О СХЕМАХ ЛОГАРИФМИЧЕСКОЙ ГЛУБИНЫ ДЛЯ ИНВЕРТИРОВАНИЯ В КОНЕЧНЫХ ПОЛЯХ ХАРАКТЕРИСТИКИ ДВА*)

И. С. СЕРГЕЕВ

(МОСКВА)

§ 1. Введение

В последние 30 лет арифметика конечных полей получила сильный импульс к развитию со стороны, в первую очередь, криптографических приложений. Особенный интерес представляет реализация операций в конечных полях характеристики 2 (см., например, [3]).

В настоящей работе рассматривается операция вычисления обратного элемента (или *инвертирования*) в поле $GF(2^n)$ (такое обозначение принято для поля Галуа порядка 2^n). Она обыкновенно является частью (причем самой существенной с точки зрения схемного быстродействия) операции деления.

Для реализации указанных операций мы будем использовать схемы из функциональных элементов в базисе из всех двухвходовых булевых функций. Для наиболее часто употребляемых функций отрицания, конъюнкции, дизъюнкции и суммы по модулю 2 мы применяем обозначения $\bar{}$, $\cdot$, $\vee$, $\oplus$ соответственно.

Важнейшими показателями эффективности схемы являются ее сложность (количество элементов) и глубина (наибольшее количество элементов в цепи, идущей от входов к выходам). Понятия сложности и глубины распространяются на булевы функции. Сложностью (глубиной) функции называется минимально возможная сложность (глубина) для схемы, реализующей данную функцию.**) Сложность и глубина функции f обозначаются через $L(f)$ и $D(f)$. Более подробное изложение понятий глубины и сложности имеется в [9].

Конечным полем называется конечное кольцо с единицей, множество ненулевых элементов которого образует абелеву группу относительно операции умножения. Конечное поле $GF(2^n)$ является векторным пространством размерности n над двухэлементным полем $GF(2)$ (с операцией умножения векторов). Различные представления поля связаны с выбором различных базисов в нем. Наиболее употребимыми являются *стандартные*

*) Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект 05–01–00994), программы поддержки ведущих научных школ РФ (проект НШ–5400.2006.1) и программы фундаментальных исследований Отделения математических наук РАН «Алгебраические и комбинаторные методы математической кибернетики» (проект «Синтез и сложность управляющих систем»).

**) Для многомерной булевой функции будем использовать термин «отображение».

(или *полиномиальные*) и *нормальные* базисы. В основном тексте будут рассмотрены только полиномиальные базисы и полиномиальное представление элементов поля. Фундаментальное изложение теории конечных полей можно найти в [8], а алгоритмические аспекты изложены в [3, 28].

При вычислениях в полиномиальном базисе элементы $GF(2^n)$ интерпретируются как многочлены степени $n-1$ над $GF(2)$, а арифметические операции выполняются по модулю некоторого неприводимого многочлена $m_n(t)$ степени n. Как правило, в качестве характеристического многочлена поля выбирается неприводимый многочлен, содержащий наименьшее количество ненулевых коэффициентов, обычно трехчлен или пятичлен.

Для инвертирования в конечном поле $GF(2^n)$ на практике применяются две группы алгоритмов. Алгоритмы первой группы основаны на методе аддитивных цепочек. Так как для любого $x \in GF(2^n)$ справедливо тождество Ферма $x = x^{2^n}$, то инвертирование совпадает с возведением в степень $2^n - 2$, которое можно выполнить, построив аддитивную цепочку для показателя степени $2^n - 2$. Метод Брауэра сводит инвертирование к выполнению $O(\log n)$ умножений и операций Фробениуса (т. е. возведений в степени вида 2^k). Выполняя последние операции методом Брента—Кунга [18] (см. § 6), можно получить оценку сложности инвертирования $O(n^{1.667})$ и глубины $O(\log^2 n)$ (подробнее о применении аддитивных цепочек к инвертированию см. [3]). Практически используются методы с глубиной $O(\log^2 n)$ и сложностью $O(n^2)$.

