Gauss multiplication trick?

А. Шень

Владимиру Андреевичу Успенскому

§ 1. Недоразумение

В 2014 году в издательстве МЦНМО вышел русский перевод книги [6]; переводил её Александр Куликов (ПОМИ РАН, Петербург), а я был редактором перевода. В этой книге в качестве одного из первых примеров быстрых алгоритмов излагается алгоритм быстрого умножения многозначных чисел. Алгоритм этот сводит умножение $2n$-значных чисел к умножению n-значных. Пусть нам надо умножить два числа из $2n$ битов (мы рассматриваем двоичные числа, но это не принципиально). Разобьём двоичные записи $2n$-битовых сомножителей x и y на две половины по n знаков:

$$x = 2^n x_1 + x_0, \quad y = 2^n y_1 + y_0$$

(сдвиг на n двоичных разрядов соответствует умножению на 2^n; все четыре половинки x_1, x_0, y_1, y_0 содержат по n битов каждая). Тогда

$$xy = 2^{2n} x_1 y_1 + 2^n (x_0 y_1 + y_0 x_1) + x_0 y_0.$$

Мы свели нашу задачу к умножению четырёх пар n-битовых чисел $x_1 y_1$, $x_0 y_1$, $y_0 x_1$, $x_0 y_0$ — к четырём задачам половинного размера (умножение на степени двойки, т. е. дописывание нулей, а также операции сложения мы не считаем, так как они проще — сравните сложение и умножение столбиком). Повторяя это сведéние ещё раз, мы получим на следующем шаге 16 задач вчетверо меньшего размера и так далее, пока не придём к однозначным числам, где задача тривиальна.

Несложно понять, что этот подход сам по себе не даёт выигрыша. Если всё подсчитать аккуратно, то общее число операций при умножении n-значных чисел будет расти пропорционально n^2 — ровно так же, как при обычном алгоритме умножения столбиком, где мы умножаем

каждый из n разрядов первого числа на каждый из n разрядов второго. (Увеличение размера вдвое соответствует увеличению числа операций при умножении столбиком вчетверо.)

Но мы получим более быстрый алгоритм, если заметим, что можно обойтись *тремя* умножениями n-битовых чисел вместо четырёх. А именно, если за одно умножение вычислить $(x_1+x_0)(y_1+y_0)$, то потом можно вычесть x_1y_1 и x_0y_0, которые всё равно нужно вычислять, и получить сразу сумму $x_0y_1+y_0x_1$, не вычисляя каждое из произведений по отдельности. Делая так на каждом шаге рекурсии, мы достигаем существенной экономии: вместо n^2 получается

$$n^{\log_2 3} \approx n^{1{,}59},$$

что заметно даже для не очень больших значений n. Этот более быстрый алгоритм является стандартным алгоритмом умножения в библиотеках работы с многозначными числами[1)].

Его[2)] придумал Анатолий Алексеевич Карацуба (1937–2008) осенью 1960 года; алгоритм был опубликован в 1962 году в [22]. Подробности этой истории рассказал сам автор в [20]:

> Осенью 1960 г. в Московском университете на механико-математическом факультете начал работать семинар по математическим вопросам кибернетики под руководством А. Н. Колмогорова, где А. Н. Колмогоровым была сформулирована гипотеза n^2 ⟨про порядок роста числа битовых операций, необходимых при умножении n-значных чисел[3)]⟩ и поставлен ряд задач об оценке сложности решений линейных систем уравнений и других сходных вычислений. Я активно стал размышлять над гипотезой n^2 и ровно через неделю обнаружил, что алгоритм, которым я надеялся получить нижнюю ⟨так в статье⟩ оценку величины $M(n)$, даёт оценку вида
>
> $$M(n) = O(n^{\log_2 3}), \quad \log_2 3 = 1{,}5849\ldots$$

[1)] Впоследствии были придуманы и другие алгоритмы, более быстрые (асимптотически, т. е. для достаточно длинных чисел); о некоторых из них можно прочитать в [19].

[2)] На самом деле в статье [22] рассматривается задача о возведении в квадрат многозначных чисел — частный случай умножения, к которому легко сводится общий с помощью формулы

$$ab = \frac{(a+b)^2 - a^2 - b^2}{2}.$$

Для этого частного случая сведение к трём задачам меньшего размера выглядит немного иначе: чтобы вычислить $(2^n a + b)^2 = 2^{2n}a^2 + 2^n \cdot 2ab + b^2$, достаточно возвести a и b в квадрат и потом найти $2ab = (a+b)^2 - a^2 - b^2$, сделав ещё одно возведение в квадрат.

[3)] Здесь и далее угловыми скобками отмечены комментарии, не являющиеся частью цитаты.

