ЧЕБЫШЕВСКИЙ СБОРНИК
Том 21. Выпуск 1.

УДК 519.7 DOI 10.22405/2226-8383-2020-21-1-101-134

Умножение[1]

С. Б. Гашков, И. С. Сергеев

Гашков Сергей Борисович — доктор физико-математических наук, профессор, Московский государственный университет им. М. В. Ломоносова (г. Москва).
e-mail: sbgashkov@gmail.com
Сергеев Игорь Сергеевич — кандидат физико-математических наук; начальник лаборатории, Научно-исследовательский институт «Квант» (г. Москва).
e-mail: isserg@gmail.com

Аннотация

В работе предпринимается обзор современного состояния теории быстрых алгоритмов умножения чисел и многочленов. Рассматривается процесс эволюции методов умножения от первых блочных алгоритмов Карацубы и Тоома 1960-х гг. к методам 1970-х гг., опирающимся на дискретное преобразование Фурье (ДПФ), и далее к новейшим методам, разработанным в 2007–2019 гг. Современные методы умножения сочетают использование специальных алгебраических структур, переход к приближенным вычислениям, особые формы преобразований Фурье: многомерное ДПФ, аддитивный аналог ДПФ. Эти и другие существенные для быстрых методов умножения концепции подробно рассматриваются в настоящем обзоре. Отдельно предусмотрено введение в теорию ДПФ с извлечением необходимых для изложения материала фактов. В заключительной части обзора приводятся краткие сведения о результатах в области параллельных алгоритмов умножения, аккуратных оценок сложности базовых методов умножения, алгоритмов умножения в реальном времени, мультипликативной сложности умножения многочленов над конечными полями. Отмечены модели вычислений, в которых умножение имеет линейную или квадратичную сложность.

Ключевые слова: быстрые вычисления, умножение, сложность, дискретное преобразование Фурье, аддитивное ДПФ.

Библиография: 89 названий.

Для цитирования:
С. Б. Гашков, И. С. Сергеев. Умножение // Чебышевский сборник, 2020, т. 21, вып. 1, с. 101–134.

[1]Исследование выполнено при поддержке гранта РФФИ (проект 19-01-00294а).

CHEBYSHEVSKII SBORNIK
Vol. 21. No. 1.

UDC 519.7 DOI 10.22405/2226-8383-2020-21-1-101-134

Multiplication

S. B. Gashkov, I. S. Sergeev

Gashkov Sergey Borisovich — doctor of science, professor, Lomonosov Moscow State University (Moscow).
e-mail: sbgashkov@gmail.com
Sergeev Igor Sergeevich — candidate of science; head of laboratory, Federal State Unitary Enterprise "Scientific and Research Institute "Kvant"(Moscow).
e-mail: isserg@gmail.com

Abstract

The paper provides a survey of the modern state of the theory of fast multiplication of numbers and polynomials. We consider the development of multiplication methods from the initial block algorithms of 1960s due to Karatsuba and Toom to the methods of 1970s based on the Discrete Fourier transform (DFT) and further to the novel methods invented in 2007–2009. Modern multiplication methods combine exploiting of special algebraic structures, the use of approximate computations, special forms of Fourier transforms, namely, multidimensional DFT, additive analogue of DFT. These and other concepts essential for the fast multiplication algorithms are thoroughly discussed in the present survey. In particular, we provide an introduction to the theory of DFT with derivation of facts necessary for the exposition. The final part of the survey contains a brief discussion of results on parallel multiplication algorithms, accurate complexity bounds of the basic methods, online multiplication algorithms, multiplicative complexity of the multiplication of polynomials over finite fields. We mention computational models where multiplication has either linear, or quadratic complexity.

Keywords: fast computations, multiplication, complexity, discrete Fourier transform, additive DFT.

Bibliography: 89 titles.

For citation:
S. B. Gashkov, I. S. Sergeev, 2020, "Multiplication" , *Chebyshevskii sbornik*, vol. 21, no. 1, pp. 101–134.

1. Введение

Интерес к приемам быстрых вычислений возникал еще у математиков древности: достаточно вспомнить алгоритм Евклида для наибольшего общего делителя. Но только в начале 20-го века появилась математическая теория, позволяющая строго формулировать и давать точные ответы на вопросы типа «как быстро работает данный алгоритм?» или «какой алгоритм быстрее?». Центральное понятие этой теории — *сложность*.

Теория сложности, которую можно отнести к математической логике, дискретной математике или математической кибернетике, возникла в работах американского инженера-теоретика К. Шеннона в 1930-40-е гг.

Технологическая революция середины 20-го века, которая привела к рождению современной электроники, спровоцировала стремительное развитие сопутствующих научных дисциплин: информатики, кибернетики, вычислительных методов, теории алгоритмов и сложности.

Помимо Шеннона, значительный вклад в создание математического аппарата этих теорий внес А. Н. Колмогоров (см., например, [13]). Он же активно поддерживал работы по новым направлениям в СССР и за рубежом.

Быстрые алгоритмы для отдельных арифметических операций стали появляться, не дожидаясь ни окончательного оформления теории, ни появления мощных вычислительных средств. Несколько примеров. В 1938 г. Д. Лехмер построил ускоренную версию алгоритма НОД Евклида, что интересно, специально ориентированную для применения в вычислительных машинах. Метод Лехмера лежит в основе современных быстрых алгоритмов НОД. Год спустя А. Брауэр опубликовал оптимальный по числу умножений метод возведения в натуральную степень — он примерно вдвое быстрее тривиального бинарного метода.

В 1950-е годы инженеры и математики уже активно разрабатывали эффективные конструкции схем или программ для быстрой реализации арифметики. Из теоретических результатов можно отметить полученное в конце 1950-х гг. учениками А. Г. Витушкина (который, в свою очередь, был учеником Колмогорова) В. Я. Паном и Э. Г. Белагой окончательное решение задачи о сложности вычисления значения заданного многочлена в точке. Чуть раньше в США этой задачей занимался Т. Моцкин. В частности, ими были построены быстрые алгоритмы вычисления многочлена, использующие примерно в 2 раза меньше умножений коэффициентов, чем в схеме Горнера.

Однако именно с появлением быстрого метода умножения чисел А. А. Карацубы в 1960 г. (а вернее, с его опубликованием в 1962 г.) связывается становление нового научного направления — теории быстрых вычислений. Экстраординарное влияние, которое оказал результат Карацубы на развитие теории, объясняется совокупным действием нескольких обстоятельств. С одной стороны, результат оказался сенсационным (многие считали, что школьный метод умножения оптимален). С другой стороны, метод Карацубы демонстрирует эффективность общего приема сведения большой задачи к задачам меньшего размера, который затем успешно применялся для построения быстрых алгоритмов многих других арифметических операций. В-третьих, умножение является фундаментальной операцией — строительным блоком для многих других операций арифметики, которые, как следствие, автоматически ускоряются за счет ускорения алгоритма умножения.

В результате, как объективные факторы (фундаментальность операции умножения), так и субъективные (эффект, произведенный методом Карацубы) привели к тому, что задача исследования сложности умножения стала центральной для теории быстрых вычислений.

К моменту появления первых быстрых алгоритмов умножения (Карацубы и Тоома) в начале 1960-х гг. аппарат теории сложности вполне сформировался, и центральный вопрос о сложности реализации произвольной булевой функции был решен в асимптотическом смысле для нескольких основных вычислительных моделей О. Б. Лупановым.

Можно увидеть сходство сюжетов развития быстрых методов вычисления булевых функций и быстрых методов умножения. Первую нетривиальную оценку сложности вычисления функции дает метод каскадов Шеннона. При помощи формулы

$$f(x_1, x_2, \ldots, x_n) = f(1, x_2, \ldots, x_n) \cdot x_1 \vee f(0, x_2, \ldots, x_n) \cdot \overline{x_1}$$

он сводит исходную задачу к двум подобным задачам меньшего размера — реализации подфункций ($\overline{x}$ — отрицание переменной x). Аналогично, метод Карацубы сводит исходное умножение к трем умножениям меньшего размера, только используемая в нем формула менее очевидна. Асимптотически оптимальный метод Лупанова нетривиально развивает идею метода каскадов и состоит в сведении к вычислению функции в локальных областях. Принцип, лежащий в основе метода, получил название принципа локального кодирования. В задаче умножения этому принципу соответствует идея интерполяции. На интерполяции основаны все быстрые методы умножения, начиная с метода Тоома.

Наряду с числовым умножением, мы рассматриваем умножение многочленов одной переменной. Это мотивируется, во-первых, тем, что методы умножения чисел и многочленов тесно взаимосвязаны, а во-вторых, значение арифметики многочленов продолжает расти по мере развития криптографии. Пожалуй, особенно актуальны методы быстрого умножения многочленов над конечными полями характеристики 2.

Формально, умножение n-разрядных чисел можно интерпретировать как булев оператор, входами и выходами которого являются двоичные разряды соответственно перемножаемых чисел и произведения. Для оценки сложности булевых операторов удобно использовать вычислительную модель булевых схем. *Булевы схемы* строятся из функциональных элементов с двумя входами и одним выходом, реализующих всевозможные булевы операции. *Сложностью* схемы называется число функциональных элементов в ней. Последнее понятие очень близко к понятию *битовой сложности* вычисления, а само понятие схемы очень близко к понятию *неветвящейся программы* (straight-line program). В теоретических работах по компьютерной арифметике вычисления нередко моделируются на (многоленточных) *машинах Тьюринга*, принцип работы которых заключается в считывании и перезаписывании информации на ленте при помощи подвижной головки, функционирующей как автомат. На практике машины Тьюринга в чистом виде не используются.

Для исследования методов умножения многочленов над некоторым кольцом K естественно воспользоваться моделью *арифметических схем*, иначе называемыми *схемами над кольцом* K. В отличие от булевых схем арифметические схемы строятся из элементов сложения, умножения и умножения на константы кольца K.

Более подробное разъяснение вычислительных моделей и основных понятий теории сложности можно найти, например, в [1, 12, 25, 53, 86].

Далее $M(n)$ обозначает сложность умножения n-разрядных чисел при реализации булевыми схемами (или неветвящимися программами). $M_K(n)$ обозначает сложность умножения многочленов степени $< n$ над K при реализации схемами над K. M_K означает сложность умножения в кольце K при реализации булевыми схемами.

2. Метод Карацубы

Стандартный (машинный) метод умножения основан на известном школьном приеме: один из множителей последовательно умножается на разряды другого, соответствующим образом сдвинутые результаты записываются друг под другом и затем складываются. Умножение чисел 26 и 21 в этой интерпретации будет выглядеть так:

$$
\begin{array}{cccccccccc}
 & & & & & 1 & 1 & 0 & 1 & 0 \\
 & & & & & 1 & 0 & 1 & 0 & 1 \\
\hline
 & & & & & 1 & 1 & 0 & 1 & 0 \\
 & & & & 0 & 0 & 0 & 0 & 0 & \\
 & & & 1 & 1 & 0 & 1 & 0 & & \\
 & & 0 & 0 & 0 & 0 & 0 & & & \\
 & 1 & 1 & 0 & 1 & 0 & & & & \\
\hline
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

Рис. 1

Сложность умножения n-разрядных чисел этим способом, как легко проверить, составляет по порядку n^2 операций. Если считать аккуратнее, то можно указать оценку сложности $6n^2 - 8n$: схема умножения объединяет n^2 элементов умножения битов и $n - 1$ стандартных сумматоров для вычисления суммы n слагаемых.

Эту оценку можно также вывести рекурсивным способом. Разобьем $2n-$разрядные числа A и B на блоки по n разрядов: $A = A_1 2^n + A_0$, $B = B_1 2^n + B_0$. Теперь умножение исходных чисел выполняется при помощи четырех умножений «половинок» и нескольких сложений по формуле

$$AB = A_1 B_1 2^{2n} + (A_0 B_1 + A_1 B_0) 2^n + A_0 B_0. \tag{1}$$

Размышляя над поставленной А. Н. Колмогоровым задачей о сложности умножения, в 1960 г. А. А. Карацуба (в то время аспирант МГУ) обнаружил более экономную, чем (1), формулу

$$AB = A_1 B_1 2^{2n} + (A_1 B_1 + A_0 B_0 - (A_0 - A_1)(B_0 - B_1)) 2^n + A_0 B_0, \tag{2}$$

которая требует только трех умножений n-разрядных чисел и нескольких сложений-вычитаний. Для сложности метода $K(n)$ получается рекуррентное соотношение:

$$K(2n) \leqslant 3K(n) + O(n),$$

которое разрешается как $K(n) = O\left(n^{\log 3}\right)$. (Здесь и далее по тексту все логарифмы — по основанию 2.)

Подробно об истории появления первого быстрого метода умножения рассказывает сам его автор в [9, 10].

Аналогичный метод и результат имеют место для многочленов. Полиномиальная аналогия позволяет увидеть в основе метода Карацубы идею интерполяции. Так, умножение многочленов степени 1 выполняется в три приема:

i) Вычисление значений многочленов в трех точках[2]: 0, -1, ∞.

ii) Перемножение значений в каждой точке.

iii) Восстановление многочлена, имеющего заданные значения в точках интерполяции, см. формулу (2).

Развитие идеи интерполяции привело к появлению теоретически более быстрых методов умножения.

3. Блочные методы умножения 1960-х

Первый такой метод предложил А. Л. Тоом (тогда студент мехмата МГУ) в 1963 г. [22]. В методе Тоома каждый множитель разбивается на r блоков:

$$A = \sum_{i=0}^{r-1} A_i 2^{iq}, \qquad B = \sum_{i=0}^{r-1} B_i 2^{iq}, \qquad A_i, B_i < 2^q. \tag{3}$$

Произведение чисел восстанавливается из произведения многочленов $A(X) = \sum A_i X^i$ и $B(X) = \sum B_i X^i$ подстановкой $X = 2^q$. Для умножения многочленов выполняется интерполяция в $2r-1$ точках: $-r+1, \ldots, r-1$.

Заметим, что умножения в методе Тоома возникают не только в точках интерполяции, но также и при вычислении значений многочленов $A(X)$ и $B(X)$ в этих точках, и при восстановлении многочлена $A(X)B(X)$ по значениям в точках. В оригинальном методе этап восстановления рассматривается как решение системы линейных уравнений на коэффициенты искомого многочлена.

Метод Тоома практичен при небольших значениях r. При выборе $r = 2$ получается просто вариант метода Карацубы. По сей день для умножения чисел в диапазоне от нескольких десятков до 1000 двоичных разрядов обычно применяются методы Карацубы и троичный ($r = 3$)

[2]Значение многочлена в точке ∞ полагается равным старшему коэффициенту.

вариант метода Тоома. Но для вывода теоретической оценки сложности следует выбирать r растущим вместе с n, оптимальным является выбор $r = 2^{\Theta(\sqrt{\log n})}$.

Авторская оценка сложности метода имеет вид $O\left(n2^{5\sqrt{\log n}}\right)$. Более тщательный анализ, выполненный Д. Кнутом (см. [11]), привел к уточнению оценки до $O\left(n2^{\sqrt{2\log n}}\log n\right)$.

Адаптированный для умножения многочленов вариант метода Тоома (указан С. Куком [40]) предлагает в качестве точек интерполяции выбирать многочлены малых степеней с коэффициентами 0 и 1. Такой выбор универсален: он позволяет умножать и двоичные многочлены. Рекордные оценки сложности умножения двоичных многочленов степени в пределах одной-двух сотен получаются специальным образом оптимизированными блочными методами, см., например, [37, 46, 73].

