not optimal. *Numerische Mathematik*. 1969;13:354-356.
7. Demmel J., Dumitriu I., Holtz O., Kleinberg R. Fast matrix multiplication is stable. *arXiv*:math/0603207. 2006. https://doi.org/10.48550/arXiv.math/0603207
8. Cohn H., Kleinberg R., Szegedy B., Umans C. Group-theoretic algorithms for matrix multiplication. In: 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05). Pittsburgh, PA, USA: IEEE Press; 2005. p. 379-388. https://doi.org/10.1109/SFCS.2005.39
9. Cohn H., Umans C. Fast matrix multiplication using coherent configurations. In: Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms (SODA '13). New Orleans, Louisiana: Society for Industrial and Applied Mathematics, USA; 2013. p. 1074-1086.
10. Coppersmith D., Winograd S. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation.* 1990;9(3):251-280. https://doi.org/10.1016/S0747-7171(08)80013-2
11. Le Gall F. Powers of tensors and fast matrix multiplication. In: Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC '14). New York, NY, USA: Association for Computing Machinery; 2014. p. 296-303. https://doi.org/10.1145/2608628.2608664
12. Zhdanovich D.V. The matrix capacity of a tensor. *Journal of Mathematical Sciences*. 2012;186(4):599-643. https://doi.org/10.1007/s10958-012-1009-7
13. Smirnov A.V. The bilinear complexity and practical algorithms for matrix multiplication. *Computational Mathematics and Mathematical Physics.* 2013;53:1781-1795. https://doi.org/10.1134/S0965542513120129
14. Moosbauer J., Poole M. Flip Graphs with Symmetry and New Matrix Multiplication Schemes. In: Proceedings of the 2025 International Symposium on Symbolic and Algebraic Computation (ISSAC '25). New York, NY, USA: Association for Computing Machinery; 2025. p. 233-239. https://doi.org/10.1145/3747199.3747566
15. Aidagulov R.R., Shamolin M.V. Groups of colors. *Journal of Mathematical Sciences*. 2009;161(5):615-627. https://doi.org/10.1007/s10958-009-9592-y
16. Aidagulov R.R. Fast multiplication algorithms. *Intellektual'nye Sistemy. Teoriya i Prilozheniya* = Intelligent Systems. Theory and Applications. 2022;26(1):134-139. (In Russ., abstract in Eng.) EDN: DVKLFX
17. Aidagulov R.R., Glavatsky S.T., Mikhalev A.V. Clustering Models. *Journal of Mathematical Sciences*. 2022;262(5):603-616. https://doi.org/10.1007/s10958-022-05841-9
18. Aidagulov R.R. Bigroup Algebras and Potter's Theorem. *Intellektual'nye Sistemy. Teoriya i Prilozheniya* = Intelligent Systems. Theory and Applications. 2022;26(1):140-145. (In Russ., abstract in Eng.) EDN: VBJJXI
19. Alexeev N., Aidagulov R., Alekseyev M.A. A Computational Method for the Rate Estimation of Evolutionary Transpositions. In: Ortuño F., Rojas I. (eds.) Bioinformatics and Biomedical Engineering. IWBBIO 2015. *Lecture Notes in Computer Science*. Vol. 9043. Cham: Springer; 2015. p. 471-480. https://doi.org/10.1007/978-3-319-16483-0_46
20. Aidagulov R.R., Shamolin M.V. Fast Matrix Multiplication by Using Color Algebras. *Journal of Mathematical Sciences*. 2017;227(4):402-406. https://doi.org/10.1007/s10958-017-3593-z

Поступила 01.04.2025; одобрена после рецензирования 10.06.2025; принята к публикации 08.07.2025.

Submitted 01.04.2025; approved after reviewing 10.06.2025; accepted for publication 08.07.2025.

Об авторах:

Айдагулов Рустем Римович, старший научный сотрудник кафедры теоретической информатики отделения математики механико-математического факультета, ФГБОУ ВО «Московский государственный университет имени М. В. Ломоносова» (119991, Российская Федерация, г. Москва, ГСП-1, Ленинские горы, д. 1), кандидат физико-математических наук, **ORCID: https://orcid.org/0000-0002-6579-429X,** a_rust@bk.ru
Васенин Валерий Александрович, заведующий кафедрой математического моделирования и компьютерных исследований механико-математического факультета; заведующий лабораторией автоматизации экспериментальных исследований НИИ механики МГУ, ФГБОУ ВО «Московский государственный университет имени М. В. Ломоносова» (119991, Российская Федерация, г. Москва, ГСП-1, Ленинские горы, д. 1), доктор физико-математических наук, профессор, **ORCID: https://orcid.org/0000-0003-2499-6132,**
vasenin@msu.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the authors:

Rustem R. Aidagulov, Senior Researcher of Department of Theoretical Informatics, Faculty of Mechanics and Mathematics, Lomonosov Moscow State University (1 Leninskie gory, Moscow 119991, GSP-1, Russian Federation), Cand. Sci. (Phys.-Math.), **ORCID: https://orcid.org/0000-0002-6579-429X,** a_rust@bk.ru
Valery A. Vasenin, Head of the Department of Mathematical Modeling and Computer Research at the Faculty of Mechanics and Mathematics; Head of the Laboratory of Automation of Experimental Research at the Research Institute of Mechanics at Moscow State University, Lomonosov Moscow State University (1 Leninskie gory, Moscow 119991, GSP-1, Russian Federation), Dr. Sci. (Phys.-Math.), Professor, **ORCID: https://orcid.org/0000-0003-2499-6132,**
vasenin@msu.ru

All authors have read and approved the final manuscript.