> После очередного заседания семинара я сообщил А. Н. Колмогорову о новом алгоритме умножения и об опровержении гипотезы n^2. Это сильно взволновало А. Н. Колмогорова, так как противоречило его довольно правдоподобной гипотезе. На следующем заседании семинара мой метод умножения был рассказан самим А. Н. Колмогоровым, и на этом семинар прекратил свою работу. Позднее, в 1962 г., А. Н. Колмогоров написал (может быть, при участии Ю. П. Офмана[4]) небольшую статью и опубликовал её в Докладах АН СССР. Статья называлась так: А. Карацуба, Ю. Офман, Умножение многозначных чисел на автоматах (ДАН СССР, 1962, т. 145, № 2, с. 293–294). Об этой статье я узнал только тогда, когда мне были даны её оттиски. Необычность способа публикации подчёркивается и тем, что обе статьи [5] и [8] ⟨соответственно [24] и [22] в нашем списке⟩ представлены А. Н. Колмогоровым к опубликованию одновременно 13.II.1962.

Удивительным образом (как заметил Куликов) имя Карацубы вовсе не упоминалось в переводимой книге, а идея о возможности сокращения числа умножений с четырёх до трёх приписывалась Гауссу [6, с. 45 (55 в электронном варианте)]:

> The mathematician Carl Friedrich Gauss (1777–1855) once noticed that although the product of two complex numbers
>
> $$(a+bi)(c+di) = ac-bd+(bc+ad)i$$
>
> seems to involve *four* real-number multiplications, it can in fact be done with just *three*: ac, bd, and $(a+b)(c+d)$, since
>
> $$bc+ad = (a+b)(c+d)-ac-bd.$$
>
> ⟨…⟩ this modest improvement becomes very significant *when applied recursively*.

(Здесь речь идёт об умножении комплексных чисел, но разница лишь в знаке перед bd.) Мы с Куликовым подготовили соответствующее примечание: после слов «Время работы соответствующего алгоритма» должна была следовать сноска «Его предложил А. А. Карацуба (ДАН СССР, 1962, т. 145, с. 293–294). — *Прим. перев.*». К сожалению, в процессе подготовки оригинал-макета это добавление было забыто (и теперь должно дожидаться переиздания[5]), когда и если таковое будет), на что обратила внима-

[4] Юрий Петрович Офман (родился в 1939 году), ученик Колмогорова, занимался теорией сложности вычислений и теорией автоматов. Автор книги [25].

[5] Книга переиздана в 2019 г., ошибка исправлена.

ние (в своём письме в издательство, пересланном нам с Куликовым 6 августа 2016 года) дочь А. А. Карацубы, Екатерина Анатольевна Карацуба:

> В изданном вами переводе книги «Алгоритмы» С. Дасгупта, Х. Пападимитриу, У. Вазирани содержится фактическая ошибка. Первый в мире быстрый алгоритм, алгоритм быстрого умножения, открытый А. А. Карацубой (задача была поставлена А. Н. Колмогоровым), в этой книге приписан Гауссу, безо всяких ссылок на какие-либо документальные источники и свидетельства.
>
> Американские авторы и раньше пытались приписать идею метода Карацубы (названную Шёнхаге «дивайд энд конкур» ⟨divide and conquer, англ.⟩) кому-то другому, так же как и создание первого быстрого алгоритма. Однако у них нет и не было никаких подтверждающих их страстное желание приписать этот метод какому-то более для них предпочитаемому автору документов (лишь словесные мифы). Нет ни одного быстрого алгоритма в каком-либо тексте, изданном раньше 1963 г. (А. А. Карацуба создал свой алгоритм и рассказал его на колмогоровском семинаре в 1960 г., издан в журнале «Доклады АН СССР» и переведён на английский в 1962 г., рассказан Колмогоровым на многих международных конференциях, начиная с 1960 г., включая международный математический конгресс в Стокгольме в 1962 г., опубликован в США отдельно по стокгольмской лекции Колмогорова в книге-сборнике лекций конгресса в 1963 г.).
>
> Издавать в России (безо всяких комментариев) книгу, в которой воруется основное российское открытие в области вычислительной математики, алгоритм, внедрённый в виде софт- и хардвэр в основные компьютеры мира — это как-то не очень порядочно. А что вы об этом думаете?
>
> С уважением, Е. А. Карацуба

Мне как редактору перевода оставалось только принести извинения за допущенную оплошность[6)] — но стало интересно: что вообще пишут разные авторы об истории вопроса, были ли у Карацубы предшествен-

[6)] Вот ответное письмо:

Добрый день, Екатерина Анатольевна!