Если посмотреть шире, в основе метода Тоома можно увидеть идею перехода к модулярному представлению, поскольку интерполяция — это частный случай этой идеи. В модулярном представлении многочлен записывается как набор остатков по модулям некоторых заданных многочленов $f_i(x)$. Метод Тоома—Кука, как изложен выше, использует многочлены $x^q - \alpha_i$, где α_i — это точки: числа или многочлены.

Развивая модулярную идею, А. Шёнхаге в 1966 г. [78] построил подходящую систему модулей вида $2^{q_i} - 1$ и получил еще один метод умножения чисел с оценкой сложности $O\left(n2^{\sqrt{2\log n}}\log^{3/2} n\right)$.

За год до этого появилась статья Дж. Кули и Дж. Тьюки [41], в которой авторы указали быстрый способ вычисления дискретного преобразования Фурье (ДПФ) и вообще обратили внимание на ДПФ как инструмент быстрых вычислений. С тех пор все новые быстрые алгоритмы умножения опираются на ДПФ.

4. Дискретное преобразование Фурье

Привлечение ДПФ к задаче умножения фактически положило конец дискуссии об оптимальном выборе точек для интерполяции. Умножение при помощи ДПФ использует в качестве точек интерполяции группу корней из единицы. Далее приводятся некоторые сведения о ДПФ, необходимые для описания быстрых методов умножения.

Следуя [53], дадим формальное определение. Пусть K — коммутативное, ассоциативное кольцо с единицей. Элемент $\zeta \in K$ называется *примитивным* (*первообразным*) *корнем степени* $N \in \mathbb{N}$, если $\zeta^N = 1$, и никакой из элементов $\zeta^{N/p} - 1$, где p — простой делитель числа N, не является делителем нуля в K. (Напомним, что элемент a называется делителем нуля, если существует ненулевой элемент b, такой, что $ab = 0$.)

Дискретным преобразованием Фурье (*ДПФ*) *порядка* N называется $(K^N \to K^N)$-преобразование

$$\text{ДПФ}_{N,\zeta}[K](\gamma_0, \ldots, \gamma_{N-1}) = (\gamma_0^*, \ldots, \gamma_{N-1}^*), \qquad \gamma_j^* = \sum_{i=0}^{N-1} \gamma_i \zeta^{ij}. \tag{4}$$

где ζ — примитивный корень степени N.

Фундаментальное свойство ДПФ формулируется следующим образом:

Лемма 1. *Пусть элементы* γ_j^* *определяются из* (4). *Тогда*

$$\text{ДПФ}_{N,\zeta^{-1}}[K](\gamma_0^*, \ldots, \gamma_{N-1}^*) = (N\gamma_0, \ldots, N\gamma_{N-1}),$$

где под N *в правой части формулы понимается сумма* N *единиц кольца.*

Как следствие, получаем, что если элемент $N = 1 + \ldots + 1 \in K$ обратим, то определено обратное к ДПФ преобразование

$$\text{ДПФ}^{-1}_{N,\zeta}[K] = N^{-1} \cdot \text{ДПФ}_{N,\zeta^{-1}}[K].$$

4.1. Полиномиальная интерпретация ДПФ

Рассмотрим многочлен $\Gamma(x) = \gamma_0 + \ldots + \gamma_{N-1}x^{N-1}$. Тогда, по определению,

$$\text{ДПФ}_{N,\zeta}[K](\gamma_0, \ldots, \gamma_{N-1}) = \left(\Gamma(\zeta^0), \ldots, \Gamma(\zeta^{N-1})\right),$$

т.е. ДПФ вычисляет значения многочлена $\Gamma(x)$ в точках ζ^i. Смысл обратного преобразования $\text{ДПФ}^{-1}_{N,\zeta}[K]$ заключается в восстановлении коэффициентов единственного многочлена степени $< N$, имеющего заданный набор значений в точках $\zeta^0, \ldots, \zeta^{N-1}$.

Формально, связь между ДПФ и интерполяцией описывается следующей леммой:

Лемма 2. *Преобразование* $\text{ДПФ}_{N,\zeta}[K]$ *задает изоморфизм:* $K[x]/(x^N - 1) \to K^N$.

Рассмотренный изоморфизм приводит к эффективному способу умножения многочленов над K.

Теорема 1. *Пусть в кольце K определено преобразование* $\text{ДПФ}_{N,\zeta}[K]$ *и обратное к нему. Тогда умножение двух многочленов по модулю $x^N - 1$ над K можно выполнить при помощи двух преобразований* $\text{ДПФ}_{N,\zeta}[K]$, *одного* $\text{ДПФ}^{-1}_{N,\zeta}[K]$, *$N$ умножений в K и N умножений на константы $N^{-1} \in K$.*

Если рассматривать многочлены как вектора коэффициентов, то умножение по модулю $x^N - 1$ совпадает с *циклической сверткой порядка N*.

4.2. Вычисление ДПФ

Эффективность ДПФ обеспечивается семейством алгоритмов быстрого его вычисления. Эти алгоритмы получили название быстрого преобразования Фурье (БПФ). Вообще, под алгоритмами БПФ понимаются такие алгоритмы, в которых используется прием сведения ДПФ составного порядка N к ДПФ порядка множителей числа N. Иногда термин БПФ прилагается к любым алгоритмам сложности $O(N \log N)$.

Самым популярным примером БПФ является метод Кули—Тьюки [41], основанный на следующей лемме.

Лемма 3 ([41]). *ДПФ порядка ST реализуется при помощи S ДПФ порядка T, T ДПФ порядка S и $(S-1)(T-1)$ операций умножения на степени ζ — примитивного корня степени ST.*

Доказательство леммы опирается на декомпозицию ДПФ на «внешние» ДПФ порядка S и «внутренние» ДПФ порядка T. При любых $s = 0, \ldots, S-1$ и $t = 0, \ldots, T-1$ справедливо

$$\gamma^*_{sT+t} = \sum_{I=0}^{ST-1} \gamma_I \zeta^{I(sT+t)} = \sum_{i=0}^{T-1}\sum_{j=0}^{S-1} \gamma_{iS+j}\zeta^{(iS+j)(sT+t)} =$$
$$= \sum_{i=0}^{T-1}\sum_{j=0}^{S-1} \gamma_{iS+j}\zeta^{itS+jsT+jt} = \sum_{j=0}^{S-1} (\zeta^T)^{js} \cdot \zeta^{jt} \cdot \gamma_{(j),t}, \quad (5)$$

где

$$\gamma_{(j),t} = \sum_{i=0}^{T-1} \gamma_{iS+j} (\zeta^S)^{it}.$$

Учитывая, что ДПФ порядка 2 выполняется при помощи сложения и вычитания, рекурсивным применением леммы 3 выводится

Лемма 4. *ДПФ порядка 2^k может быть выполнено за $k2^k$ операций сложения-вычитания и $(k-2)2^{k-1}+1$ операций скалярного умножения.*

Эта оценка является асимптотически наилучшей из известных верхних оценок сложности ДПФ порядка 2^k.

Для случая, когда N раскладывается на взаимно простые множители, существует более эффективный алгоритм И. Гуда [54], который не требует дополнительных умножений на степени примитивного корня. Этот алгоритм также называют алгоритмом Гуда—Томаса, т.к. он был переоткрыт Л. Томасом [84] несколько лет спустя после работы Гуда.

Лемма 5 ([54]). *Если* НОД$(S, T) = 1$, *то для вычисления ДПФ порядка ST достаточно выполнить S ДПФ порядка T и T ДПФ порядка S.*

Доказательство опирается на следующую декомпозицию. Пусть m, n — коэффициенты Безу из равенства $nS + mT = 1$. Предварительно для $I = 0, \ldots, ST-1$ обозначим $\gamma_I = \gamma_{i,j}$, где $i = I \bmod T$ и $j = I \bmod S$, а для $K = 0, \ldots, ST-1$ обозначим $\gamma^*_K = \gamma^*_{s,t}$, где $s = mK \bmod S$ и $t = nK \bmod T$. Заметим, что $I = (inS + jmT) \bmod ST$ и $K = (sT + tS) \bmod ST$. Тогда при любом K справедливо:

$$\gamma^*_K = \gamma^*_{s,t} = \sum_{I=0}^{ST-1} \gamma_I \zeta^{IK} = \sum_{j=0}^{S-1} \sum_{i=0}^{T-1} \gamma_{i,j} \zeta^{(inS+jmT)(sT+tS)} =$$

$$= \sum_{j=0}^{S-1} \sum_{i=0}^{T-1} \gamma_{i,j} \zeta^{(intS^2+jmsT^2)} = \sum_{j=0}^{S-1} \sum_{i=0}^{T-1} \gamma_{i,j} (\zeta^S)^{it} (\zeta^T)^{js} = \sum_{j=0}^{S-1} (\zeta^T)^{js} \gamma_{(j),t}, \quad (6)$$

где

$$\gamma_{(j),t} = \sum_{i=0}^{T-1} \gamma_{i,j} (\zeta^S)^{it}.$$

Сравнивая правые части (5) и (6), видим, что умножения на степени ζ^{jt} во втором случае удается избежать.

В случае, когда число N простое или когда оно не имеет малых делителей, ДПФ порядка N можно свести к свертке методом Л. Блюстайна [34] или методом Ч. Рейдера [76].

Метод Блюстайна требует наличия корня из единицы степени $2N$ в кольце K и состоит в следующем. Пусть, как и выше, ζ — примитивный корень степени N. Положим $\beta = \zeta^{(N-1)/2} \in K$. Заметим, что $\beta^{-2} = \zeta$. Тогда

$$\gamma^*_j = \sum_{i=0}^{N-1} \gamma_i \zeta^{ij} = \beta^{-j^2} \sum_{i=0}^{N-1} \beta^{(j-i)^2} \beta^{-i^2} \gamma_i = \beta^{-j^2} \sum_{i+k \equiv j \bmod N} U_k V_i = \beta^{-j^2} C_j,$$

где $U_k = \beta^{k^2}$, $V_i = \beta^{-i^2}\gamma_i$. Если определить многочлены $u(x) = \sum U_i x^i$, $v(x) = \sum V_i x^i$, $c(x) = \sum C_i x^i$, то $c(x) = u(x)v(x) \bmod (x^N - 1)$. Фактически доказана

Лемма 6 ([34]). *ДПФ порядка N может быть выполнено посредством циклической свертки порядка N и $2N-2$ умножений на корни из единицы кольца K.*

Альтернативой методу Блюстайна служит метод Рейдера. Он применим в случае простого N, но не предъявляет к кольцу дополнительных требований. Сохраним обозначение ζ для примитивного корня степени N в кольце K. Пусть $g \bmod N$ — порождающий элемент мультипликативной группы поля $GF(N)$. Для $k = 1, \ldots, N-1$ обозначим $\gamma_{[k]} = \gamma_{g^k \bmod N}$, $\gamma^*_{[k]} = \gamma^*_{g^k \bmod N}$, а также $\zeta^{[k]} = \zeta^{g^k}$ при любых k. Тогда

$$\gamma^*_{[j]} = \gamma_0 + \sum_{i=1}^{N-1} \gamma_{[i]} \zeta^{[i+j]}.$$

Суммы в правой части формулы являются компонентами циклической свертки векторов $\left(\gamma_{[1]}, \ldots, \gamma_{[N-1]}\right)$ и $\left(\zeta^{[N-1]}, \ldots, \zeta^{[1]}\right)$. Компонента γ^*_0 вычисляется отдельно как $\sum \gamma_i$. Таким образом, справедлива

Лемма 7 ([76]). *ДПФ простого порядка N может быть выполнено посредством циклической свертки порядка $N-1$ и $2N-2$ сложений в кольце K.*

Метод Рейдера обобщается на случай нечетных примарных чисел $N = p^n$ (p — простое число). В этом случае ДПФ сводится к циклической свертке порядка $(p-1)p^{n-1}$, см., например, [2, 14].

Еще один полезный прием предложили Р. Крэндалл и Б. Фейгин [42]. Он в некоторых случаях позволяет свести ДПФ «неудобного» порядка к ДПФ «удобного» порядка. Например, умножение по модулю простого числа Мерсенна $2^p - 1$ совпадает с циклической сверткой порядка p векторов двоичной записи перемножаемых чисел. Поэтому оно реализуется при помощи ДПФ порядка p с примитивным корнем $2 \in \mathbb{Z}_{2^p-1}$. Но p — тоже простое число, и ДПФ такого порядка, как правило, реализуется не очень эффективно.

Однако, задачу можно свести к приближенному вычислению вещественного ДПФ произвольного порядка N. Разобьем перемножаемое число $X = [x_{p-1}, \ldots, x_0]$ на N блоков приблизительно равной длины: $X = [X_{N-1}, \ldots, X_0]$. Пусть B_i — позиция начала i-го блока. Запишем X в виде:

$$X = \sum_{i=0}^{N-1} X_i \cdot 2^{B_i} = \sum_{i=0}^{N-1} \left(X_i \cdot 2^{B_i - ip/N}\right) 2^{ip/N} = \sum_{i=0}^{N-1} X'_i \cdot 2^{ip/N}.$$

Теперь умножение X на $Y = \sum_{i=0}^{N-1} Y'_i \cdot 2^{ip/N}$ можно выполнить при помощи двух ДПФ порядка N с примитивным корнем $2^{p/N} \in (\mathbb{R} \bmod 2^p - 1)$ и одного обратного ДПФ. Из вектора результата $[Z'_{N-1}, \ldots, Z'_0]$ искомое произведение находится как

$$XY = \sum_{i=0}^{N-1} \left(Z'_i \cdot 2^{ip/N - B_i}\right) 2^{B_i} \bmod 2^p - 1.$$

При выборе размера блоков близким к длине машинного слова описанный метод эффективно реализуется на стандартных компьютерах.

4.3. Многомерное ДПФ

Понятия ДПФ и алгоритмы БПФ легко распространяются на многомерные пространства. Пусть в кольце K определены ДПФ порядков $n_1, \ldots, n_d$ с примитивными корнями соответственно $\zeta_1, \ldots, \zeta_d$. В полиномиальной интерпретации входом d-мерного преобразования является некоторый многочлен $\Gamma(x_1, \ldots, x_d) \in K[x_1, \ldots, x_d]$, а компонентами ДПФ порядка $n_1 \times \ldots \times n_d$ над K являются всевозможные значения

$$\Gamma\left(\zeta_1^{j_1}, \ldots, \zeta_d^{j_d}\right), \qquad j_1 \in \{0, \ldots, n_1 - 1\}, \ \ldots, \ j_d \in \{0, \ldots, n_d - 1\}.$$

Несложно проверить, что при этом справедлив аналог леммы 2: ДПФ задает изоморфизм колец $K[x_1,\ldots,x_d]/\left((x_1^{n_1}-1)\cdot\ldots\cdot\left(x_d^{n_d}-1\right)\right)$ и $K^{n_1\cdot\ldots\cdot n_d}$. Обратное ДПФ получается из прямого заменой примитивных корней ζ_i на ζ_i^{-1} и домножением на величину $(n_1\cdot\ldots\cdot n_d)^{-1}$ (имеет место аналог леммы 1).