Вторая группа методов основана на расширенном алгоритме Евклида нахождения НОД многочленов. Быстрый вариант этого алгоритма, принадлежащий Шёнхаге и Монку (см. [22, гл. 11]), в сочетании с методом умножения двоичных многочленов Шёнхаге [32] приводит к оценке сложности инвертирования $O(n \log^2 n \log\log n)$ и глубины $O(\log^2 n)$. На практике, однако, применяются асимптотически более сложные методы. Ввиду того, что алгоритм Евклида затруднительно реализовать в виде схемы, алгоритмы указанной группы используются, как правило, в программной реализации.

В работе [29] была построена схема для инвертирования в поле $GF(2^n)$ глубины $O(\log n)$. Еще один способ построения такой схемы был предложен в работе [21]. Ни показатель степени в оценке сложности $n^{O(1)}$, ни мультипликативный коэффициент в оценке глубины в [21, 29] не оцениваются (см. § 7.1). Вероятно, настоящая работа является хронологически следующей за двумя перечисленными, посвященной инвертированию в конечных полях с логарифмической глубиной (не считая краткого сообщения [11]).

Основной результат состоит в доказательстве следующей теоремы.

Т е о р е м а. *Инвертирование и деление в поле* $GF(2^n)$ *реализуются с глубиной* $6.44 \log n + o(\log n)$ *и сложностью* $\frac{2}{3}n^4 + o(n^4)$; *либо со сложностью* $O(n^{1.667})$ *и глубиной* $O(\log n)$.

Здесь и везде ниже мы опускаем символ основания у двоичных логарифмов.

Настоящая работа носит теоретический характер: несмотря на то, что предлагаемый метод при достаточно больших n может опережать известные методы, по крайней мере, по глубине, в полях с размерностью $n < 1000$, имеющих прикладное значение, он вряд ли может быть востребован. О практическом построении схем для инвертирования см. [4 ,14].

Основным инструментом при построении схем деления и инвертирования является схема, реализующая возведение в произвольную степень M в поле $GF(2^n)$. Она описана в § 3, там же получена оценка глубины инвертирования. Перед этим, в § 2, рассматриваются вспомогательные операции конечных полей, которые используются в дальнейших построениях. В § 4 подробно рассмотрена схема для дискретного логарифмирования, необходи-

мая для обоснования результата § 3. Способ уменьшения асимптотической сложности инвертирования изложен в § 5. В § 6 представлен окончательный результат работы о сложности инвертирования. В § 7 собраны замечания, касающиеся смежных вопросов.

§ 2. Некоторые операции конечных полей

Отображение U: $GF(2)^n \to GF(2)^m$ называется *линейным*, если для любых векторов $x, y \in GF(2)^n$ выполняется $U(x+y) = U(x) + U(y)$. Последнее условие эквивалентно условию существования такой $m \times n$-матрицы U над $GF(2)$, что для любого вектора $x \in GF(2)^n$ выполнено $U(x) = Ux$. В этом случае будем говорить, что отображение U имеет размерность $m \times n$.

2.1. Примеры линейных операций. Перечислим используемые далее операции арифметики конечных полей, для которых выполняется свойство линейности.

Через $S_{n,l}$ обозначим оператор возведения в степень 2^l в поле $GF(2^n)$ — он называется *оператором Фробениуса* и является линейным: для любых $a, b \in GF(2^n)$ справедливо тождество Фробениуса

$$(a+b)^{2^l} = a^{2^l} + b^{2^l}.$$

При работе в полиномиальном базисе используется операция вычисления остатка от деления произвольного многочлена $h(t)$ на (неприводимый) многочлен $m_n(t)$, определяющий рассматриваемый базис поля. Введем обозначение $B_{n,p}$ для преобразования, приводящего многочлен степени не выше $p-1$ над $GF(2)$ по модулю $m_n(t)$. Эта операция является линейной, так как для любых двоичных многочленов $h_1(t)$ и $h_2(t)$ выполняется

$$(h_1(t) + h_2(t)) \bmod m_n(t) = (h_1(t) \bmod m_n(t)) + (h_2(t) \bmod m_n(t)).$$

Рассмотрим также операцию подстановки в произвольный многочлен степени не выше $p-1$ над $GF(2)$ вместо переменной фиксированного элемента a поля $GF(2^n)$. Эта операция является так называемой *модулярной композицией* многочленов, обозначим ее через $C_{n,p,a}$. Она линейна, так как для любой пары многочленов $h_1(x)$ и $h_2(x)$ выполнено тождество $(h_1+h_2)(a) = h_1(a) + h_2(a)$, непосредственно вытекающее из правила сложения коэффициентов при подобных степенях a:

$$(h_{1,i} + h_{2,i})a^i = h_{1,i}a^i + h_{2,i}a^i,$$

где $h_{1,i}$ и $h_{2,i}$ обозначают коэффициенты соответствующих многочленов при x^i.