Редактор издательства, Юрий Николаевич Торхов, переслал мне Ваше письмо, и я отвечаю Вам в качестве редактора перевода книги Дасгупта, Пападимитриу и Вазирани «Алгоритмы». Да, разумеется, алгоритм умножения по половинам был предложен А. А. Карацубой, и мы с переводчиком даже подготовили соответствующее примечание: после слов «Время работы соответствующего алгоритма» должна была следовать сноска «Его предложил А. А. Карацуба (ДАН СССР, 1962, т. 145, с. 293–294). — *Прим. перев.*». К сожале-

ники, и обнаружилось много любопытных вещей, иллюстрирующих, помимо прочего, пути распространения информации и ошибок в ней. К сожалению, это «расследование» осталось незаконченным — может быть, кто-то из тех, кто увидит этот текст, сможет довести его до конца.

§ 2. Почему Гаусс?

Переводимая книга не давала никаких ссылок на источники, и я попытался выяснить, что пишут на эту тему другие авторы. Поиск в интернете показывает, что Гаусс как изобретатель способа умножения комплексных чисел с помощью трёх умножений действительных чисел упоминается разными авторами. Например, Мур и Мертенс в своей известной книге [12, с. 37] (2011) пишут:

> The first $O(n^{\log_2 3})$ algorithm for multiplying n-digit integers was found in 1962 by Karatsuba and Ofman [447]. However, the fact that we can reduce the number of multiplications from four to three goes back to Gauss! He noticed that in order to calculate the product of two complex numbers, $(a+bi)(c+di) = (ac-bd)+(ad+bc)i$ we only need three real multiplications, such as ac, bd, and $(a+c)(b+d)$, since we can get the real and imaginary parts by adding and substracting these products. The idea of [447] is then to replace i with $10^{n/2}$, and to apply this trick recursively.

Здесь «[447]» — это ссылка на оригинальную статью [22]; в ней были сформулированы два результата с указанием авторства (Ю. П. Офман

нию, по моей вине в процессе подготовки оригинал-макета это добавление было забыто. Если будет переиздание, эту ошибку мы исправим.

Всего хорошего,

Александр Шень

P. S. Прошу прощения за задержку с ответом: мне стало интересно, какова история вопроса и откуда взялось название «Gauss trick». Не будучи специалистом по истории науки, я тем не менее предпринял некоторые любительские разыскания, но к успеху они не привели: действительно, название Gauss trick во многих местах встречается в применении к умножению комплексных чисел за три вещественных умножения, но авторы не приводят соответствующих ссылок, а приведённая в Википедии ссылка на Кнута (Knuth D., The Art of Computer programming, vol. 2, 1998, pp. 519, 706, см. `https://en.wikipedia.org/wiki/Multiplication_algorithm#Karatsuba_multiplication` по состоянию на 9 июня 2016) вводит в заблуждение: Гаусс вообще в этом томе в связи с быстрым умножением не упоминается, а в связи с обсуждаемым алгоритмом умножения даётся ссылка на работу А. А. Карацубы (естественно). Само умножение комплексных чисел за три вещественных умножения обсуждается, но без упоминания Гаусса. (Конец письма.)

для одного из них, А. А. Карацуба для второго, того са́мого алгоритма быстрого умножения многозначных чисел). Но ссылок на Гаусса снова не приводится. Нет их и в другой публикации, упоминающей Гаусса как предшественника Карацубы [16]:

> Indeed, when it comes to multiplying two numbers, the best (or fastest) way to do it is often far from obvious.
>
> One particularly intriguing and efficient multiplication algorithm was developed in the late 1950s by Anatolii Alexeevich Karatsuba, now at the Steklov Institute of Mathematics in Moscow.
>
> Karatsuba's "divide-and-conquer" multiplication algorithm has its roots in a method that Carl Friedrich Gauss (1777–1855) introduced involving the multiplication of complex numbers.
>
> ⟨объясняется, как сэкономить одно умножение⟩
>
> So, Gauss optimization saves one multiplication out of four.
>
> Karatsuba's divide-and-conquer multiplication algorithm takes advantage of this saving. ⟨...⟩
>
> Karatsuba's insight was to apply Gauss optimization to this divide-conquer-and-glue approach, replacing some multiplications with extra additions. For large numbers, decimal or binary, Karatsuba's algorithm is remarkably efficient.

Ссылка на Гаусса (наряду с корректной ссылкой на алгоритм Карацубы и его первую публикацию) была в английской Википедии, когда я туда посмотрел (январь 2017 — в текущей версии этого нет, поскольку я внёс соответствующие исправления в текст), но с ней совсем странная история. Там говорилось [13]:

> **Gauss's complex multiplication algorithm**
>
> Complex multiplication normally involves four multiplications and two additions.
>
> $$(a + bi)(c + di) = (ac - bd) + (bc + ad)i.$$
>
> ⟨...⟩ By 1805 Gauss had discovered a way of reducing the number of multiplications to three [11].