Вычислять многомерное ДПФ проще всего как композицию одномерных ДПФ покоординатно. В случае $d=2$ запишем $\Gamma(x_1,x_2)=\sum_{j=0}^{n_2-1}\Gamma_j(x_1)\cdot x_2^j$. При помощи n_2 ДПФ порядка n_1 найдем всевозможные значения $\Gamma_j(\zeta_1^i)$. Применим ДПФ порядка n_2 к каждому из многочленов $\Gamma_i'(x_2)=\sum_{j=0}^{n_2-1}\Gamma_j(\zeta_1^i)\cdot x_2^j$. В итоге будут получены все значения $\Gamma(\zeta_1^i,\zeta_2^j)$. Обобщение изложенного способа приводит к следующему результату.

Лемма 8. *Пусть $N=n_1\cdot\ldots\cdot n_d$. ДПФ порядка $n_1\times\ldots\times n_d$ можно реализовать посредством N/n_1 ДПФ порядка n_1, N/n_2 ДПФ порядка n_2, ... и N/n_d ДПФ порядка n_d.*

Подробнее о вычислении ДПФ см., например, в [2, 3, 5].

5. Умножение при помощи БПФ

Теорема 1 и лемма 4 показывают, что сложность умножения комплексных многочленов степени $n=2^k$ можно оценить как

$$M_{\mathbb{C}}(n)\leqslant(9+o(1))n\log n. \tag{7}$$

Как заметил Шёнхаге [82], такая же оценка справедлива и для произвольных n: в доказательстве нужно от n перейти к ближайшему сверху числу $n'=2^tr$, где $r=O(\log n)$. На практике предпочтительнее применять ДПФ порядка ближайшей сверху степени двойки $2^{\lceil\log n\rceil}$, усеченное до n компонент, см. [16, 29, 63, 69].

Разумеется, оценка (7) имеет место для умножения над произвольным кольцом, в котором есть примитивные корни из единицы любой степени. Для применения БПФ к умножению над кольцами, не имеющими подходящих корней из единицы, а также для умножения целых чисел потребовались специальные приемы.

Простой метод расширения кольца хорошо работает в случае многочленов с действительными коэффициентами: достаточно перейти к вычислениям над полем $\mathbb{C}$. Более того, поскольку комплексный Фурье-образ вещественного вектора полностью определяется половиной своих компонент (остальные являются комплексно-сопряженными), примерно половину операций можно сэкономить, см., например, [3, 5].

С 1960-х гг. известны алгоритмы вычисления комплексного ДПФ порядка 2^k с вещественной сложностью[3] $\sim 4k2^k$. Впервые такая оценка была получена Р. Явне [88] методом, в котором все умножения выполняются на вещественные константы (real-factor FFT), см. также [3]. Однако бóльшую известность приобрели методы «БПФ с расщепленным основанием» (split-radix FFT, см. [3, 32]) — в них все умножения являются умножениями на корни из единицы, что благоприятно для контроля погрешности вычислений. В 2004 г. Дж. ван Бускирку удалось улучшить оценку сложности до $\sim(34/9)k2^k$ (метод опубликован в [66, 68]) при помощи модификации алгоритма с расщеплением основания. Другие практичные алгоритмы были предложены в [61, 89], а наилучшая, на сегодняшний день, оценка $\sim 3\frac{123}{160}\cdot k2^k$ опубликована в [20].

Таким образом, для сложности умножения вещественных многочленов справедлива оценка $M_{\mathbb{R}}(n)\leqslant\left(11\frac{49}{160}+o(1)\right)n\log n$.

Указанные методы используют $\Theta(k2^k)$ как аддитивных операций, так и скалярных умножений. Однако, известно, что для реализации ДПФ произвольного порядка N достаточно $O(N)$ умножений [87, 62], но соответствующие алгоритмы имеют бóльшую общую сложность.

[3] Символ $\sim$ означает асимптотическое равенство.

Теория умножения вещественных и комплексных многочленов адаптируется к умножению над полями $GF(q)$ и $GF(q^2)$, где $q = 2^p - 1$ — псевдомерсенново простое число. В поле $GF(q^2) \cong GF(q)[x]/(x^2+1)$ имеются примитивные корни порядка 2^k, $k \leqslant p+1$, поэтому для умножения многочленов степени $< 2^k$ возможны быстрые алгоритмы. Подробнее см. в [2, 4].

Сразу после появления алгоритмов БПФ их стали пытаться применять к умножению чисел. Согласно [13], первый быстрый алгоритм умножения чисел, основанный на ДПФ, был построен Н. С. Бахваловым — его метод имел сложность $O(n \log^3 n)$.

Затем, в 1971 г. А. Шёнхаге и Ф. Штрассен [79] опубликовали сразу два быстрых метода умножения. Первый из них эксплуатирует естественную идею перехода к вычислениям над полем комплексных чисел $\mathbb{C}$, которое допускает ДПФ произвольного порядка.

5.1. Метод комплексного приближения

Пусть $2N = 2^n q$. В первом методе Шёнхаге—Штрассена [79] перемножаемые N-разрядные числа разбиваются на блоки длины q, как в (3). Полученные многочлены умножаются при помощи комплексного ДПФ порядка 2^n. Вычисления с комплексными числами выполняются с точностью s знаков после запятой. Параметр s подбирается так, чтобы коэффициенты многочлена-произведения, которые на самом деле являются целыми числами, были найдены с ошибкой $< 1/2$; тогда их можно восстановить путем округления. Несложный анализ (см. [79] или [11]) показывает, что достаточно выбрать $s = 2n + 2q + \log n + O(1)$.

Основная сложность алгоритма сосредоточена в ДПФ: $O(2^n n)$ умножений и сложений-вычитаний $O(s)$-разрядных чисел. Рекуррентное соотношение для сложности метода имеет вид

$$M(N) \leqslant O(2^n n(M(s) + s))$$

и решается как $M(N) = O(N \log N \log\log N \log\log\log N \ldots)$. Такая же оценка сложности (и, вероятно, тем же самым методом) была ранее получена А. А. Карацубой [10], но не опубликована.

5.2. Умножение в кольце-расширении

В основе второго метода Шёнхаге—Штрассена [79] фактически лежит сведение умножения многочленов над кольцом K, не имеющим корней из единицы подходящей степени, к умножению в кольце-расширении. А именно, предлагается использовать расширение $K_{2,n}(x) = K[x]/(x^{2^n}+1)$, при этом двойка должна быть обратима в K.

В кольце $K_{2,n}(x)$ определено ДПФ порядка 2^{n+1} с примитивным корнем x (для наглядности, здесь и далее вместо элементов фактор-кольца $K_{2,n}(x)$, которыми являются классы эквивалентных по модулю $x^{2^n}+1$ многочленов, фигурируют многочлены-представители классов).

Из леммы 4 вытекает

Лемма 9. *ДПФ порядка 2^k над кольцом $K_{2,n}(x)$, $k \leqslant n+1$, может быть выполнено за $k2^{k+n}$ аддитивных операций в K.*

Для доказательства достаточно заметить, что умножение на степени примитивного корня x в кольце $K_{2,n}(x)$ реализуется «бесплатно».

Кольцо $K_{2,n}(x)$ можно рассмотреть как расширение аналогичного кольца $K_{2,m}(y)$ меньшей размерности: справедлива

Лемма 10. *Пусть $m < n$. Имеет место изоморфизм*

$$K_{2,n}(x) \cong K_{2,m}(y)[x]/(x^{2^{n-m}} - y),$$

порождаемый подстановкой $x^{2^{n-m}} = y$.

Важно заметить, что указанный изоморфизм реализуется простой перестановкой коэффициентов. Например, многочлену $x^3 + 2x^2 - 1 \in K_{2,2}(x)$ соответствует многочлен $yx + (2y-1) \in K_{2,1}(y)[x]/(x^2 - y)$. Для реализации умножения можно использовать и другие изоморфизмы, см. [32].

В методе Шёнхаге—Штрассена для умножения в кольце $K_{2,n}(x)$ используется изоморфизм леммы 10 с выбором параметра $m = \lceil n/2 \rceil$. При помощи трех ДПФ порядка $2^{n-m+1} = 2^{\lfloor n/2 \rfloor + 1}$ над кольцом $K_{2,m}(y)$ умножение сводится к умножениям в меньшем кольце. Рекурсивное применение этой процедуры приводит к следующему результату.

Теорема 2. *Умножение в кольце $K_{2,n}(x)$ может быть выполнено при помощи $3 \cdot 2^n n(\log_2 n + O(1))$ аддитивных операций и $O(n2^n)$ операций умножения в K.*

Умножение многочленов над K теперь можно выполнить при помощи подходящей схемы умножения в $K_{2,n}(x)$.

Чтобы получить алгоритм умножения чисел, вместо кольца $K_{2,n}(x)$ рассматривается кольцо вычетов $\mathbb{Z}_{F_n}$ по модулю числа Ферма $F_n = 2^{2^n} + 1$. Двойка играет роль переменной x — она является примитивным корнем порядка 2^{n+1} в кольце Ферма.

Перемножаемые числа разбиваются на блоки длины 2^{m-1}, которые интерпретируются как коэффициенты целочисленных многочленов. Произведение чисел восстанавливается из произведения этих многочленов. Основная часть числового метода следует алгоритму теоремы 2. Многочлены перемножаются как многочлены на кольцом $\mathbb{Z}_{F_m}$, и поэтому коэффициенты произведения исходных многочленов вычисляются не точно, а по модулю $F_m = 2^{2^m} + 1$. В действительности, коэффициенты произведения исходных многочленов могут иметь размер порядка $2^{n-m} \cdot 2^{2^m}$. Поэтому в методе [79] дополнительно вычисляется произведение исходных многочленов над кольцом вычетов по модулю $2^{n-m+O(1)}$. Искомые коэффициенты затем восстанавливаются при помощи Китайской теоремы об остатках.

Сложность метода определяется сложностью умножения над кольцом $\mathbb{Z}_{F_m}$, поэтому в результате для нее справедлива по порядку такая же как в теореме 2 оценка $O(n \log n \log\log n)$.

В том же 1971 г. Дж. Поллард [75] предложил идейно близкую, но более простую и практичную версию алгоритма. В ней числа разбиваются на 2^{k-1} блоков длины m, соответствующей длине машинного слова. Числовое произведение восстанавливается из произведения многочленов, выполняемого по нескольким простым модулям p_i, таким, что $\prod p_i \geqslant 2^{2m+k}$. Модули выбираются с условием $p_i \equiv 1 \bmod 2^k$, тогда умножение многочленов выполняется при помощи ДПФ порядка 2^k над $GF(p_i)$.

Исходный метод Шёнхаге—Штрассена не позволяет умножать многочлены над кольцами, в которых необратима двойка, поскольку над такими кольцами не определено ДПФ четного порядка. В работе [80] А. Шёнхаге предложил модифицированный алгоритм умножения для колец, в которых обратима тройка. Для такого кольца K рассматривается расширение $K_{3,n}(x) = K[x]/(x^{2\cdot 3^n} + x^{3^n} + 1)$ — в нем x является примитивным корнем степени 3^{n+1}. Умножение выполняется в духе теоремы 2, но при помощи ДПФ порядка степеней тройки.

Для сложности указанного метода умножения в кольце $K_{3,n}(x)$ можно указать оценку $13 \cdot 3^n n(\log n + O(1))$, которая в случае кольца характеристики 2 может быть понижена до $10 \cdot 3^n n(\log n + O(1))$ операций в K [6].

Стратегию умножения в случае $2^{-1}, 3^{-1} \notin K$ указывает метод Кантора—Калтофена [36]. Способом теоремы 2 и ее троичного аналога, заменяя обратные преобразования $\text{ДПФ}_{N,\zeta}^{-1}$ ненормированными преобразованиями $\text{ДПФ}_{N,\zeta^{-1}}$, вычисляются «почти произведения»

$$2^{N_1} fg = 2^{N_1} fg \bmod (x^{2^{n_1}} + 1), \quad 3^{N_2} fg = 3^{N_2} fg \bmod (x^{2\cdot 3^{n_2}} + x^{3^{n_2}} + 1)$$

при подходящих $n_i, N_i \in \mathbb{N}$, где f, g — перемножаемые многочлены. Окончательно произведение fg можно вычислить как $q2^{N_1} fg + s3^{N_2} fg$, где q, s — коэффициенты Безу из соотношения

$q2^{N_1} + s3^{N_2} = 1$. Впрочем, актуальность разработки быстрых алгоритмов умножения над такими достаточно экзотическими кольцами представляется пока незначительной.

Практичность числового метода Шёнхаге—Штрассена долгое время вызывала сомнения. Тем не менее, метод реализован во многих современных библиотеках быстрой арифметики и обычно вызывается, когда длина перемножаемых чисел превышает несколько тысяч разрядов[4]. Более того, в последнее десятилетие появились приложения, использующие умножение чисел с миллионами знаков и более, в частности, полностью гомоморфное шифрование (FHE). Но вместе с тем возникли и теоретически более быстрые методы умножения.

6. Новейшие методы умножения

Долгое время методы Шёнхаге и Штрассена оставались рекордными, пока в 2007 г. М. Фюрер [49] не предъявил метод умножения n-разрядных чисел сложности $n \log n \cdot 2^{O(\log^* n)}$. В определенном смысле, метод Фюрера получается комбинацией двух методов из [79]. Через год группа индийских математиков [43] предложила модификацию метода Фюрера с аналогичной оценкой сложности.

Любопытно, что Фюрер еще в 1989 г. [48] построил метод, имевший такую сложность в предположении о достаточно высокой плотности распределения простых чисел Ферма — но сегодня более правдоподобной считается гипотеза о конечности множества таких чисел.

В 2014 г. Д. Харви, Ж. ван дер Хувен и Г. Лесер [55, 56] построили семейство алгоритмов умножения как для чисел, так и для многочленов, с теоретической оценкой сложности $O\left(n \log n \cdot 8^{\log^* n}\right)$. В последующих работах [57, 58] оценки сложности были улучшены до $O\left(n \log n \cdot 4^{\log^* n}\right)$.

На одной итерации каждого из новых методов умножения выполняется логарифмическая редукция размера задачи: с n до $2^{\log^{O(1)} \log n}$. Это достигается за счет перехода к вычислениям над кольцом, допускающим компактную кодировку элементов и содержащим примитивные корни высокого и гладкого порядка: например, кольцом комплексных многочленов небольшой степени или конечным полем подходящего размера.

В отличие от предыдущих методов умножения, у новых методов основная сложность сосредоточена в преобразованиях Фурье, а не в умножениях Фурье-образов. Более точно, сложность алгоритмов сконцентрирована в скалярных умножениях, возникающих при разложении БПФ в композицию преобразований малого порядка.

6.1. Метод Фюрера

Рассмотрим кольцо $C_p = \mathbb{C}[x]/(x^{2^p} + 1)$. В нем элемент x является примитивным корнем степени 2^{p+1} из единицы. При этом кольцо содержит примитивные корни из единицы произвольной степени $q2^{p+1}$, которые одновременно являются корнями (степени q) из x.

Один из таких примитивных корней $\rho_q(x)$ степени $q2^{p+1}$ можно построить следующим образом. Обозначим $\zeta = e^{\mathbf{i}\pi/2^p}$ и $\xi = e^{\mathbf{i}\pi/(q2^p)}$. Определим многочлен $\rho_q(x)$ степени $< 2^p$ значениями в точках:

$$\rho_q(\zeta^{2k+1}) = \xi^{2k+1}, \quad k = 0, \ldots, 2^p - 1.$$

Как можно проверить, этот многочлен обладает дополнительным важным свойством: модули коэффициентов его степеней $\rho_q^m(x)$ в кольце C_p не превосходят 1.