В поле $GF(2^k)$ зафиксируем набор элементов $\{\alpha_j \mid j = 1, \ldots, p\}$. Отображение $F_{k,s,p}$ произвольному многочлену степени не выше $s-1$ над $GF(2)$ ставит в соответствие вектор его значений на данном наборе.

Пусть $s = p$. Согласно основному свойству интерполяции, отображение $F_{k,p,p}$ устанавливает взаимно однозначное соответствие между $GF(2)^p$ и $\mathrm{Im}(F_{k,p,p}) \subset GF(2^k)^p$, поэтому существует обратное отображение $F_{k,p}^{-1}$.

Фактически отображения $F_{k,s,p}$ и $F_{k,p}^{-1}$ выполняют соответственно обратную и прямую операцию интерполяции для множества многочленов с коэффициентами из $GF(2)$ (можно было бы считать коэффициенты многочленов относящимися к полю $GF(2^k)$, что более традиционно, но такое обобщение нам не понадобится).

Отображение $F_{k,s,p}$ является объединением p линейных отображений C_{k,s,α_j}, $j=1,\ldots,p$, тем самым оно линейное. Отображение $F_{k,p}^{-1}$, как обратное к $F_{k,p,p}$, также является линейным.

2.2. Сложность линейных операций. Рассмотрим произвольное линейное отображение $A_{m,n}$ размерности $m\times n$ с матрицей A. Умножение такой матрицы на произвольный вектор длины n можно интерпретировать как вычисление системы m линейных комбинаций компонент вектора над полем $GF(2)$.

Сложность вычисления одной линейной комбинации n переменных не превосходит $n-1$ при глубине $\lceil\log n\rceil$ (здесь и везде далее символ основания у двоичных логарифмов опускается). Следовательно, для независимого вычисления m комбинаций требуется не более $m(n-1)$ функциональных элементов. Методом О. Б. Лупанова [10] (см. также [9]), который излагается далее, можно построить схему сложности $O(mn/\log n)$ и глубины $\lceil\log n\rceil+1$, т. е. асимптотически оптимальную по глубине и сложности в классе всех линейных отображений размерности $m\times n$.

Лемма 1. *Существует схема вычисления всех линейных комбинаций s переменных, минимальная по глубине и по сложности.*

Доказательство. Параллельно вычисляются всевозможные суммы двух переменных. На следующем уровне — суммы различных троек и четверок переменных (с помощью уже вычисленного). Затем суммы наборов по 5, 6, 7 и 8 переменных, и т. д. Все элементы, а также все входы построенной схемы являются ее выходами, откуда следует минимальность по сложности. Число элементов в схеме равно 2^s-s-1, а глубина равна $\lceil\log s\rceil$ и является минимально возможной.

Теорема 1 (О. Б. Лупанов, 1956). *Сложность умножения двоичной матрицы A размера $m\times n$ на двоичный вектор x длины n не превосходит $\frac{mn}{\log m}\left(1+\frac{\log\log m+2}{\log m-\log\log m}\right)$, а глубина не превосходит $\lceil\log n\rceil+1$.* *)

Доказательство. Разобьем компоненты $x_0,\ldots,x_{n-1}$ вектора x на группы по s штук. Для каждой из групп реализуем все линейные комбинации методом леммы 1. Произвольная линейная комбинация всех компонент вектора x строится из «коротких», дополнительно требуя $\lceil n/s\rceil-1$ элементарных сложений. Оценим сложность всей схемы:

$$L(A_{m,n})<\frac{n}{s}(2^s-s-1)+m\frac{n}{s}.$$

Выберем $s=\lceil\log m-\log\log m\rceil$ и подставим в полученную оценку,

$$L(A_{m,n})<\frac{n}{\log m-\log\log m}\left(\frac{2m}{\log m}+m\right)=\frac{mn}{\log m}\cdot\frac{\log m+2}{\log m-\log\log m},$$

что доказывает теорему в части сложности.