Ссылка «[11]» гласила: "Knuth, Donald E. (1988), The Art of Computer Programming, volume 2: Seminumerical algorithms, Addison-Wesley, pp. 519, 706", и, судя по описанию, имелась в виду классическая книга Кнута [10]. Но указанный год (1988) не соответствует ни второму изданию (около 1981), ни третьему (около 1998). Об алгоритме Карацубы в книге Кнута говорится следующее (Section 4.3.3, с. 294 третьего издания, во втором издании этот текст есть на с. 278):

***4.3.3. How Fast Can We Multiply?**

⟨...⟩ let us consider the following question: *Does every general computer algorithm for multiplying two n-place numbers require an execution time proportional to n^2, as n increases?*

⟨...⟩ The answer to the question above is, rather surprisingly, "No," and, in fact, it is not very difficult to see why. ⟨...⟩ If we have two $2n$-bit numbers $u = (u_{2n-1}\ldots u_1u_0)_2$ and $v = (v_{2n-1}\ldots v_1v_0)_2$, we can write

$$u = 2^nU_1 + U_0, \quad v = 2^nV_1 + V_0 \tag{1}$$

where $U_1 = (u_{2n-1}\ldots u_n)_2$ is the "most significant half" of the number u and $U_0 = (u_{n-1}\ldots u_0)_2$ is the "least significant half"; similarly $V_1 = (v_{2n-1}\ldots v_n)_2$ and $V_0 = (v_{n-1}\ldots v_0)_2$. Now we have

$$uv = (2^{2n} + 2^n)U_1V_1 + 2^n(U_1 - U_0)(V_0 - V_1) + (2^n + 1)U_0V_0. \tag{2}$$

This formula reduces the problem of multiplying $2n$-bit numbers to three multiplications of n-bit numbers, namely, U_1V_1, $(U_1-U_0)(V_1-V_0)$, and U_0V_0, plus some simple shifting and adding operations.

⟨...⟩ the main advantage of (2) is that we can use it to define a recursive process for multiplication that is significantly faster than the familiar order-n^2 method when n is large: If $T(n)$ is the time required to perform multiplication of n-bit numbers, we have

$$T(2n) \leqslant 3T(n) + cn \tag{3}$$

for some constant c. ⟨...⟩ the running time for multiplication can be reduced from order n^2 to order $n^{\lg 3} \approx n^{1.585}$, so the recursive method is much faster than the traditional method when n is large.

⟨...⟩ (A similar but slightly more complicated method for doing multiplication with running time of order $n^{\lg 3}$ was apparently first suggested by A. Karatsuba in Doklady Akad. Nauk SSSR, **145** (1962), 293–294. ⟨...⟩ Curiously, this idea does not seem to have been discovered before 1962; none of the "calculating prodigies" who have become famous for their ability to multiply large numbers mentally have been reported to use any such method, although formula (2) adapted to decimal notation would seem to lead to a reasonably easy way to multiply eight-digit numbers in one's head.)

Ни о Гауссе, ни об умножении комплексных чисел здесь не говорится[7]. Но на указанных в Википедии страницах (519, 706) умножение комплекс-

[7] Предметный указатель книги Кнута перечисляет упоминания Гаусса на страницах 20, 101, 363, 417, 422, 449, 578, 679, 685, 688, 701. Ни одна из этих страниц не содержит никаких упоминаний приписываемого ему способа умножения комплексных чисел.

ных чисел действительно упоминается. На с. 519 (во втором издании с. 501) имеется упражнение (к разделу 4.6.4):

> **41.** [*22*] Show that real and imaginary parts of $(a+bi)(c+di)$ can be obtained by doing 3 multiplications and 5 additions of real numbers, where two of the additions involve a and b only.

А на с. 706 (раздел «Answer to exercises»; во втором издании на с. 647) даётся ответ к этому упражнению:

> **41.** $a(c+d)-(a+b)d+i(a(c+d)+(b-a)c)$. $\langle\ldots\rangle$ Without the restriction on additions there are other possibilities. For example, the symmetric formula $ac-bd+i((a+b)(c+d)-ac-bd)$ was suggested by Peter Ungar in 1963. $\langle\ldots\rangle$ See I. Munro, *STOC* **3** (1971), 40–44; S. Winograd, *Linear Algebra and its Applications*, **4** (1971), 381–388.

Ссылок на Унгара у Кнута тоже нет (и в любом случае алгоритм Карацубы опубликован раньше 1963 года)[8]. В статьях Винограда и Мунро, ссылки на которые приводит Кнут, доказывается, что меньше трёх умножений не получится. В статье Винограда [18] про это сказано так:

> ABSTRACT
>
> The two main results of this note are:
>
> (i) The minimum number of multiplications required to multiply two 2×2 matrices is seven.
>
> (ii) The minimum number of multiplications/divisions required to multiply two complex numbers is three.
>
> $\langle\ldots\rangle$ we note that it is possible to compute a complex product using only three multiplications. For example,
>
> $$ac-bd=ac-bd,$$
> $$ad+bc=(a+b)(c+d)-ac-bd.$$
>
> So the three products which are formed are ac, bd, $(a+b)(c+d)$.