Таким образом, в кольце C_p определено ДПФ порядка $N = 2^{s(p+1)}$ с примитивным корнем $\rho_{2^s}(x)$, и при реализации методом Кули—Тьюки оно имеет следующую структуру: s слоев,

[4] Возможно, преодолению скептицизма в отношении метода Шёнхаге—Штрассена способствовала его реализация, выполненная самим Шёнхаге и его учениками на эмуляторе многоленточной машины Тьюринга в начале 1990-х гг. [83].

в которых параллельно выполняется по $N2^{-(p+1)}$ ДПФ порядка 2^{p+1} чередуются со слоями, в которых выполняется по N умножений на степени $\rho_{2^s}(x)$.

Схема состоит из $O(N\log N)$ простых операций — сложений-вычитаний и умножений на степени x (циклических сдвигов), при помощи которых выполняются внутренние ДПФ — и $(s-1)N$ сложных операций: умножений на степени примитивного корня.

Пусть для записи действительных и мнимых частей комплексных коэффициентов многочленов из C_p используется E двоичных разрядов. Тогда сложность умножения $M(N,E)$ многочленов из $C_p[y]$ по модулю y^N-1 можно оценить как

$$M(N,E) \leqslant O(2^p EN\log N) + 3sN\cdot M(1,E).$$

Умножение в кольце C_p естественным образом сводится к умножению $O(2^p(E+p))$-разрядных чисел.

В силу накопления погрешности (абсолютной величины ошибки) по ходу вычислений, точность, с которой вычисляются коэффициенты произведения, вообще говоря, существенно меньше, чем точность исходных коэффициентов. Однако, используя ограниченность коэффициентов $\rho_{2^s}^m(x)$, несложно проверить, что погрешность возрастает в $A^2N^{O(1)}$ раз, где A — максимум абсолютных значений комплексных коэффициентов исходных многочленов.

В методе Фюрера используется выбор параметров[5] $E \asymp \log N$ и $2^p \asymp \log N$. Перемножаемые числа представляются многочленами над C_p степени $< N/2$ следующим образом: они разбиваются на блоки длины $E/3$, которые интерпретируются как целые числа, каждые 2^{p-1} подряд расположенных блоков интерпретируются как коэффициенты многочлена из кольца $\mathbb{C}[x]/(x^{2^p}+1)$. Параметр точности E выбирается так, чтобы можно было восстановить коэффициенты произведения полученных многочленов простым округлением.

Таким образом, для сложности умножения n-разрядных чисел выводится соотношение

$$M(n) \leqslant O(2^p EN\log N) + 3sN\cdot M(O(2^p(E+p))),$$

где $N \asymp n/\log^2 n$, $E \asymp \log n$, $2^p \asymp \log n$, $s \asymp \log n/\log\log n$. Оно разрешается как $M(n) \preceq n\log n\cdot 2^{O(\log^* n)}$.

Метод [43] развивает идею Фюрера, используя вместо C_p подходящие расширения колец вычетов $\mathbb{Z}_{p^c}$.

6.2. Метод Харви—ван дер Хувена—Лесера

Отметим, что ни метод Фюрера [49], ни его p-адичная версия [43] не допускают адаптацию к умножению многочленов. А семейство методов, предложенных в [55, 56], включает как числовые методы умножения, так и методы умножения многочленов. Рассмотрим новый подход на примере умножения двоичных многочленов.

Умножение двоичных многочленов можно интерпретировать как умножение в кольце $GF(2)[x]/(x^n-1)$. Вставкой необходимого числа нулевых коэффициентов такое умножение сводится к умножению в кольце $GF(2^k)[y]/(y^N-1)$, где $kN \asymp n$, $N = N_1\cdot\ldots\cdot N_d \mid 2^k-1$ — гладкое число.

Существование подходящего гладкого числа N гарантируется следующим теоретико-числовым результатом.

Лемма 11 ([27]). *Минимальное число $\lambda(t)$, такое, что*

$$t \leqslant \prod_{p\in\mathbb{P},\ (p-1)\mid\lambda(t)} p,$$

имеет величину $\lambda(t) = (\log t)^{\Theta(\log\log\log t)}$.

[5] Символы $\asymp$ и $\preceq$ означают равенство и неравенство порядков.

Согласно лемме, для заданного порога t найдется число $M \geqslant t$ — произведение простых чисел $2 < p \leqslant \lambda(t) + 1$ со свойством $p - 1 \mid \lambda(t)$. Последнее условие по малой теореме Ферма означает, что $p \mid 2^{\lambda(t)} - 1$. Можно проверить, что если $S > \lambda(t)$, то из M можно выделить произведение $N = N_1 \cdot \ldots \cdot N_d \in [t,\ (\lambda(t) + 1)t]$ взаимно простых множителей, $S \leqslant N_i \leqslant S^3$.

В рассматриваемом методе выбирается $t \asymp n$ и $S = 2^{\Theta(\log^2 \log n)}$. Параметр k выбирается кратным $\lambda(t)$ и имеет размер порядка $(\log n)^{\Theta(\log \log \log n)}$. По построению, множители N_i имеют величину $2^{\Theta(\log^2 \log n)}$.

Далее, умножение в кольце $GF(2^k)[y]/(y^N - 1)$ выполняется при помощи ДПФ порядка N над $GF(2^k)$. ДПФ порядка N методом Кули—Тьюки (лемма 3) или методом Гуда (лемма 5) сводится к последовательному выполнению ДПФ порядков $N_1, \ldots, N_d$. ДПФ порядка N_i методом Блюстайна (лемма 6) сводится к умножению в кольце $GF(2^k)[y]/(y^{N_i} - 1)$.

Такое умножение, в свою очередь, сводится (вставкой нулевых коэффициентов) к умножению в кольце $GF(2)[x]/(x^{n_i} - 1)$, где $n_i \asymp kN_i$. Тем самым задача умножения размерности n сведена к умножениям размерности $n_i = 2^{\Theta(\log^2 \log n)}$.

Используя вытекающую из леммы 5 оценку сложности ДПФ составного порядка N, получаем для сложности $M_{GF(2)}(n)$ умножения в кольце $GF(2)[x]/(x^n - 1)$ рекуррентное соотношение

$$M_{GF(2)}(n) \leqslant N \sum_{i=1}^{d} \frac{M_{GF(2)}(n_i)}{N_i} + 2dNM_{GF(2^k)},$$

где $M_{GF(2^k)}$ — сложность умножения в поле $GF(2^k)$ — ее оценим грубо как $O(k \log k \log \log k)$ (метод Шёнхаге [80]).

Обозначим $\mu(n) = \frac{M_{GF(2)}(n)}{n}$. Тогда с учетом $d \asymp \log n / \log^2 \log n$ имеем (при некоторой постоянной c)

$$\mu(n) \leqslant c \sum_{i=1}^{d} \mu(n_i) + O(d \log k \log \log k) = c \sum_{i=1}^{d} \mu(n_i) + o(\log n).$$

Несложно проверить, что это соотношение разрешается как $\mu(n) = \log n \cdot 2^{O(\log^* n)}$.

Для обоснования оценки $\mu(n) \preceq \log n \cdot 4^{\log^* n}$ в [56, 57] используются несколько дополнительных приемов и более аккуратный анализ. В частности, применяется адаптированный для расширений конечных полей метод Крэндалла—Фейгина [42].

Аналогичным образом можно выполнять умножение над произвольным конечным полем и, привлекая идею Кантора—Калтофена [36] (см. выше), над произвольным кольцом. Аккуратная оценка [57] гласит, что умножение многочленов степени n над произвольным кольцом K выполняется при помощи $O\left(n \log n \cdot 4^{\log^* n}\right)$ аддитивных операций и $O\left(n \cdot 2^{\log^* n}\right)$ мультипликативных операций в K.

Числовые версии [55, 58] метода следуют близкой к приведенной выше общей схеме. Вместо полей $GF(2^k)$ используются или кольца вычетов $\mathbb{Z}_q$ со специальным представлением элементов или поле комплексных чисел, вычисления в котором выполняются с контролируемой точностью, а порядок ДПФ выбирается равным степени двойки.

6.3. Умножение со сложностью $O(n \log n)$

В 2019 г. Д. Харви и Ж. ван дер Хувен [59, 60] построили методы умножения сложности $O(n \log n)$, и вполне возможно, что этот результат по порядку уже неулучшаем. В основе новых методов лежит сведение к многомерной интерполяции.

Действительно, от умножения многочленов степени $< n$ над кольцом K, допускающим лишь БПФ порядка $m \ll n$, несложно путем подстановки Кронекера $x_i = x^{r \cdot 2^i}$ перейти к умножению в кольце многочленов $K[x_1, \ldots, x_d]$. Для умножения в этом кольце доступно d-мерное

БПФ на m^d точках. Но для того, чтобы обратная подстановка позволила восстановить произведение, параметр r следует выбирать чуть меньшим чем $m/2$. Потому и размер перемножаемых многочленов вырастает по меньшей мере в 2^d раз. Для построения теоретически быстрых методов умножения такой рост неприемлем.

Роста размерности можно избежать при переходе к многомерному ДПФ с взаимно простыми компонентами. Действительно, пусть $n = n_1 \cdot \ldots \cdot n_d$, где $n_1, \ldots, n_d$ — попарно взаимно просты. Тогда имеет место изоморфизм $K[x]/\left(x^n - 1\right) \cong K[x_1, \ldots, x_d]/\left(x_1^{n_1} - 1, \ldots, x_d^{n_d} - 1\right)$, порождаемый подстановкой $x_1 = x^{n/n_1}, \ldots, x_d = x^{n/n_d}$. Таким образом, одномерное ДПФ порядка n с точностью до перестановки компонент совпадает с d-мерным ДПФ порядка $n_1 \times \ldots \times n_d$. Эта связь отмечена в работе [28], а метод Гуда—Томаса позволяет свести вычисление такого преобразования к ДПФ порядков n_i. При этом камнем преткновения может стать неэффективность реализации ДПФ с различными порядками n_i.

Такой способ перехода к многомерному ДПФ применяется в методе [59, 60]. Как в методе Фюрера, перемножаемые n-разрядные числа сначала разбиваются на $N = O(n/\log n)$ блоков длины порядка $\log n$, интерпретируемых как комплексные коэффициенты. Далее числа рассматриваются как многочлены над кольцом $C_p = \mathbb{C}[y]/(y^{2^p} + 1)$, где $p = \lceil (1/d) \log N \rceil$. Умножение таких многочленов выполняется как умножение в кольце $C_p[x]/\left(x^{n_1 \cdot \ldots \cdot n_{d-1}} - 1\right)$, где $3N > n_1 \cdot \ldots \cdot n_{d-1} \cdot 2^p > 2N$, и все n_i — различные простые числа. Последнее умножение, как описано выше, сводится к вычислению ДПФ порядка $n_1 \times \ldots \times n_{d-1}$ и умножениям в точках интерполяции. Все вычисления выполняются с запасом точности, который можно определить в $O(\log n)$ разрядов. Пусть для хранения действительных и мнимых частей комплексного коэффициента используется по $E = O(\log n)$ двоичных разрядов.

Для преодоления проблемы неэффективности ДПФ простых порядков n_i авторами [59, 60] предложено два способа. В первом способе в качестве параметров n_i выбираются числа вида $n_i = k_i \cdot 2^p + 1$, где множители k_i не очень велики. Посредством многомерного варианта метода Рейдера ДПФ порядка $n_1 \times \ldots \times n_{d-1}$ сводится к циклической свертке более гладкого порядка $(n_1 - 1) \times \ldots \times (n_{d-1} - 1)$. На этом пути может быть получена оценка сложности умножения $O(n \log n)$ при условии, что k_i можно выбрать действительно малыми, скажем, $k_i < 2^{\varepsilon p}$, где ε — подходящая константа. Но возможность такого выбора опирается на недоказанную гипотезу о размере простых чисел в арифметической прогрессии. Если гипотеза будет подтверждена, то одновременно будет доказана и оценка сложности $O(n \log n)$ для умножения многочленов степени n, см. [60].

Второй способ работает только в числовом случае. Простые числа n_i выбираются чуть меньшими, чем степени двойки $t_i \in \{2^{p-1}, 2^p\}$, так, что $T = t_1 \cdot \ldots \cdot t_{d-1} \leqslant 2N' = 2\, n_1 \cdot \ldots \cdot n_{d-1}$. Закон распределения простых чисел позволяет это сделать. Ключевым моментом метода является сведение ДПФ порядка $n_1 \times \ldots \times n_{d-1}$ к ДПФ порядка $t_1 \times \ldots \times t_{d-1}$: естественно, речь идет о построении приближения с достаточно высокой точностью. Возможность быстрого приближенного вычисления ДПФ одного порядка при помощи ДПФ другого порядка неочевидна и малоизвестна. На материале одномерных преобразований вопрос изучался в работе [45]. Авторы [59] обобщили на многомерный случай и несколько усовершенствовали метод [45]. В результате удалось показать, что при определенных соотношениях между n_i и t_i (числа не должны быть слишком близки) и некоторых других выполнимых условиях указанное сведение может быть выполнено со сложностью $o(n \log n)$ и требуемой точностью.

Напомним, что ДПФ порядка степени двойки t_i над кольцом C_p методом Кули—Тьюки вычисляется за $O(t_i \log t_i)$ операций в C_p, т.е. с общей сложностью $O(t_i \log t_i \cdot 2^p E)$. Тогда лемма 8 позволяет оценить сложность ДПФ порядка $t_1 \times \ldots \times t_{d-1}$ как $O(N' \log N' \cdot 2^p E) = O(n \log n)$. Используя обозначение $M(N, E)$ из §6.1, приходим к соотношению

$$M(n) \leqslant M(N', E) + O(n) \leqslant N' \cdot M(1, E) + O(n \log n) \leqslant N' \cdot M(O(2^p(E + p))) + O(n \log n), \quad (8)$$

оценивая сложность $M(1, E)$ (приближенного) умножения в кольце C_p через $M(O(2^p(E+p)))$,

как и в §6.1. Ввиду $N' = O(n \cdot 2^{-p}/\log n)$ и $p \sim (1/d)\log n$ соотношение (8) разрешается как $M(n) = O(n \log n)$ при достаточно большом постоянном параметре d.

Такова общая схема вычисления. Анализ погрешности и выбор параметра E выполняется примерно так же, как в методе Фюрера. Основная сложность доказательства приходится на построение приближения многомерного ДПФ. Многомерный случай сводится к одномерному, т.е. вычислению ДПФ порядка s через ДПФ порядка $t > s$, где $(s,t) = 1$. Для этого используется матричное тождество $R \cdot \Pi_s \cdot \Phi_s = \Pi_t \cdot \Phi_t \cdot R'$, где Φ_z — матрицы ДПФ порядка z, Π_z — подходящие матрицы перестановок размера $z \times z$, а R и R' — специальные матрицы размера $t \times s$. При фиксированном векторе аргументов $x \in \mathbb{C}^s$ получается переопределенная система линейных уравнений $A \cdot X = b$ относительно вектора неизвестных $X = \Phi_s \cdot x$, где $A = R \cdot \Pi_s$ и $b = \Pi_t \cdot \Phi_t \cdot R' \cdot x$. Построение приближенного решения системы облегчает особенность матриц R и R' — быстрое убывание их коэффициентов при удалении от некоторой диагонали. Подробнее см. в [59].