Оценим глубину схемы:

$$D(A_{m,n})\leqslant\lceil\log s\rceil+\lceil\log\lceil n/s\rceil\rceil\leqslant\lceil\log n\rceil+1.$$

2.3. Нелинейные операции. Отметим, что каждую из приведенных в качестве примеров линейных операций можно реализовать с меньшей сложностью, чем линейное отображение вообще.

Так, методом [18] операцию модулярной композиции $C_{n,p,a}$ можно свести к умножению матрицы размера $\sqrt{p}\times\sqrt{p}$ на матрицу размера $\sqrt{p}\times n$,

*) Здесь и во всех последующих формулировках оценки сложности и глубины функций указываются для реализации одной схемой.

которое можно выполнить быстрее, чем за $O(pn/\log p)$ операций (см., например, [27]). Операция Фробениуса $S_{n,l}$ является частным случаем модулярной композиции и также требует менее $O(n^2/\log n)$ операций и, в частности, может быть реализована со сложностью $O(n^{1.667})$ (см. § 6).

Фундаментальной нелинейной операцией в поле $GF(2^n)$ является умножение, обозначим ее через M_n. Умножение в поле обычно выполняется в два действия: умножение многочленов, характеризующих перемножаемые элементы, и приведение результата по модулю $m_n(t)$. «Школьный» алгоритм умножения многочленов степени $n-1$ имеет сложность $2n^2$ и глубину $\lceil \log n \rceil + 1$. Приведение по модулю является линейной операцией типа $B_{n,2n-1}$ и может быть выполнено с такой же глубиной и сложностью.

Однако, метод Шёнхаге [32] позволяет умножать двоичные многочлены со сложностью $O(n \log n \log\log n)$ и глубиной $O(\log n)$. В свою очередь, приведение по модулю может быть сведено к умножению многочленов. Опишем это сведение, следуя [22]. Представим многочлен $h(t)$ степени не выше $2n-1$ в виде $a(t)t^n + b(t)$, где $\deg a, b < n$. Введем обозначение $\tilde{c}(t) = t^{\deg c} c(1/t)$, т. е. коэффициенты многочлена $\tilde{c}(t)$ являются коэффициентами $c(t)$, записанными в обратном порядке. Пусть $a(t)t^n = q(t)m_n(t) + r(t)$, где $\deg q, r < n$, тогда $h(t) \bmod m_n(t) = r(t) + b(t)$. Остаток $r(t)$ вычисляется посредством двух умножений следующим образом. Имеем,

$$\tilde{a}(t) = \tilde{q}(t)\tilde{m}_n(t) + t^n \tilde{r}(t).$$

Если $i(t)$ — обратный к $\tilde{m}_n(t)$ многочлен по модулю t^n, т. е. $i(t)\tilde{m}_n(t) = 1 \bmod t^n$ (он существует, так как младший коэффициент $\tilde{m}_n(t)$ равен 1), тогда

$$\tilde{q}(t) = \tilde{a}(t)i(t) \bmod t^n \quad \text{и} \quad r(t) = q(t)m_n(t) \bmod t^n.$$

Таким образом,

$$L(B_{n,2n}) \leqslant 2M(n) + n, \qquad D(B_{n,2n}) \leqslant 2D(n) + 1,$$

и следовательно

$$L(M_n) \leqslant 3M(n) + n, \qquad D(M_n) \leqslant 3D(n) + 1,$$

где $M(n)$, $D(n)$ — сложность и глубина умножения двоичных многочленов степени $n-1$.