Ссылок на источник формулы у него нет, как и у Мунро, который пишет [14]:

> The Multiplication of Complex Numbers
>
> If the number of multiplications required for a computation is regarded as a measure of its difficulty and these computations are performed using complex numbers, it is natural to ask how many real multiplications are necessary to evaluate the real and imaginary parts

[8] Анатолий Воробей в июле 2018 года написал Унгару с просьбой прояснить ситуацию, и он в ответном письме объяснил, что действительно предложил такой способ в 1960-х годах (“I did have the idea in the mid-sixties and told a few people at New York University and I think to Vinograd too, but I did not know Knuth credited me with it.”)

of a complex product. The natural way of forming a complex product requires four real multiplications. It may, however, be done in three but not in two multiplications

$$(a+bi)(c+di) = (ac-bd)+(ad+bc)i$$
$$\begin{array}{ccc} a(c+d) & - & d(a+b) = ac-bd \\ (1) & & (2) \\ a(c+d) & + & c(b-a) = ad+bc \end{array}$$

Theorem. The evaluation of the product of two complex numbers requires three real multiplications, even if multiplication by real constants is not counted.

Унгар упоминается и в некоторых других публикациях. Например, в [8] написано:

1. Introduction. How many real multiplications are required to multiply two complex numbers? In view of the familiar identity

$$z = (a+ib)(c+id) = ac-bd+i(ad+bc),$$

the answer may appear to be four. However, it is possible to make do with three multiplications, because

$$z = ac-bd+i[(a+b)(c+d)-ac-bd]. \qquad (1.1)$$

This formula was suggested by Peter Ungar in 1963, according to Knuth [14, p. 647].

Ссылка «[14]» указывает на второе издание книги Кнута (где процитированное выше решение упражнения 41 находится на с. 647), но прямых ссылок на Унгара нет.

Таким образом, появление Гаусса в Википедии остаётся загадочным. Согласно данным с сайта, раздел про умножение комплексных чисел «по Гауссу» появился впервые в версии 22:55, 27 May 2009 (в версии 21:24, 22 May 2009 его ещё не было); соответствующая правка — с той же ссылкой на Кнута, что и сейчас — была внесена участником Dmcq (`https://en.wikipedia.org/wiki/User_talk:Dmcq`; сейчас (2017) по этому адресу указывается “Semi-retired. This user is no longer very active on Wikipedia.”). Ссылка на алгоритм Карацубы там была до этого (с первой же версии статьи, 15 июня 2002).

Похоже, что тем не менее эта запись в Википедии привела к распространению недоразумений. Скажем, в [2] (2014) Гаусс также упоминается со ссылкой на Кнута:

It is well known too, that the complex multiplication can be carried out using only three real multiplications and five real additions, because ⟨...⟩

$$(a+jb)(c+jd) = ac - bd + j[(a+b)(c+d) - ac - bd]. \qquad (4)$$

Expression (4) is well known as Gauss' trick for multiplication of complex numbers [17].

Здесь «[17]» — это второе издание книги Кнута [10].

Это «словесное квипрокво»[9] огорчительно, тем более когда вообще алгоритм Карацубы приписывается Гауссу (как, например, в [17]).

§ 3. Бэббидж, divide et impera

Занимаясь этими разысканиями, я обнаружил (благодаря ссылке в [12]) удивительную цитату из Бэббиджа, разрабатывавшего (в XIX веке!) программируемую вычислительную машину (так и не реализованную «в железе»). Он называл её «Analytical Engine». Говоря об умножении многозначных чисел с её помощью, Бэббидж пишет [1, с. 61; в публикации 1864 года с. 125]:

...Thus if $a \cdot 10^{50} + b$ and $a' \cdot 10^{50} + b'$ are two numbers each of less than a hundred places of figures, then each can be expressed upon two columns of fifty figures, and a, b, a', b' are less than fifty places of figures ⟨...⟩ The product of two such numbers is

$$aa'10^{100} + (ab' + a'b)10^{50} + bb'.$$

This expression contains four pairs of factors, $aa', ab', a'b, bb'$, each factor of which has less than fifty places of figures. Each multiplication can therefore be executed in the Engine. The time, however, of multiplying two numbers, each consisting of any number of digits between fifty and one hundred, will be nearly four times as long as that of two such numbers of less then fifty places of figures ⟨...⟩

Thus it appears that whatever may be the number of digits the Analytical Engine is capable of holding, if it is required to make all the computations with k times that number of digits, then it can be executed by the same Engine, but in an amount of time equal to k^2 the former.