7. Аддитивное ДПФ и умножение

Методы, основанные на ДПФ, как правило, не позволяют быстро выполнять умножение многочленов над конечными полями, поскольку конечное поле редко поддерживает ДПФ достаточно большого гладкого порядка. Теоретически быстрые методы не работают в практическом диапазоне размерностей. Однако, оказывается, что можно умножать быстро, используя интерполяцию в точках аддитивных подгрупп конечного поля, при этом основанные на этом методы оказываются практичными.

Частный случай аддитивного метода умножения был предложен Я. Ван и Ш. Цу [85] в 1988 г. Через год Д. Кантор [35] построил быстрый метод умножения над полями порядка q^{q^k}. Позже И. фон цур Гатен и Ю. Герхард [52] распространили метод на поля произвольного порядка.

Пусть в поле $GF(q^k)$ выбран базис $\{\alpha_1, \alpha_2, \ldots, \alpha_k\}$. Положим $W_0 = \{0\}$ и обозначим $W_i = \langle \alpha_1, \ldots, \alpha_i \rangle$ — линейную оболочку первых i базисных элементов над полем $GF(q)$. Таким образом, W_i является аддитивной подгруппой. По построению,

$$W_i = \bigcup_{c \in GF(q)} \{c\alpha_i + W_{i-i}\}.$$

Аддитивное ДПФ (АДПФ) порядка q^i определяется как множество значений многочлена $\Gamma(x) \in GF(q^k)[x]$ в точках подгруппы W_i. Обратное АДПФ определяется как набор коэффициентов единственного многочлена степени $q^i - 1$, имеющего заданные значения в точках W_i.

Определим многочлены $S_i(x)$ как

$$S_i(x) = \prod_{\beta \in W_i} (x - \beta). \tag{9}$$

Несложно доказывается

Лемма 12. *Имеют место следующие свойства многочленов $S_i(x)$:*

(i) $S_0(x) \mid S_1(x) \mid \ldots \mid S_k(x) = x^{q^k} - x$;

(ii) $S_{i+1}(x) = S_i^q(x) - S_i^{q-1}(\alpha_{i+1}) S_i(x)$;

(iii) Многочлен $S_i(x)$ является линеаризованным над $GF(q)$, т.е. имеет ненулевые коэффициенты только при степенях x^{q^j}, $0 \leqslant j \leqslant i$.

Для линеаризованных многочленов, в частности, для многочленов $S_i(x)$, выполнены свойства линейности: $S_i(\beta+\gamma) = S_i(\beta) + S_i(\gamma)$ при любых $\beta, \gamma \in GF(q^k)$ и $S_i(c\beta) = cS_i(\beta)$ при любых $\beta \in GF(q^k)$ и $c \in GF(q)$.

Можно проверить [52], что как прямое, так и обратное АДПФ порядка q^i над полем $GF(q^k)$ реализуются схемами над $GF(q^k)$ сложности $O(i^2 q^{i+1})$.

7.1. Быстрое АДПФ

Оценки сложности АДПФ могут быть понижены при специальном выборе базиса. Эффективные базисы для АДПФ над полями $GF\left(q^{q^k}\right)$ предложил Д. Кантор [35]. Элементы базисов Кантора[6] определяются соотношениями $\alpha_1 = 1$ и $\alpha_{i+1}^q - \alpha_{i+1} = \alpha_i$ при $i > 1$. Такой выбор базиса позволил Кантору в случае простого числа q улучшить оценку сложности до $O\left(i^{\log_q \binom{q+1}{2}} q^{i+1}\right)$.

Для двоичных полей, которые представляют основной интерес, Ш. Гао и Т. Матир [51, 69] построили более быстрые алгоритмы. Изложим быстрый алгоритм АДПФ над полем $GF\left(2^{2^k}\right)$.

Базис Кантора в двоичном поле строится по правилам: $\alpha_1 = 1$ и $\alpha_{i+1}^2 + \alpha_{i+1} = \alpha_i$ при $i > 1$. Тогда характеристические многочлены (9) удовлетворяют соотношению $S_{i+1}(x) = S_i^2(x) + S_i(x)$ согласно лемме 12.

Быстрый алгоритм Гао—Матира эффективно использует тот факт, что многие многочлены S_i являются двучленами. А именно, $S_{2^i}(x) = x^{2^{2^i}} + x$ и $W_{2^i} \cong GF\left(2^{2^i}\right)$. Иначе говоря, последовательность $\{W_i\}$ содержит башню полей

$$GF(2) \subset GF\left(2^2\right) \subset \ldots \subset GF\left(2^{2^i}\right) \subset \ldots \subset GF\left(2^{2^k}\right).$$

Чтобы проявить аналогию между быстрым алгоритмом АДПФ и БПФ, следует переписать многочлен-прообраз в специальном базисе.

На множестве многочленов степени $< 2^i$ над $GF(2^k)$ определим базис

$$P_i = \{m_b(x) \mid 0 \leqslant b < 2^i\}$$

так: если в двоичной записи $b = [b_{i-1}, \ldots, b_0]$, то $m_b(x) = \prod_{j=0}^{i-1} S_j^{b_j}(x)$. По построению, $P_0 \subset P_1 \subset \ldots \subset P_k$. Справедлива

Лемма 13 ([69]). *Переход между представлениями многочлена степени $< 2^m$ в стандартном базисе и базисе P_m в любом направлении может быть выполнен схемой над $GF\left(2^{2^k}\right)$ из $\leqslant 2^{m-2} m \log m$ аддитивных операций.*

Алгоритм перехода к базису P_m использует только операции деления с остатком на двучлены, чем обусловлена его малая сложность. Пусть $2^i < m = 2^i + t \leqslant 2^{i+1}$. Задача размера 2^m сводится к 2^{2^i} задачам размера 2^t и 2^t задачам размера 2^{2^i}. Общая схема такова.

1) Сначала многочлен $f(x)$ переписывается в виде

$$\sum_{j=0}^{2^t-1} f_j(x) S_{2^i}^j(x),$$

$\deg f_j < 2^{2^i}$. Это выполняется методом «деления пополам»: сначала многочлен $f(x)$ делим на $S_{2^i}^{2^{t-1}}$, затем частное и остаток — на $S_{2^i}^{2^{t-2}}$, и т.д.

[6] Базис Кантора является частным случаем базиса Артина—Шрайера.

2) Внутренние многочлены $f_j(x)$ переписываются в базисе P_{2^i}.

3) Полагая $y = S_{2^i}(x)$, результат предыдущего шага можно записать как

$$\sum_{b=0}^{2^{2^i}-1} m_b(x) g_b(y),$$

$\deg g_b < 2^t$. Приведем многочлены g_b к представлению в базисе P_t (относительно переменной y). Заметим, что $S_{2^i+l}(x) = S_l(y)$, поэтому переход завершен.

При переходе от специального базиса P_m к стандартному все действия выполняются в обратном порядке.

Доказательство следующей леммы полностью аналогично доказательству леммы 4.

Лемма 14. *Вычисление значений многочлена $f(x) \in GF\left(2^{2^k}\right)[x]$, записанного в базисе P_m, в точках $\omega + W_m$, где $\omega \in GF\left(2^{2^k}\right)$, и обратная задача восстановления многочлена, имеющего заданные значения в указанных точках, выполняется схемой над $GF\left(2^{2^k}\right)$ из $m2^m$ сложений и $m2^{m-1}$ скалярных умножений. В случае $\omega = 0$ можно дополнительно сэкономить по $2^m - 1$ сложений и скалярных умножений.*

На очередном шаге алгоритма многочлен $f(x)$ степени $2^m - 1$ записывается как

$$f_1(x) S_{m-1}(x) + f_0(x).$$

Поэтому

$$f(\omega + W_{m-1}) = f'(\omega + W_{m-1}), \quad f(\omega + \alpha_m + W_{m-1}) = f''(\omega + \alpha_m + W_{m-1}),$$

где $f' = f_1 S_{m-1}(\omega) + f_0$, $f'' = f_1 S_{m-1}(\omega) + f_0 + f_1$.

При выполнении обратной процедуры многочлены f_0, f_1 восстанавливаются по формулам $f_1 = f'' + f'$, $f_0 = f' + S_{m-1}(\omega)(f'' + f')$.

7.2. Умножение при помощи АДПФ

АДПФ позволяет быстро умножать многочлены над полями $GF(q^k)$, если степень произведения меньше q^k. Если порядок поля мал, то вычисления погружаются в поле большего размера.

Например, для умножения двоичных многочленов многочлены разбиваются на блоки длины 2^{k-1}, которые интерпретируются как элементы поля $GF\left(2^{2^k}\right)$ при подходящем k.

В результате умножение многочленов степени N сводится к $\Theta(N)$ умножениям в поле степени $O(\log N)$, поэтому теоретическая сложность метода $O(N \log N \log^{1+o(1)} \log N)$ — несколько выше, чем у метода Шёнхаге—Штрассена или метода [56]. Однако практически метод АДПФ имеет приоритет. Еще и по той причине, что многие умножения в быстром алгоритме АДПФ выполняются на элементы подполей.

В работе [33] показано, что указанный метод позволяет умножать двоичные многочлены степени порядка нескольких сотен быстрее, чем любой другой ранее применявшийся метод.

Совсем недавно [65, 67] было обнаружено, что дополнительный выигрыш (в пределе около двух раз, но практически меньше) удается получить за счет эксплуатации тождества Фробениуса $(a+b)^q = a^q + b^q$, справедливого при любых $a, b \in GF(q)$. В силу этого тождества значение многочлена с коэффициентами из $GF(q)$ в точке $\omega \in GF(q^k)$ автоматически определяет его значения во всех точках ω^{q^l}. Действительно, если $f \in GF(q)[x]$ и $\omega \in GF(q^k)$, то $f\left(\omega^{q^l}\right) = (f(\omega))^{q^l}$.

Автоморфизм Фробениуса $x \to x^{2^i}$ сохраняет аддитивные подгруппы W_j, определенные для базиса Кантора: если $\omega \in W_j$, то $\omega^2 \in W_j$, как следует из определения базиса. Можно проверить, что группа автоморфизмов Фробениуса при действии на элемент $\omega \in W_j \setminus W_{j-1} \in GF\left(2^{2^k}\right)$ порождает класс эквивалентности (орбиту) $\{\omega^{2^i} \mid i = 0, \ldots, 2^k - 1\}$ мощности $2^{\lceil \log j \rceil}$. Как следствие, множество точек АДПФ порядка 2^m над $GF\left(2^{2^k}\right)$ распадается на не более чем $2^m/(m - o(1))$ классов эквивалентности.

Этим наблюдением продиктован следующий метод умножения многочленов суммарной степени менее 2^m. Для вычислений используется поле $GF(2^{2^k})$, $k = \lceil \log m \rceil$. Сначала каждый из перемножаемых многочленов $f, g \in GF(2)[x]$, интерпретируемый как многочлен над $GF\left(2^{2^k}\right)$, приводится к представлению в базисе P_m. Разбиение на блоки и вставка нулевых блоков не требуются. Далее, значения каждого многочлена вычисляются на множестве $\Sigma \subset W_m$ представителей классов эквивалентности. Используется алгоритм леммы 14, адаптированный для многочленов с коэффициентами из подполя $GF(2)$. Обратная процедура восстанавливает произведение по значениям $f(\omega)g(\omega)$, $\omega \in \Sigma$. Подробности см. в [38].

Отметим аналогию с умножением многочленов с вещественными коэффициентами: наличие автоморфизма комплексного сопряжения, сохраняющего подполе $\mathbb{R}$ и множество корней из единицы, позволяет вычислять комплексное ДПФ с вещественным образом или прообразом примерно вдвое быстрее, чем в общем случае.

Идея сокращенного ДПФ над конечным полем, использующего тождество Фробениуса, предложена Ж. ван дер Хувеном и Р. Ларью в [65]. Описанный выше аддитивный аналог метода построен группой математиков из Тайваня в 2018 г. [67]. Полученные с его помощью верхние оценки сложности умножения двоичных многочленов степени от нескольких сотен и выше на момент написания настоящего обзора являются наилучшими из опубликованных.

8. Параллельные схемы для умножения

До сих пор мы качество алгоритмов измеряли общим числом выполняемых операций. Теперь рассмотрим другую важную характеристику — время выполнения алгоритма. В теоретических работах время обычно формализуется понятием глубины.

Глубина — это максимальное число элементов в ориентированной цепи, соединяющей вход и выход схемы. Если сложность соответствует размеру или площади реальной электронной микросхемы, то глубина соответствует быстродействию или времени срабатывания микросхемы. Глубина $D(f)$ функции f (т.е. минимальная глубина схемы, реализующей функцию) характеризует время ее параллельного вычисления. Глубина функции, существенно зависящей от n переменных, в базисе из бинарных операций не может быть меньше $\log n$.

В работе [8] помимо метода Карацубы представлен еще один результат, принадлежащий Ю. П. Офману. Выражаясь современным языком, он говорит о том, что умножение n-разрядных чисел может быть реализовано булевой схемой глубины $O(\log n)$.

Умножение n-коэффициентных многочленов очевидно выполняется с глубиной $\lceil \log n \rceil + 1$ стандартным методом, и эта оценка точная. Вопрос с глубиной умножения чисел обстоит сложнее. В числовом случае не прост даже вопрос о глубине сумматора. Впрочем, В. М. Храпченко [23] разрешил его в асимптотическом смысле, построив сумматор n-разрядных чисел глубины $(1+o(1)) \log n$. Наилучшая, на сегодняшний день, верхняя оценка глубины сумматора имеет вид $\log n + \log\log n + O(1)$ и принадлежит М. И. Гринчуку [7].

В начале 1960-х гг. сразу в нескольких работах, в том числе в [8], было показано, что умножение n-разрядных чисел выполняется с глубиной $O(\log n)$. В основе доказательства лежит сведение к задаче суммирования n бит. Действительно, если в школьном методе умножения (см. рис. 1) выполнить сложения отдельно в каждом столбце и надлежащим образом сгруп-

пировать биты результатов, умножение сведется к суммированию $\log n$ чисел. Можно продолжить в том же духе, сократив число слагаемых до $\log\log n$ и т.д., пока не останется только два слагаемых, чтобы прийти к соотношению

$$D(M_n) \leqslant (D(C_n) + D(C_{\log n}) + \ldots) + D(A_{2n}) = D(C_n) + \log n + o(\log n), \tag{10}$$

где через M_n, A_n и C_n обозначены соответственно операторы умножения и сложения n-разрядных чисел и оператор суммирования n бит. Последний переход выполнен при условии $D(C_n) = o(n)$. В практическом плане описанная стратегия не идеальна, но она позволяет вывести асимптотически наилучшие известные оценки глубины умножения.

Быстрые методы выполнения многократного сложения основаны на идее компрессоров.

Двоичный $(k,l)-$*компрессор*[7] ширины 1 — это схема, реализующая булев оператор $(x_1,\ldots,x_k) \to (y_1,\ldots,y_l)$ по правилу $\sum 2^{a_i}x_i = \sum 2^{b_j}y_j$, где $k > l$, а $a_i, b_j \in \mathbb{Z}$. $(k,l)-$компрессор произвольной ширины строится из параллельных копий компрессоров ширины 1 — он преобразует k многоразрядных чисел в l чисел, сохраняя сумму.

Действительно, k чисел $x^i = [x^i_{n-1},...,x^i_0]$, $1 \leqslant i \leqslant k$, параллельными преобразованиями $(x^1_{j+a_1},\ldots,x^k_{j+a_k}) \to (y^1_{j+b_1},\ldots,y^l_{j+b_l})$, $j \in \mathbb{Z}$, переводятся в l чисел $y^i = [y^i_{n+h},...,y^i_0]$, $1 \leqslant i \leqslant l$, где $h < \log k$, с сохранением суммы (все не определенные разряды в приведенных формулах полагаются равными нулю).