Операция $F_{k,s,p}$, т. е. вычисление значений многочлена степени $s-1$ в p точках поля $GF(2^k)$ в интересующем нас случае $s \leqslant p$ может быть выполнена алгоритмом [1] за $O(M(p)\log p)$ операций в поле $GF(2^k)$ с глубиной $O(\log^2 p)$ над тем же полем (для сравнения, из теоремы 1 следует оценка сложности $O(skp/\log(kp))$). Следовательно,

$$L(F_{k,p,p}) \leqslant O(M(p)\log p)L(M_k), \qquad D(F_{k,p,p}) \leqslant O(\log^2 p)D(M_k).$$

Аналогичный алгоритм (см. [1]) позволяет получить те же самые оценки для операции интерполяции $F_{k,p}^{-1}$. В § 6 будет описан «параллельный» вариант этого алгоритма, с глубиной логарифмического порядка.

Отметим, что метод Лупанова позволяет реализовать произвольное линейное отображение с асимптотически оптимальной глубиной, поэтому он будет преимущественно использоваться в той части, которая посвящена минимизации глубины инвертирования — в той части, которая относится к результату о сложности инвертирования с логарифмическим порядком глубины (§ 6), будут использоваться альтернативные методы, в том числе вышеперечисленные.

§ 3. Алгоритм возведения в степень

В основу последующих построений будет положен алгоритм возведения элемента поля в произвольную (но фиксированную для алгоритма) степень. Приведем его краткое описание.

Пусть элемент $x \in GF(2^n)$ задан полиномиальным представлением, требуется вычислить x^M, где $M = 2^{e_1} + 2^{e_2} + \ldots + 2^{e_m}$. Можно считать, что $M < 2^n - 1$ (так как справедливо тождество Ферма $x^{2^n} = x$) и, следовательно, что $m < n$.

1. Вычислим степени $x^{2^{e_1}}, \ldots, x^{2^{e_m}}$. Пусть элементу $x^{2^{e_i}}$ соответствует многочлен $f_i(t)$ в представлении поля. Пусть далее $f(t) = f_1(t) \cdot \ldots \cdot f_m(t)$. Положим $p = m(n-1)+1$, выберем поле $GF(2^k)$, содержащее не менее p элементов; в нем выберем набор элементов $\alpha_1, \ldots, \alpha_p$.

2. Вычислим всевозможные $f_i(\alpha_j) \in GF(2^k)$, где $i = 1, \ldots, m$, $j = 1, \ldots, p$.

3. Для всех j вычислим произведения $f_1(\alpha_j) \cdot \ldots \cdot f_m(\alpha_j) = f(\alpha_j)$. Для этого в поле $GF(2^k)$ выберем примитивный элемент α. Если $f_i(\alpha_j) \neq 0$ для всех i, тогда

3.1. Вычислим дискретные логарифмы, $\log_\alpha f_i(\alpha_j)$.

3.2. Вычислим $\sum_{i=1}^{m} \log_\alpha f_i(\alpha_j) \bmod (2^k - 1) = \log_\alpha f(\alpha_j)$.

3.3. Вычислим $f(\alpha_j) = \alpha^{\log_\alpha f(\alpha_j)}$.

4. По известным значениям $f(\alpha_j)$, $j = 1, \ldots, p$, восстанавливается многочлен $f(t)$ степени не выше $p-1$.

5. Элементу x^M соответствует многочлен $f(t) \bmod m_n(t)$.

Легко видеть, что в основе алгоритма лежит идея интерполяции, восходящая к работе А. Л. Тоома [13]. Она позволяет свести многократное умножение в поле $GF(2^n)$ к умножению в поле меньшей размерности $GF(2^k)$. Для умножения в «небольшом» поле $GF(2^k)$ используется дискретное логарифмирование, идея применения которого взята из работы [19].

Операцию возведения в некоторую степень веса m (весом называется число единиц в двоичной записи) в поле $GF(2^n)$ будем обозначать через $\boldsymbol{E}_{n,m}$, для краткости опуская информацию о том, в какую именно степень возводится элемент поля — это будет понятно из контекста. Под $L(\boldsymbol{E}_{n,m})$ и $D(\boldsymbol{E}_{n,m})$ будем понимать сложность и глубину реализации самой сложной (глубокой) из операций для степеней веса m.