[9] Термин, предложенный в [26, 27] Владимиром Андреевичем Успенским, большим знатоком подобных историй о недоразумениях и ошибках восприятия [28, 29] — он обнаружил, среди прочего, что памятник, воспринимаемый многими парижанами как памятник погибшей принцессе Диане, в действительности не имеет к ней никакого отношения!

Перевод:

> …Таким образом, если два числа $a \cdot 10^{50} + b$ и $a' \cdot 10^{50} + b'$ состоят менее чем из ста цифр каждое, то оба числа могут быть разбиты на две части по пятьдесят цифр ⟨буквально: записаны в двух столбцах из пятидесяти цифр⟩; числа a, b, a', b' содержат до пятидесяти разрядов… Произведение двух таких чисел равно
>
> $$aa'10^{100} + (ab' + a'b)10^{50} + bb'.$$
>
> Это выражение содержит четыре пары сомножителей aa', ab', $a'b$, bb', и каждый сомножитель содержит до пятидесяти цифр. Таким образом, Машина сможет выполнить эти умножения. Однако время умножения двух чисел, содержащих от пятидесяти до ста цифр каждое, будет примерно в четыре раза больше, чем время умножения чисел до пятидесяти цифр…
>
> Получается, что каково бы ни было количество цифр в числах, помещающихся в Аналитическую Машину, при необходимости та же Машина может выполнять вычисления и с числами, в которых в k раз больше цифр, но это потребует в k^2 раз большего времени.

Видно, что хотя у Бэббиджа совершенно ясно изложена схема divide et impera (латинское выражение, по-английски говорят «divide and conquer», по-русски обычно переводят это как «разделяй и властвуй» — в данном случае мы делим числа на две части), но он не подозревает о возможности замены четырёх умножений на три и соответственного сокращения времени вычисления — так что даже если предположить, что в каких-то бумагах Гаусса и была подобная идея, то видно, что распространения она не получила.

Отметим ещё классический пример алгоритма типа «разделяй и властвуй», который тоже был придуман в докомпьютерную эру — алгоритм сортировки слиянием (подлежащие сортировке объекты произвольно делятся на две группы; каждая из групп отдельно сортируется, а потом группы сливаются с сохранением порядка). Как пишет Кнут [11, с. 385],

> The idea of merging goes back to another card-walloping machine, the *collator*, which was a much later ⟨в сравнении с машинами, сортирующими карты сначала по одной колонке, потом по другой (radix-sort)⟩ invention (1938). With its two feeding stations, it could merge two sorted decks of cards into one, in only one pass; the technique for doing this was clearly explained in the first IBM collator manual (April 1939). [See James W. Bruce, *U. S. Patent 2189024* (1940).]

Модель «IBM 77 electric punched card collator», разработанная фирмой IBM в 1937 году, описывается так [9]:

> As a filing machine, the Type 77 fed and compared simultaneously two groups of punched cards: records already in file and records to be filed. These two groups were merged in correct numerical or alphabetical sequence. ⟨...⟩ Introduced in 1937, the IBM 77 collator rented for $80 a month. It was capable of handling 240 cards a minute ⟨...⟩ IBM withdrew the Type 77 from marketing on November 27, 1957.

§ 4. Снова о Гауссе

Вопрос о том, откуда взялась версия о Гауссе и трёх умножениях, так и остаётся непонятным. Можно предположить, что всё-таки в каких-то рукописях Гаусса такое замечание имеется (что, разумеется, никак не отменяет бесспорного приоритета Карацубы по части алгоритма быстрого умножения). Но это сейчас кажется мне малоправдоподобным, поскольку никаких подтверждающих упоминаний найти не удалось. Другой вариант, может быть, более правдоподобный — что это результат смешения двух историй: быстрого умножения и быстрого преобразования Фурье.

Преобразование Фурье (в интересующем нас конечном варианте) можно описать как вычисление n значений многочлена $P(x)$ степени меньше n во всех корнях степени n из единицы. Интерполяционная формула Лагранжа говорит, что имеется взаимно однозначное соответствие между наборами коэффициентов и наборами значений, и алгоритм быстрого преобразования Фурье позволяет вычислить это преобразование (в любую сторону) за $O(n \log n)$ действий. Этот алгоритм тоже основан на сведении задачи к меньшей, если n есть степень двойки. А именно, пусть $n = 2k$ и ζ — корень из единицы, порождающий все остальные. Мы хотим вычислить $P(1)$, $P(\zeta)$, $P(\zeta^2)$, ..., $P(\zeta^{2k-1})$, где $P(z)$ — многочлен степени меньше $2k$. Если сгруппировать в нём чётные и нечётные члены порознь, то получится $P(z) = P_0(z^2) + zP_1(z^2)$, где P_0 и P_1 — многочлены степени меньше k. Таким образом, нам достаточно вычислить значения многочленов P_0 и P_1 в точках

$$1,\ \zeta^2,\ \zeta^4,\ \ldots,\ \zeta^{2k-2},\ \zeta^{2k} = 1,\ \zeta^{2k+2} = \zeta^2,\ \zeta^{2k+4} = \zeta^4,\ \ldots,\ \zeta^{4k-2} = \zeta^{2k-2},$$

которые являются квадратами корней степени $2k$, т. е. в корнях степени k (каждый встречается дважды). Мы свели задачу к двум задачам вдвое меньшего размера и $O(n)$ умножениям и сложениям (нужным для соединения результатов). Рекурсивное применение этого алгоритма даёт оцен-

ку в $O(n \log n)$ арифметических операций для n, являющихся степенями двойки.