Простейшим компрессором является $(3,2)-$компрессор. Он реализует сумму трех битов по правилу $x_1 + x_2 + x_3 = 2y_2 + y_1$ формулами

$$y_2 = x_1(x_2 \oplus x_3) \oplus x_2x_3, \qquad y_1 = x_1 \oplus x_2 \oplus x_3$$

сложности 5 и глубины 3.

Элементарный способ параллельной реализации оператора C_n — построить дерево $(3,2)$-компрессоров ширины $\log n$. Дерево имеет n входов, два $\log n$-разрядных выхода и $\lceil \log_{3/2}(n/2) \rceil$ уровней компрессоров. Остается присоединенить к выходам $\log n$-разрядный сумматор. Ясно, что глубина полученной схемы равна $O(\log n)$.

Приемы быстрого вычисления C_n при помощи компрессоров совершенствовались в работах В. М. Храпченко [24], М. Патерсона, Н. Пиппенджера, У. Цвика [71, 72] (в этих же работах приводится подробная история вопроса). В частности, в [71, 72] установлен способ оптимального размещения заданных компрессоров в схеме. Рекордные, на сегодняшний день, результаты получены одним из авторов настоящего обзора: конструктивная оценка $D(M_n) \lesssim 4.34 \log n$ [17] и неконструктивная оценка $D(M_n) \lesssim 4.02 \log n$ [19]. В доказательстве этих оценок сумма бит восстанавливается из остатков по модулям 2^k и 3^l, и вычисленного с подходящей точностью приближенного значения.

И прикладной, и теоретический интерес представляет построение параллельных версий быстрых методов умножения, не приводящих к значительному росту сложности. Рассмотренные методы умножения многочленов и так имеют по порядку логарифмическую глубину. Известно по меньшей мере два подхода к параллельному перестроению числовых алгоритмов — оба связаны с применением специального кодирования чисел, в котором аддитивные операции выполняются с глубиной $O(1)$. Первый способ использует запись в знаковой четверичной системе счисления и известен с 1960-х гг. [30], см. также [86]. Во втором способе, предложенном А. В. Чашкиным [25], число u кодируется парой чисел (u_1, u_2), разность которых доставляет истинное значение $u = u_1 - u_2$.

[7] В западной литературе принят термин CSA (carry save adder).

9. Еще о сложности базовых методов умножения

Несмотря на развитие быстрых методов умножения, стандартный метод и его модификации по прежнему широко используются. На практике перемножаемые числа обычно имеют сравнительно небольшой размер, до нескольких сотен разрядов — для таких размеров стандартный метод эффективен.

Итак, суть стандартного метода — в сведении умножения к многократному сложению чисел. Сложность n-кратного сложения n-разрядных чисел имеет порядок n^2, что очевидно, исходя из размерности задачи.

Зная, что $5n-3$ операций необходимо для сложения двух n-разрядных чисел [15], кажется естественным, что сложение m чисел требует примерно $5(m-1)n$ операций. Тем неожиданнее оказался результат, полученный в работе [44] 2010 г. группой математиков из Санкт-Петербургского отделения МИ РАН: указанная оценка сложности m-кратного сложения может быть улучшена уже при $m=3$. Предложенная авторами схема основана на новой экономной конструкции компрессора. Используя эту конструкцию, верхнюю оценку сложности стандартного метода умножения можно уточнить до $5.5n^2-6.5n-1+(n \bmod 2)$ [18].

Для сложности метода Карацубы, в котором многократные сложения-вычитания, диктуемые формулой (2), оптимизированы с учетом результата [44], можно указать оценку $K(2^k) \leqslant 25\frac{83}{405}\cdot 3^k - O(2^k)$ [18].

10. Сложность: квадратичная или линейная

Первоначальное предположение о квадратичной сложности умножения оправдывается в вычислительной модели клеточных схем. Содержательно, плоская клеточная схема — это прямоугольник, составленный из квадратных клеток-элементов. На стороне клетки расположено не более одного входа или выхода, при помощи которых клетки соединяются друг с другом. Например, схема может быть составлена из функциональных, коммутационных и разделительных элементов, изображенных на рис. 2 (f символизирует двуместную, а g — одноместную булеву функцию). Входы и выходы схемы расположены по периметру прямоугольника. Сложность клеточной схемы — число элементов в ней. Подробнее см., например, в [26].

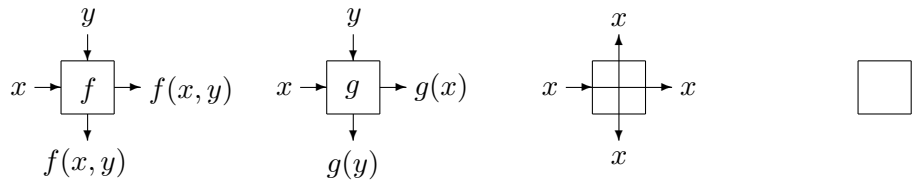

Рис. 2

Н. А. Шкаликова [26] доказала, что порядок сложности умножения n-разрядных чисел при реализации плоскими клеточными схемами равен n^2. То же верно и для трехмерных клеточных схем. Понятно, что хотя клеточные схемы до некоторой степени подобны реальным электронным схемам, относительный размер проводников (коммутационных элементов) в модели несколько завышен. Нижняя оценка сложности умножения как раз и получается из оценки числа проводников, соединяющих фрагменты схемы. Впрочем, в современной схемотехнике плотность размещения проводников и плотность размещения функциональных элементов сближаются.

Напротив, можно указать вычислительные модели, в которых умножение доказуемо выполняется исключительно быстро, не в пример клеточным схемам. Одна из таких моделей —

это RAM-программы (альтернативное обозначение log-RAM). RAM-программа выполняет арифметические действия с аргументами длины $\log n$ за единичное время, где n — размер входа, подробнее см. в [1, 21, 50]. Сложность измеряется суммарным временем выполняемых инструкций. Модель вводится для адекватного отображения процесса вычислений на универсальном процессоре с памятью, хотя, конечно, физический процессор не способен поддерживать единичное время выполнения операций при $n \to \infty$.

А. О. Слисенко [21] и позднее М. Фюрер [50] заметили, что уже первый метод Шёнхаге–Штрассена, основанный на комплексном ДПФ, реализуется RAM-программами линейной сложности. Действительно, метод сводит умножение n-разрядных чисел к ДПФ порядка $\Theta(n/\log n)$ с $O(\log n)$-разрядными аргументами, которое выполняется за $O(n)$ арифметических операций с такими аргументами.

Впрочем, линейная оценка сложности умножения в модели RAM-программ не должна вводить в заблуждение. Модель принципиально допускает вычисления с сублинейной сложностью. Например, сложение n-разрядных чисел реализуется RAM-программами сложности $O(n/\log n)$.

Около 1979 г. А. Шёнхаге [81] получил линейную сложность умножения в более изощренной вычислительной модели — машин с указателями (storage modification machines), см. также [11, 21].

11. Умножение в реальном времени

В некоторых приложениях возникает потребность в алгоритмах умножения реального времени (или online-умножения). Реальное время означает, что k-й разряд или коэффициент сомножителя поступает на вход алгоритма только после того, как вычислен $(k-1)$-й разряд (коэффициент) произведения. Подобная ситуация естественным образом возникает в случае, когда очередные разряды входов зависят от найденных разрядов выхода. Так бывает, в частности, при решении функциональных уравнений итеративной процедурой (см., например, [64]).

Обозначим через $M^{\mathrm{ONLINE}}(n)$ сложность умножения n-разрядных чисел в реальном времени неветвящимися программами[8]. М. Фишер и Л. Стокмайер [47] получили оценку

$$M^{\mathrm{ONLINE}}(2^k) \preceq \sum_{i=0}^{k-1} 2^{k-i} M(2^i) + k2^k$$

через сложность обычного умножения[9]. Из нее в предположении равномерности роста функции $M(n)$ следует $M^{\mathrm{ONLINE}}(n) \preceq M(n)\log n$. Если, например, для внутренних умножений алгоритма использовать метод Карацубы, то полученная online-версия метода Карацубы будет иметь такой же порядок сложности $n^{\log 3}$, как и у исходного метода. Аналогичный результат справедлив для умножения многочленов.

Ж. ван дер Хувен в серии работ (см., например, [64]) показал, что множитель $\log n$ может быть понижен при некоторых дополнительных предположениях о свойствах кольца, над которым выполняется умножение.

Нетривиальная нижняя оценка $M^{\mathrm{ONLINE}}(n) = \Omega(n\log n)$ принадлежит М. Патерсону, М. Фишеру и А. Майеру [70].

[8] Напомним, что неветвящаяся программа — это схема с фиксацией порядка срабатывания элементов.

[9] В работе [47] использовалась вычислительная модель машины Тьюринга, но соотношение справедливо и для неветвящихся программ.

12. Заключение

Итак, на смену пессимистичной гипотезе $M(n) = \Omega(n^2)$ 1950-х гг. поначалу пришли оптимистичные ожидания: в некоторых публикациях ставился вопрос о существовании метода умножения линейной сложности. Сегодня же можно констатировать, что начиная с метода Карацубы и заканчивая методами 2014–2019 гг. магистральная линия развития методов умножения — редукция размера задачи при помощи интерполяции — оставалась неизменной. Этот путь привел к покорению рубежа $O(n \log n)$ в целочисленном варианте задачи; нет сомнений, что в самое ближайшее время аналогичная оценка будет доказана для сложности умножения многочленов. Многие исследователи склоняются к тому, что полученный результат по порядку сложности уже не может быть улучшен.

Впрочем, некоторые надежды связаны с развитием методов умножения линейной мультипликативной сложности. *Мультипликативная сложность* умножения многочленов из $K[x]$ — это число выполняемых при этом нескалярных умножений в K. Вопрос о мультипликативной сложности нетривиален, когда мощность кольца K меньше числа коэффициентов в произведении многочленов, и особенно интересен в экстремальном случае $K = GF(2)$.

Около 1987 г. братья Чудновские [39] предложили теорию для построения методов умножения над любым конечным полем $GF(q)$ с линейной мультипликативной сложностью. Исследования были продолжены в десятках работ других авторов, см., например, обзоры [31, 77]. Все построенные методы используют интерполяцию в группах дивизоров алгебраических кривых и имеют весьма нетривиальное обоснование[10].

Пока основные результаты здесь связаны с уточнением констант в оценках мультипликативной сложности. Например, к сегодняшнему дню доказано, что мультипликативная сложность умножения многочленов степени n над $GF(2)$ не превосходит $(35/3 + o(1))n$ [74]. Но общая сложность соответствующих алгоритмов умножения, по всей видимости, далека от желаемой.

С другой стороны, нелинейная мультипликативная сложность является одним из ограничивающих факторов для известных быстрых алгоритмов умножения, поскольку она определяет размерность линейных преобразований в билинейной структуре алгоритмов. Для сравнения, в методах Шёнхаге и Штрассена выполняется порядка $n \log n$ элементарных нескалярных умножений, в методе [56] — порядка $n \cdot c^{\log^* n}$, а в методе [60] — порядка $n \log^c n$.

СПИСОК ЦИТИРОВАННОЙ ЛИТЕРАТУРЫ

1. Ахо А., Хопкрофт Дж., Ульман Дж. Проектирование и анализ вычислительных алгоритмов. — М.: Мир, 1979.

2. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов. — М.: Мир, 1989.

3. Власенко В. А., Лаппа Ю. М., Ярославский Л. П. Методы синтеза быстрых алгоритмов свертки и спектрального анализа сигналов. — М.: Наука, 1990.

4. Гашков С. Б. Замечания о быстром умножении многочленов, преобразовании Фурье и Хартли // Дискретная математика. 2000. 12, № 3. С. 124–153.

5. Гашков С. Б., Сергеев И. С. Алгоритмы быстрого преобразования Фурье // Дискретная математика и ее приложения. V. — М.: Изд-во ИПМ РАН, 2009. — С. 3–23.

[10] Сложность теории подчеркивает тот факт, что во всех первоначальных и во многих последующих работах по методам умножения линейной мультипликативной сложности обнаружены пробелы в рассуждениях.

6. Гашков С. Б., Сергеев И. С. О сложности и глубине булевых схем для умножения и инвертирования в конечных полях характеристики 2 // Дискретная математика. 2013. 25, № 1. С. 3–32.

7. Гринчук М. И. Уточнение верхней оценки глубины сумматора и компаратора // Дискретный анализ и исследование операций. Серия 1. 2008. 15, № 2. С. 12–22.

8. Карацуба А. А., Офман Ю. П. Умножение многозначных чисел на автоматах // Доклады АН СССР. 1962. 145, № 2. С. 293–294.

9. Карацуба А. А. Сложность вычислений // Труды МИАН. 1995. 211. С. 186–202.

10. Карацуба А. А. Комментарии к моим работам, написанные мной самим // Современные проблемы математики. 2013. 17. С. 7–29.

11. Кнут Д. Искусство программирования. Т. 2. Получисленные алгоритмы. — М.: Вильямс, 2004.

12. Лупанов О. Б. Асимптотические оценки сложности управляющих систем. — М.: Изд. МГУ, 1984.

13. Лупанов О. Б. А. Н. Колмогоров и теория сложности схем // Математические вопросы кибернетики. Вып. 17. — М.: Физматлит, 2008. — С. 5–12.

14. Нуссбаумер Г. Быстрое преобразование Фурье и алгоритмы вычисления сверток. — М.: Радио и связь, 1985.

15. Редькин Н. П. О минимальной реализации двоичного сумматора // Проблемы кибернетики. Вып. 38. — М.: Наука, 1981. — С. 181–216.

16. Сергеев И. С. Регулярные оценки сложности умножения многочленов и усеченного ДПФ // Прикладная дискретная математика. 2011. № 4(14). С. 72–88.

17. Сергеев И. С. Верхние оценки глубины симметрических булевых функций // Вестник Московского университета. Серия 15. Вычислительная математика и кибернетика. 2013. № 4. С. 39–44.

18. Сергеев И. С. О схемной сложности стандартного метода умножения чисел и метода Карацубы // Труды XXII Междунар. науч.-техн. конф. “Информационные средства и технологии”. Том 3. — М.: Изд. дом МЭИ, 2014. — С. 180–187.

19. Сергеев И. С. О сложности и глубине формул для симметрических булевых функций // Вестник Московского университета. Серия 1. Математика. Механика. 2016. № 3. С. 53–57.

20. Сергеев И. С. О вещественной сложности комплексного ДПФ // Проблемы передачи информации. 2017. 53, № 3. С. 90–99.

21. Слисенко А. О. Сложностные задачи теории вычислений // Успехи математических наук. 1981. 36, № 6(222). С. 21–103.

22. Тоом А. Л. О сложности схемы из функциональных элементов, реализующей умножение целых чисел // Доклады АН СССР. 1963. 150, № 3. С. 496–498.

23. Храпченко В. М. Об асимптотической оценке времени сложения параллельного сумматора // Проблемы кибернетики. Вып. 19. — М.: Наука, 1967. — С. 107–120.

24. Храпченко В. М. Некоторые оценки для времени умножения // Проблемы кибернетики. Вып. 33. — М.: Наука, 1978. — С. 221–227.