Этот алгоритм был опубликован в статье Кули и Тьюки [5] в 1965 году. Он оказался очень важным с практической точки зрения (непосредственным поводом к их работе была компьютерная обработка сигналов, в частности, данных о волнах в земной коре после ядерных испытаний). Вскоре после публикации обнаружилось, что этот алгоритм неоднократно использовался и публиковался и раньше [4]. Более того, впоследствии выяснилось, что по существу эта же идея содержалась (и использовалась) в записях Гаусса, видимо, относящихся к 1805 году и опубликованных в 1866 году — но написанных на современной Гауссу версии латыни, см. [3, 7], и потому мало кому понятных в настоящее время.

Может быть, эти две истории смешались в чьём-то сознании? Тем более что преобразование Фурье оказалось полезным для быстрого умножения многочленов (вычислим значения в корнях из единицы, перемножим их за $O(n)$ действий, а потом сделаем обратное преобразование), что в свою очередь позволило улучшить оценку Карацубы (алгоритм Шёнхаге — Штрассена, 1971: двоичная запись числа по существу есть значение многочлена с коэффициентами 0 и 1 в точке 2 в конечном поле).

В любом случае, хорошо бы по возможности уменьшить путаницу в этом деле, независимо от причины, по которой она возникла…

§ 5. Дополнение

Алексей Устинов, член редколлегии «Математического просвещения», задал вопрос про «трюк Гаусса» на сайте MathOverflow [23]. Отвечая на этот вопрос, Карло Бенакер предпринял библиографические разыскания, которые также не привели к цели (выяснению того, почему описанный способ умножения комплексных чисел приписывают Гауссу) — зато он обнаружил чуть более ранний текст [15], записки лекций Папаконстантину в университете Йорка 2005 года, где этот способ также приписывается Гауссу (но снова без конкретной ссылки).

Список литературы

[1] *Babbage C.* On the principles and development of the calculator and other seminal writings / Ed. by P. Morrison and E. Morrison. Dover publications, 1961. Более ранняя публикация (1864): *Babbage C.* Passages from the life of a philosopher. Longman et al. London, 1974. https://books.google.ru/books?id=Fa1JAAAAMAAJ&pg=PA125 (с. 125).

[2] *Cariow A, Cariowa G.* A Hardware-oriented Algorithm for Complex-Valued Constant Matrix-Vector Multiplication. https://arxiv.org/pdf/1410.6937v1.pdf

[3] *Cooley J. W.* The Re-Discovery of the Fast Fourier Transform Algorithm // Mikrochimica Acta [Wien]. 1987. III. P. 33–45. См. также: *Cooley J. W.* How the FFT Gained Acceptance // HSNC'87 Proceedings of the ACM Conference on History of scientific and numeric computation. Princeton, NJ, USA, May 13–15, 1987. ACM Publishers.

[4] *Cooley J. W., Lewis P. A. W., Welsh P. D.* Historical Notes on the Fast Fourier Transform // IEEE Transactions on Audio and Electroacoustics. 1967. Vol. 15, iss. 2. P. 76–79. DOI:10.1109/TAU.1967.1161903

[5] *Cooley J. W., Tukey J. W.* An Algorithm for the Machine Calculation of Complex Fourier Series // Mathematics of Computation. 1965. Vol. 19, № 90. P. 297–301. https://www.jstor.org/stable/2003354

[6] *Dasgupta S., Papadimitriou C. H., Vazirani U. V.* Algorithms. McGraw-Hill, 2008. Copyright notice on the electronic draft: 2006. Глава 2: https://people.eecs.berkeley.edu/~vazirani/algorithms/chap2.pdf (Рус. перев.: *Дасгупта С., Пападимитриу Х., Вазирани У.* Алгоритмы / Пер. с англ. А. Куликова под ред. А. Шеня. М.: МЦНМО, 2014. 320 с.

[7] *Heideman M. T., Johnson D. H., Burrus C. S.* Gauss and the History of the Fast Fourier Transform // IEEE ASSP magazine. 1984. Vol. 1, iss. 4. P. 14–21. https://ieeexplore.ieee.org/document/1162257, http://www.cis.rit.edu/class/simg716/Gauss_History_FFT.pdf

[8] *Higham N. J.* Stability of a method for multiplying complex matrices with three real matrix multiplications // SIAM J. Matrix Anal. Appl. 1992. Vol. 13, № 3. P. 681–687. https://pdfs.semanticscholar.org/fa55/3f9528a38cba2a23986071354425ea748480.pdf

[9] International Business Machines (IBM). *IBM 77 electric punch collator*. http://www-03.ibm.com/ibm/history/exhibits/vintage/vintage_4506VV4004.html

[10] *Knuth D. E.* The Art of Computer Programming. Vol. 2: Seminumerical algorithms. Third edition. Addison-Wesley (copyright: 1998, first printing: September 1997). ISBN 0-201-89684-2. (First edition: 1969; second edition: 1981).

[11] *Knuth D. E.* The Art of Computer Programming. Vol. 3: Sorting and Searching. Second edition. Addison-Wesley (copyright: 1998, first printing: March 1998). ISBN 0-201-89685-0.

[12] *Moore C, Mertens S.* The Nature of Computation. Oxford University press, 2011.

[13] Multiplication algorithm // Wikipedia, https://en.wikipedia.org/wiki/Multiplication_algorithm, 04.01.2017.

[14] *Munro I.* Some results concerning efficient and optimal algorithms // Proceedings of the Third Annual ACM Symposium on Theory of Computing (STOC 1971). ACM, 1971. P. 40–44.

[15] *Papakonstantinou P. A.* (York University). Introduction to Divide and Conquer. Integer multiplication faster than $O(n^2)$. https://www.eecs.yorku.ca/course_archive/2005-06/F/3101/dc_intro.pdf

[16] Ivars Peterson. Divide and Conquer Multiplication. *Science News*. February 11, 2007. https://www.sciencenews.org/article/divide-and-conquer-multiplication

[17] Tim Roughgarden. Video lectures, CS161 — Design and Analysis of Algorithms. Lecture 9 of 172. http://openclassroom.stanford.edu/MainFolder/VideoPage.php?course=IntroToAlgorithms&video=CS161L1P9

[18] *Winograd S.* On Multiplication of 2 × 2 Matrices // Linear Algebra and its Applications. 1971. Vol. 4. P. 381–388. http://www.sciencedirect.com/science/article/pii/0024379571900097

[19] *Белов А.*, *Тихомиров В.* Сложность алгоритмов // Квант. 1999. № 2. С. 8–11. http://kvant.mccme.ru/pdf/1999/02/kv0299belov.pdf

[20] *Карацуба А. А.* Сложность вычислений // Труды Математического института РАН. 1995. Т. 211. С. 186–202. http://www.ccas.ru/personal/karatsuba/divcru.pdf

[21] *Карацуба А. А.* Комментарии к моим работам, написанные мной самим. (Подготовили к публикации (частичной) С. А. Гриценко и Е. А. Карацуба) // Современные проблемы математики. 2013. Вып. 17. С. 7–29. DOI:http://dx.doi.org/10.4213/spm41

[22] *Карацуба А. А.*, *Офман Ю. П.* Умножение многозначных чисел на автоматах // ДАН СССР. 1962. Т. 145, № 2. С. 293–294.

[23] Обсуждение вопроса Алексея Устинова на сайте MathOverflow. https://mathoverflow.net/questions/319559/gauss-trick-vs-karatsuba-multiplication/319589

[24] *Офман Ю. П.* Об алгоритмической сложности дискретных функций // ДАН СССР. 1962. Т. 145, № 1. С. 48–51.

[25] *Офман Ю.* О христианстве и иудаизме. М.: Изд-во ПСТГУ, 2015.

[26] *Успенский В. А.* Почему на клетке слона написано «буйвол»: Наблюдения о словесных квипрокво (подменах текста) и их причинах // Труды по **не**математике. Кн. 4: Филология. М.: Объединённое гуманитарное издательство. Фонд «Математические этюды», 2012. С. 254–461.

[27] *Успенский В. А.* Ещё раз о словесных квипрокво // Труды по **не**математике. Кн. 5: Воспоминания и наблюдения. М.: Объединённое гуманитарное издательство. Фонд «Математические этюды», 2018. С. 800–807.

[28] *Успенский В. А.* Парижские сюрпризы // Труды по **не**математике. Кн. 5: Воспоминания и наблюдения. М.: Объединённое гуманитарное издательство. Фонд «Математические этюды», 2018. С. 616–622.

[29] *Успенский В. А.* Привычные вывихи // Труды по **не**математике. Кн. 5: Воспоминания и наблюдения. М.: Объединённое гуманитарное издательство. Фонд «Математические этюды», 2018. С. 808–816.

Александр Шень, LIRMM CNRS & University of Montpellier, ИППИ РАН
sasha.shen@gmail.com