25. Чашкин А. В. Дискретная математика. — М.: Академия, 2012.

26. Шкаликова Н. А. О сложности реализации некоторых функций клеточными схемами // Сб. работ по матем. кибернетике. Вып. 1. — М.: Изд-во ВЦ АН СССР, 1976. — С. 102–115.

27. Adleman L. M., Pomerance C., Rumely R. S. On distinguishing prime numbers from composite numbers // The Annals of Mathematics, Second Series. 1983. 117, № 1. P. 173–206.

28. Agarwal R. C., Cooley J. W. New algorithms for digital convolution // IEEE Trans. on ASSP. 1977. 25, № 5. P. 392–410.

29. Arnold A. A new truncated Fourier transform algorithm // Proc. Int. Symp. on Symbolic and Algebraic Comp. (Boston, 2013). — NY: ACM, 2013. — P. 15–22.

30. Avizienis A. Signed-digit number representations for fast parallel arithmetic // IRE Trans. on Electr. Computers. 1961. EC10. P. 389–400.

31. Ballet S., Chaumine J., Pieltant J., Rolland R. On the tensor rank of multiplication in finite fields. 2011. arXiv:1107.1184.

32. Bernstein D. J. Fast multiplication and its applications // Algorithmic Number Theory, MSRI Publ. 2008. 44. P. 325–384.

33. Bernstein D. J., Chou T. Faster binary-field multiplication and faster binary-field MACs // Lecture Notes on Comp. Sci. 2014. 8781. P. 92–111.

34. Bluestein L. I. A linear filtering approach to the computation of discrete Fourier transform // ECCC Trans. on Audio and Electroacoustics. 1970. 18, № 4. P. 451–455. [NEREM record. 1968. 10. P. 218–219.]

35. Cantor D. G. On arithmetical algorithms over finite fields // J. Comb. Theory. Series A. 1989. 50. P. 285–300.

36. Cantor D., Kaltofen E. On fast multiplication of polynomials over arbitrary algebras // Acta Inf. 1991. 28, № 7. P. 693–701.

37. Cenk M., Hasan M. A. Some new results on binary polynomial multiplication // J. Cryptographic Engineering. 2015. 5, № 4. P. 289–303.

38. Chen M.-S., Cheng C.-M., Kuo P.-C., Li W.-D., Yang B.-Y. Multiplying boolean polynomials with Frobenius partitions in additive fast Fourier transform. 2018. arXiv:1803.11301.

39. Chudnovsky D. V., Chudnovsky G. V. Algebraic complexities and algebraic curves over finite fields // J. Complexity. 1988. 4. P. 285–316.

40. Cook S. On the minimum computation time of functions. Ph.D. Thesis, Harvard Univ., 1966.

41. Cooley J. W., Tukey J. W. An algorithm for the machine calculation of complex Fourier series // Math. Comp. 1965. 19, № 90. P. 297–301.

42. Crandall R., Fagin B. Discrete weighted transforms and large-integer arithmetic // Math. Comp. 1994. 62, № 205. P. 305–324.

43. De A., Kurur P. P., Saha C., Saptharishi R. Fast integer multiplication using modular arithmetic // SIAM J. Comput. 2013. 42, № 2. P. 685–699.

44. Demenkov E., Kojevnikov A., Kulikov A., Yaroslavtsev G. New upper bounds on the Boolean circuit complexity of symmetric functions // Inf. Proc. Letters. 2010. 110, № 7. P. 264–267.

45. Dutt A., Rokhlin V. Fast Fourier transforms for nonequispaced data // SIAM J. Sci. Comput. 1993. 14, № 6. P. 1368–1393.

46. Find M. G., Peralta R. Better circuits for binary polynomial multiplication // IEEE Trans. on Comp. 2019. 68, № 4. P. 624–630.

47. Fischer M. J., Stockmeyer L. Fast on-line integer multiplication // J. Comput. and System Sci. 1974. 9. P. 317–331.

48. Fürer M. On the complexity of integer multiplication (extended abstract). Tech. Report CS-89-17. Pennsylvania State Univ., 1989.

49. Fürer M. Faster integer multiplication // SIAM J. Comput. 2009. 39, № 3. P. 979–1005.

50. Fürer M. How fast can we multiply large integers on an actual computer? // Proc. Latin Amer. Symp. on Theor. Inf. (Montevideo, 2014). Lecture Notes on Comp. Sci. 2014. 8392. P. 660–670.

51. Gao S., Mateer T. Additive fast Fourier transforms over finite fields // IEEE Trans. on Information Theory. 2010. 56. P. 6265–6272.

52. von zur Gathen J., Gerhard J. Arithmetic and factorization of polynomials over $\mathbb{F}_2$ // Proc. Int. Symp. on Symbolic and Algebraic Comp. (Zürich, 1996). — NY: ACM, 1996. — P. 1–9.

53. von zur Gathen J., Gerhard J. Modern computer algebra. — Cambridge: Cambridge University Press, 1999.

54. Good I. J. The interaction algorithm and practical Fourier analysis // J. R. Statist. Soc. B. 1958. 20, № 2. P. 361–372; 1960. 22. № 2. P. 372–375.

55. Harvey D., van der Hoeven J., Lecerf G. Even faster integer multiplication // J. Complexity. 2016. 36. P. 1–30.

56. Harvey D., van der Hoeven J., Lecerf G. Faster polynomial multiplication over finite fields // J. ACM. 2017. 63, № 6. Article 52.

57. Harvey D., van der Hoeven J. Faster integer and polynomial multiplication using cyclotomic coefficient rings. 2017. arXiv:1712.03693.

58. Harvey D., van der Hoeven J. Faster integer multiplication using short lattice vectors // Proc. 13th Algorithm. Number Theory Symp. (Madison, Wisconsin, 2018) — Berkeley: Math. Sci. Publ., 2019. — P. 293–310.

59. Harvey D., van der Hoeven J. Integer multiplication in time $O(n \log n)$. Tech. report № 02070778. HAL, 2019.

60. Harvey D., van der Hoeven J. Polynomial multiplication over finite fields in time $O(n \log n)$. Tech. report № 02070816. HAL, 2019.

61. Haynal S., Haynal H. Generating and searching families of FFT algorithms // J. Satisf., Boolean Modeling and Comput. 2011. 7, № 4. P. 145–187.

62. Heideman M. T. Applications of multiplicative complexity theory to convolution and the discrete Fourier transform. Ph.D. Thesis. Rice Univ., 1986.

63. van der Hoeven J. The truncated Fourier transform and applications // Proc. Int. Symp. on Symbolic and Algebraic Comput. (Santander, 2004). — NY: ACM, 2004. — P. 290–296.

64. van der Hoeven J. Faster relaxed multiplication // Proc. Int. Symp. on Symbolic and Algebraic Comp. (Kobe, 2014). — NY: ACM, 2014. — P. 405–412.

65. van der Hoeven J., Larrieu R. The Frobenius FFT // Proc. Int. Symp. on Symbolic and Algebraic Comp. (Kaiserslautern, 2017). — NY: ACM, 2017. — P. 437–444.

66. Johnson S. F., Frigo M. A modified split-radix FFT with fewer arithmetic operations // IEEE Trans. Signal Process. 2007. 55, № 1. P. 111–119.

67. Li W.-D., Chen M.-S., Kuo P.-C., Cheng C.-M., Yang B.-Y. Frobenius additive fast Fourier transform // Proc. Int. Symp. on Symbolic and Algebraic Comp. (New York, 2018). — NY: ACM, 2018. — P. 263–270.

68. Lundy T. J., van Buskirk J. A new matrix approach to real FFTs and convolutions of length 2^k // Computing. 2007. 80, № 1. P. 23–45.

69. Mateer T. Fast Fourier transform algorithms with applications. Ph.D. thesis, Clemson Univ., 2008.

70. Paterson M. S., Fischer M. J., Meyer A. R. An improved overlap argument for on-line multiplication. Tech. Report 40. Project MAC. MIT, 1974. [Русский перевод: Патерсон М. С., Фишер М. Дж., Мейер А. Р. Улучшенный метод частичного перекрытия для умножения, выполняемого в темпе поступления информации. Кибернетический сборник. Вып. 14. — М.: Мир, 1977. — С. 78–94.]

71. Paterson M. S., Pippenger N., Zwick U. Optimal carry save networks // LMS Lecture Notes Series. Boolean function complexity. Vol. 169. — Cambridge: Cambridge Univ. Press, 1992. — P. 174–201.

72. Paterson M., Zwick U. Shallow circuits and concise formulae for multiple addition and multiplication // Comput. Complexity. 1993. 3. P. 262–291.

73. De Piccoli A., Visconti A., Rizzo O. G. Polynomial multiplication over binary finite fields: new upper bounds. Cryptology ePrint Archive, 2018/901.

74. Pieltant J., Randriam H. New uniform and asymptotic upper bounds on the tensor rank of multiplication in extensions of finite fields // Math. Comp. 2015. 84. P. 2023–2045.

75. Pollard J. M. The Fast Fourier Transform in a finite field // Math. Comp. 1971. 25, № 114. P. 365–374. [Русский перевод: Поллард Дж. М. Быстрое преобразование Фурье в конечном поле // В кн. Макклеллан Дж. Х., Рейдер Ч. М. Применение теории чисел в цифровой обработке сигналов. — М.: Радио и связь, 1983. — С. 147–155.]

76. Rader C. M. Discrete Fourier transforms when the number of data samples is prime // Proc. IEEE. 1968. 56. P. 1107–1108. [Русский перевод: Рейдер Ч. М. Дискретное преобразование Фурье, когда число отсчетов простое // В кн. Макклеллан Дж. Х., Рейдер Ч. М. Применение теории чисел в цифровой обработке сигналов. — М.: Радио и связь, 1983. — С. 89–91.]

77. Randriambololona H. Bilinear complexity of algebras and the Chudnovsky—Chudnovsky interpolation method // J. Complexity. 2012. 28, № 4. P. 489–517.

78. Schönhage A. Multiplikation großer Zahlen // Computing. 1966. 1, № 3. P. 182–196.

79. Schönhage A., Strassen V. Schnelle Multiplikation großer Zahlen // Computing. 1971. 7, № 3–4. P. 271–282. [Русский перевод: Шёнхаге А., Штрассен В. Быстрое умножение больших чисел. Кибернетический сборник. Вып. 10. — М.: Мир, 1973. — С. 87–98.]

80. Schönhage A. Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2 // Acta Inf. 1977. 7. P. 395–398.

81. Schönhage A. Storage modification machines // SIAM J. Comput. 1980. 9, № 3. P. 490–508.

82. Schönhage A. Asymptotically fast algorithms for the numerical multiplication and division of polynomials with complex coefficients // Proc. Europ. Comput. Algebra Conf. (Marseille, 1982). Lecture Notes on Comp. Sci. 1982. 144. P. 3–15.

83. Schönhage A., Grotefeld A. F. W., Vetter E. Fast algorithms: a multitape Turing machine implementation. — Mannheim, Leipzig, Wien, Zürich: BI-Wissenschaftsverlag, 1994.

84. Thomas L. H. Using a computer to solve problem in Physics // Application in digital computers. — Boston: Ginn and Co., 1963.

85. Wang Y., Zhu X. A fast algorithm for Fourier transform over finite fields and its VLSI implementation // IEEE Journal on Selected Areas in Communications. 1988. 6, № 3. P. 572–577.

86. Wegener I. The complexity of Boolean functions. — Stuttgart: Wiley–Teubner, 1987.

87. Winograd S. On the multiplicative complexity of the discrete Fourier transform // Advances in Math. 1979. 32, № 2. P. 83–117.

88. Yavne R. An economical method for calculating the discrete Fourier transform // Proc. Fall Joint Computer Conf. (San Francisco, 1968). Part I. — NY: ACM, 1968. — P. 115–125.

89. Zheng W., Li K., Li K. Scaled radix-2/8 algorithm for efficient computation of length-$N = 2^m$ DFTs // IEEE Trans. Signal Process. 2014. 62, № 10. P. 2492–2503.

REFERENCES

1. Aho, A. V., Hopcroft, J. E. & Ullman J. D. 1976, The design and analysis of computer algorithms, Addison–Wesley, Reading, Mass.

2. Blahut, R. E. 1985, Fast algorithms for digital signal processing, Addison–Wesley, Reading, Mass.

3. Vlasenko, V. A., Lappa, Yu. M. & Jaroslavsky L. P. 1990, Metody sinteza bystrykh algoritmov svertki i spektral'nogo analiza signalov [Methods to design the fast convolution algorithms and spectral signal analysis], Nauka, Moscow, USSR (in Russian).

4. Gashkov, S. B. 2000, "Remarks on the fast multiplication of polynomials, and Fourier and Hartley transforms", Diskret. mat., vol. 12, no. 3, pp. 124–153 (in Russian); 2000, Discrete Math. and Appl., vol. 10, no. 5, pp. 499–528.

5. Gashkov, S. B. & Sergeev, I. S. 2009, “Fast Fourier transform algorithms”, Diskretnaya matematika i ee prilozheniya [Discrete mathematics and its applications], vol. V, Keldysh Inst. Appl. Math., Moscow, Russia, pp. 3–23 (in Russian).

6. Gashkov, S. B. & Sergeev, I. S. 2013, “On complexity and depth of Boolean circuits for multiplication and inversion over finite fields of characteristic 2”, Diskret. mat., vol. 25, no. 1, pp. 3–32 (in Russian); 2013, Discrete Math. and Appl., vol. 23., no. 1, pp. 1–37.

7. Grinchuk, M. I. 2008, “Sharpening an upper bound on the adder and comparator depths”, Diskret. analiz i issled. operatsii. Ser. 1, vol. 15, no. 2, pp. 12–22 (in Russian); 2009, J. Appl. and Industr. Math., vol. 3, no. 1, pp. 61–67.

8. Karatsuba, A. A. & Ofman, Yu. P. 1962, “Multiplication of multidigit numbers on automata”, Doklady Akad. nauk SSSR, vol. 145, no. 2, pp. 293–294 (in Russian); 1963, Soviet Physics Doklady, vol. 7, pp. 595–596.

9. Karatsuba, A. A. 1995, “The complexity of computations”, Trudy MIAN, vol. 211, pp. 186–202 (in Russian); 1995, Proc. Steklov Inst. Math., vol. 211, pp. 169–183.

10. Karatsuba, A. A. 2013, “Comments to my works, written by myself”, Sovrem. problemy matem., vol. 17, pp. 7–29 (in Russian); 2013, Proc. Steklov Inst. Math., vol. 282, suppl. 1, pp. S1–S23.

11. Knuth, D. E. 1998, The art of computer programming. Vol. 2. Seminumerical algorithms, 3rd ed., Addison–Wesley Longman, Reading, Mass.

12. Lupanov, O. B. 1984, Asimptoticheskie otsenki slozhnosti upravlyayuscshikh sistem [Asymptotic bounds on the complexity of control systems], Mosk. Gos. Univ., Moscow, USSR (in Russian).

13. Lupanov, O. B. 2008, “A. N. Kolmogorov and the circuit complexity theory”, Matematicheskie voprosy kibernetiki [Mathematical issues of cybernetics], vol. 17, Fizmatlit, Moscow, Russia, pp. 5–12 (in Russian).

14. Nussbaumer, H. J. 1982, Fast Fourier transform and convolution algorithms”, 2nd ed., Springer, Berlin–Heidelberg.

15. Red’kin, N. P. 1981, “On the minimal realization of the binary adder”, Problemy kibernetiki [Problems of cybernetics], vol. 38, Nauka, Moscow, USSR, pp. 181–216 (in Russian).

16. Sergeev, I. S. 2011, “Regular estimates for the complexity of polynomial multiplication and truncated Fourier transform”, Prikladnaya diskretnaya matematika [Applied discrete mathematics], no. 4(14), pp. 72–88 (in Russian).

17. Sergeev, I. S. 2013, “Upper bounds on the depth of symmetric Boolean functions”, Vestnik Mosk. univ. Ser. 15. Vychislit. matem. i kibern., no. 4, pp. 39–44 (in Russian); 2013, Moscow Univ. Comput. Math. and Cybernetics, vol. 37, no. 4, pp. 195–201.

18. Sergeev, I. S. 2014, “On the circuit complexity of the standard and the Karatsuba methods of multiplying integers”, Trudy XXII Mezhdunar. nauchn.-tekhn. konf. “Informatsionnye sredstva i tekhnologii” [Proc. 22th Int. Sci.-Tech. Conf. “Information tools and technologies”], vol. 3, Izd. dom MEI, Moscow, Russia, pp. 180–187 (in Russian); 2016, arXiv:1602.02362, available at: http://arxiv.org/abs/1602.02362

19. Sergeev, I. S. 2016, “Complexity and depth of formulas for symmetric Boolean functions”, Vestnik Mosk. univ. Ser. 1. Matem. Mekh., no. 3, pp. 53–57 (in Russian); 2016, Moscow Univ. Math. Bulletin, vol. 71, no. 3, pp. 127–130.

20. Sergeev, I. S. 2017, "On the real complexity of a complex DFT", Problemy peredachi informatsii, vol. 53, no. 3, pp. 90–99 (in Russian); 2017, Problems of Information Transmission, vol. 53, no. 3, pp. 284–293.

21. Slisenko A. O. 1981, "Complexity problems in computational theory", Uspekhi mat. nauk, vol. 36, no. 6(222), pp. 21–103 (in Russian); 1981, Russian Math. Surveys, vol. 36, no. 6, pp. 23–125.

22. Toom, A. L. 1963, "The complexity of a scheme of functional elements realizing the multiplication of integers", Doklady Akad. nauk SSSR, vol. 150, no. 3, pp. 496–498 (in Russian); 1963, Soviet Math. Doklady, vol. 3, pp. 714–716.

23. Khrapchenko, V. M. 1967, "Asymptotic estimation of addition time of a parallel adder", Problemy kibernetiki [Problems of cybernetics], vol. 19, Nauka, Moscow, USSR, pp. 107–120 (in Russian); 1970, Systems Theory Research, vol. 19, pp. 105–122.

24. Khrapchenko, V. M. 1978, "Some estimates for the multiplication time", Problemy kibernetiki [Problems of cybernetics], vol. 33, Nauka, Moscow, USSR, pp. 221–227 (in Russian).

25. Chashkin, A. V. 2012, Diskretnaya matematika [Discrete mathematics], Akademiya, Moscow, Russia, 2012 (in Russian).

26. Shkalikova, N. A. 1976, "On the complexity of realization of some functions by cellular circuits", Sborn. rabot po matem. kibern. [Selected papers on math. cybernetics], vol. 1, Izd. Vychisl. Ts. AN SSSR, Moscow, USSR, pp. 102–115 (in Russian).

27. Adleman, L. M., Pomerance, C. & Rumely, R. S. 1983, "On distinguishing prime numbers from composite numbers", The Annals of Mathematics, Second Series, vol. 117, no. 1, pp. 173–206.

28. Agarwal R. C. & Cooley J. W. 1977, "New algorithms for digital convolution", IEEE Trans. on ASSP, vol. 25, no. 5, pp. 392–410.

29. Arnold, A. 2013, "A new truncated Fourier transform algorithm", Proc. Int. Symp. on Symbolic and Algebraic Comp. (Boston, 2013), ACM, NY, pp. 15–22.

30. Avizienis, A. 1961, "Signed-digit number representations for fast parallel arithmetic", IRE Trans. on Electr. Computers, vol. EC10, pp. 389–400.

31. Ballet, S., Chaumine, J., Pieltant, J. & Rolland, R. 2011. "On the tensor rank of multiplication in finite fields", arXiv:1107.1184, available at: http://arxiv.org/abs/1107.1184

32. Bernstein, D. J. 2008, "Fast multiplication and its applications", Algorithmic Number Theory, MSRI Publ., vol. 44, pp. 325–384.

33. Bernstein, D. J. & Chou, T. 2014, "Faster binary-field multiplication and faster binary-field MACs", Lecture Notes on Comp. Sci., vol. 8781, pp. 92–111.

34. Bluestein, L. I. 1970, "A linear filtering approach to the computation of discrete Fourier transform", ECCC Trans. on Audio and Electroacoustics, vol. 18, no. 4, pp. 451–455; 1968, NEREM record, vol. 10, pp. 218–219.

35. Cantor, D. G. 1989, "On arithmetical algorithms over finite fields", J. Comb. Theory. Series A, vol. 50, pp. 285–300.

36. Cantor, D. & Kaltofen, E. 1991, "On fast multiplication of polynomials over arbitrary algebras", Acta Inf., vol. 28, no. 7, pp. 693–701.

37. Cenk, M. & Hasan, M. A. 2015, “Some new results on binary polynomial multiplication”, J. Cryptographic Engineering, vol. 5, no. 4, pp. 289–303.

38. Chen, M.-S., Cheng, C.-M., Kuo, P.-C., Li, W.-D. & Yang, B.-Y. 2018, “Multiplying boolean polynomials with Frobenius partitions in additive fast Fourier transform”, arXiv:1803.11301, available at: http://arxiv.org/abs/1803.11301

39. Chudnovsky, D. V. & Chudnovsky, G. V. 1988, “Algebraic complexities and algebraic curves over finite fields”, J. Complexity, vol. 4, pp. 285–316.

40. Cook, S. 1966, On the minimum computation time of functions, Ph.D. Thesis, Harvard Univ., Cambridge, Mass.

41. Cooley, J. W. & Tukey J. W. 1965, “An algorithm for the machine calculation of complex Fourier series”, Math. Comp., vol. 19, no. 90, pp. 297–301.

42. Crandall, R. & Fagin, B. 1994, “Discrete weighted transforms and large-integer arithmetic”, Math. Comp., vol. 62, no. 205, pp. 305–324.

43. De, A., Kurur, P. P., Saha, C. & Saptharishi, R. 2013, “Fast integer multiplication using modular arithmetic”, SIAM J. Comput., vol. 42, no. 2, pp. 685–699.

44. Demenkov, E., Kojevnikov, A., Kulikov, A. & Yaroslavtsev G. 2010, “New upper bounds on the Boolean circuit complexity of symmetric functions”, Inf. Proc. Letters, vol. 110, no. 7, pp. 264–267.

45. Dutt, A. & Rokhlin, V. 1993, “Fast Fourier transforms for nonequispaced data”, SIAM J. Sci. Comput., vol. 14, no. 6., pp. 1368–1393.

46. Find, M. G. & Peralta, R. 2019, “Better circuits for binary polynomial multiplication”, IEEE Trans. on Comp., vol. 68, no. 4, pp. 624–630.

47. Fischer, M. J. & Stockmeyer, L. 1974, “Fast on-line integer multiplication”, J. Comput. and System Sci., vol. 9, pp. 317–331.

48. Fürer, M. 1989, On the complexity of integer multiplication (extended abstract), Tech. Report CS-89-17, Pennsylvania State Univ.

49. Fürer, M. 2009, “Faster integer multiplication”, SIAM J. Comput., vol. 39, no. 3, pp. 979–1005.

50. Fürer, M. 2014, “How fast can we multiply large integers on an actual computer?”, Proc. Latin Amer. Symp. on Theor. Inf. (Montevideo, 2014), Lecture Notes on Comp. Sci., vol. 8392, pp. 660–670.

51. Gao, S. & Mateer, T. 2010, “Additive fast Fourier transforms over finite fields”, IEEE Trans. on Information Theory, vol. 56, pp. 6265–6272.

52. von zur Gathen, J. & Gerhard, J. 1996, “Arithmetic and factorization of polynomials over $\mathbb{F}_2$”, Proc. Int. Symp. on Symbolic and Algebraic Comp. (Zürich, 1996), ACM, NY, pp. 1–9.

53. von zur Gathen, J. & Gerhard, J. 1999, Modern computer algebra, Cambridge Univ. Press, Cambridge, 1999.

54. Good, I. J. 1958, “The interaction algorithm and practical Fourier analysis”, J. R. Statist. Soc. B., vol. 20, no. 2, pp. 361–372; 1960, vol. 22, no. 2, pp. 372–375.

55. Harvey, D., van der Hoeven, J. & Lecerf, G. 2016, “Even faster integer multiplication”, J. Complexity, vol. 36, pp. 1–30.

56. Harvey, D., van der Hoeven, J. & Lecerf, G. 2017, “Faster polynomial multiplication over finite fields”, J. ACM, vol. 63, no. 6, Article 52.

57. Harvey, D. & van der Hoeven, J. 2017, “Faster integer and polynomial multiplication using cyclotomic coefficient rings”, arXiv:1712.03693, available at: http://arxiv.org/abs/1712.03693

58. Harvey, D. & van der Hoeven, J. 2019, “Faster integer multiplication using short lattice vectors”, Proc. 13th Algorithm. Number Theory Symp. (Madison, Wisconsin, 2018), Math. Sci. Publ., Berkeley, pp. 293–310.

59. Harvey D. & van der Hoeven J. 2019, “Integer multiplication in time $O(n \log n)$”, HAL Tech. report № 02070778, available at: https://hal.archives-ouverts.fr/hal-02070778

60. Harvey D. & van der Hoeven J. 2019, “Polynomial multiplication over finite fields in time $O(n \log n)$”, HAL Tech. report № 02070816, available at: https://hal.archives-ouverts.fr/hal-02070816

61. Haynal, S. & Haynal, H. 2011, “Generating and searching families of FFT algorithms”, J. Satisf., Boolean Modeling and Comput., vol. 7, no.4, pp. 145–187.

62. Heideman, M. T. 1986, Applications of multiplicative complexity theory to convolution and the discrete Fourier transform, Ph.D. Thesis, Rice Univ., Houston, Texas.

63. van der Hoeven, J. 2004, “The truncated Fourier transform and applications”, Proc. Int. Symp. on Symbolic and Algebraic Comput. (Santander, 2004), ACM, NY, pp. 290–296.

64. van der Hoeven, J. 2014, “Faster relaxed multiplication”, Proc. Int. Symp. on Symbolic and Algebraic Comp. (Kobe, 2014), ACM, NY, pp. 405–412.

65. van der Hoeven, J. & Larrieu, R. 2017, “The Frobenius FFT”, Proc. Int. Symp. on Symbolic and Algebraic Comp. (Kaiserslautern, 2017), ACM, NY, pp. 437–444.

66. Johnson, S. F. & Frigo, M. 2007, “A modified split-radix FFT with fewer arithmetic operations”, IEEE Trans. Signal Process., vol. 55, no. 1, pp. 111–119.

67. Li, W.-D., Chen, M.-S., Kuo, P.-C., Cheng, C.-M. & Yang, B.-Y. 2018, “Frobenius additive fast Fourier transform”, Proc. Int. Symp. on Symbolic and Algebraic Comp. (New York, 2018), ACM, NY, pp. 263–270.

68. Lundy, T. J. & van Buskirk, J. 2007, “A new matrix approach to real FFTs and convolutions of length 2^k”, Computing, vol. 80, no. 1, pp. 23–45.

69. Mateer, T. 2008, Fast Fourier transform algorithms with applications, Ph.D. thesis, Clemson Univ., Clemson, South Car.

70. Paterson, M. S., Fischer, M. J. & Meyer, A. R. 1974, An improved overlap argument for on-line multiplication, Tech. Report 40, Project MAC, MIT, Boston, Mass.

71. Paterson, M. S., Pippenger, N. & Zwick, U. 1992, “Optimal carry save networks”, LMS Lecture Notes Series. Boolean function complexity, vol. 169, Cambridge Univ. Press, Cambridge, UK, pp. 174–201.

72. Paterson, M. & Zwick, U. 1993, “Shallow circuits and concise formulae for multiple addition and multiplication”, Comput. Complexity, vol. 3, pp. 262–291.

73. De Piccoli, A., Visconti, A. & Rizzo, O. G. 2018, “Polynomial multiplication over binary finite fields: new upper bounds”, Cryptology ePrint Archive, 2018/901, available at: https://eprint.iacr.org/2018/091.pdf

74. Pieltant, J. & Randriam, H. 2015, “New uniform and asymptotic upper bounds on the tensor rank of multiplication in extensions of finite fields”, Math. Comp., vol. 84, pp. 2023–2045.

75. Pollard, J. M. 1971, “The Fast Fourier Transform in a finite field”, Math. Comp., vol. 25, no. 114, pp. 365–374.

76. Rader, C. M. 1968, “Discrete Fourier transforms when the number of data samples is prime”, Proc. IEEE, vol. 56, pp. 1107–1108.

77. Randriambololona, H. 2012, “Bilinear complexity of algebras and the Chudnovsky—Chudnovsky interpolation method”, J. Complexity, vol. 28, no. 4, pp. 489–517.

78. Schönhage, A. 1966, “Multiplikation großer Zahlen”, Computing, vol. 1, no. 3, pp. 182–196.

79. Schönhage, A. & Strassen, V. 1971, “Schnelle Multiplikation großer Zahlen”, Computing, vol. 7, no. 3–4, pp. 271–282.

80. Schönhage, A. 1977, “Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2”, Acta Inf., vol. 7, pp. 395–398.

81. Schönhage, A. 1980, “Storage modification machines”, SIAM J. Comput., vol. 9, no. 3, pp. 490–508.

82. Schönhage, A. 1982, “Asymptotically fast algorithms for the numerical multiplication and division of polynomials with complex coefficients”, Proc. Europ. Comput. Algebra Conf. (Marseille, 1982), Lecture Notes on Comp. Sci., vol. 144, pp. 3–15.

83. Schönhage, A., Grotefeld, A. F. W. & Vetter, E. 1994, Fast algorithms: a multitape Turing machine implementation, BI-Wissenschaftsverlag, Mannheim, Leipzig, Wien, Zürich.

84. Thomas, L. H. 1963, “Using a computer to solve problem in Physics”, Application in digital computers, Ginn and Co., Boston, Mass.

85. Wang, Y. & Zhu, X. “A fast algorithm for Fourier transform over finite fields and its VLSI implementation”, IEEE Journal on Selected Areas in Communications, vol. 6, no. 3, pp. 572–577.

86. Wegener, I. 1987, The complexity of Boolean functions, Wiley–Teubner, Stuttgart, FRG.

87. Winograd, S. 1979, “On the multiplicative complexity of the discrete Fourier transform”, Advances in Math., vol. 32, no.2, pp. 83–117.

88. Yavne, R. 1968, “An economical method for calculating the discrete Fourier transform”, Proc. Fall Joint Computer Conf. (San Francisco, 1968). Part I, ACM, NY, pp. 115–125.

89. Zheng, W., Li, K. & Li, K. 2014, “Scaled radix-2/8 algorithm for efficient computation of length-$N = 2^m$ DFTs”, IEEE Trans. Signal Process., vol. 62, no. 10, pp. 2492–2503.

Получено 5.03.2019 г.
Принято в печать 20.03.2020 г.