Заметим, что на шаге 1 алгоритма выполняются операции $\boldsymbol{S}_{n,e_i}$, $i = 1, \ldots, m$, на втором шаге — m операций $\boldsymbol{F}_{k,n,p}$, на шаге 4 — операция $\boldsymbol{F}^{-1}_{k,p}$, на шаге 5 — операция $\boldsymbol{B}_{n,p}$. Перечисленные операции являются линейными (см. § 2).

Введем также следующие обозначения: $\boldsymbol{\Lambda}_k$ — для операции дискретного логарифмирования с основанием α в поле $GF(2^k)$; $\boldsymbol{\Sigma}_{m,k}$ — для операции суммирования m k-разрядных чисел по модулю $2^k - 1$; $\boldsymbol{\Lambda}^{-1}_k$ — для операции возведения примитивного элемента $\alpha \in GF(2^k)$ в степень, которая является k-разрядным числом; $\boldsymbol{X}_{k,m}$ — для индикации того, что m элементов поля $GF(2^k)$ отличны от нуля.

Шаг 3 состоит в выполнении p операций m-кратного умножения в поле $GF(2^k)$ (введем обозначение $\boldsymbol{\Phi}_{k,m}$ для одного такого умножения) согласно схеме:

$$\boldsymbol{\Phi}_{k,m}(y_1, \ldots, y_m) = \boldsymbol{X}_{k,m}(y_1, \ldots, y_m) \times \boldsymbol{\Lambda}^{-1}_k \cdot \boldsymbol{\Sigma}_{m,k}(\boldsymbol{\Lambda}_k(y_1), \ldots, \boldsymbol{\Lambda}_k(y_m)),$$

где символ · означает композицию отображений, а символ × — обычное умножение скалярной величины на вектор.

Операция $X_{k,m}$ реализуется конъюнкцией дизъюнкций разрядов аргументов-элементов поля с минимально возможной для функции, существенно зависящей от всех своих переменных, сложностью $mk-1$ и глубиной $\lceil \log m \rceil + \lceil \log k \rceil$ — что несущественно с точки зрения схемы параллельно реализуемого многократного умножения.

Операция дискретного логарифмирования Λ_k будет подробно рассмотрена в § 4, где будет показано, что для произвольного $\varepsilon > 0$ можно выбрать $k = \log p + C_0(\varepsilon)$ так, что

$$L(\Lambda_k) \leqslant C_1(\varepsilon) O(p^\varepsilon), \qquad D(\Lambda_k) \leqslant \varepsilon \log p + C_2(\varepsilon) + O(\log^2 \log p).$$

Рассмотрим операцию m-кратного суммирования $\Sigma_{m,k}$. Один из способов построения схем малой глубины для данной операции заключается в следующем. Известно (см., например, [5]), что сложение нескольких чисел может быть сведено к сложению меньшего количества чисел с глубиной $O(1)$ при помощи схемы, называемой *компрессором*.

Простейшим примером такой схемы является (3,2)-компрессор. Если даны три k-разрядных числа: $a = (a_{k-1}, \ldots, a_0)$, $b = (b_{k-1}, \ldots, b_0)$, $c = (c_{k-1}, \ldots, c_0)$ (старшинство разрядов возрастает справа налево), то сумму $a_i + b_i + c_i$ можно представить в виде $2u_i + v_i$, где

$$v_i = a_i \oplus b_i \oplus c_i, \quad u_i = a_i \cdot b_i \oplus b_i \cdot c_i \oplus a_i \cdot c_i.$$

Так количество слагаемых сокращается с 3 до 2: $a + b + c = u + v$, где $u = (u_{k-1}, \ldots, u_0, 0)$, $v = (v_{k-1}, \ldots, v_0)$. Пара разрядов (u_i, v_i) вычисляется со сложностью 5 и глубиной 3. Окончательно, сложность компрессора равна $5k$, а глубина — 3.

Из подобных подсхем-компрессоров можно построить схему, которая с глубиной $O(\log m)$ преобразует m чисел на входах в $O(1)$ чисел на выходах с сохранением суммы. Окончательно, полученные числа могут быть сложены посредством обычных сумматоров.

Если (как в нашем случае) k-разрядные числа складываются по модулю $2^k - 1$, то получаемые в процессе вычислений старшие разряды следует перемещать на место младших. Например, модулярный (3,2)-компрессор должен возвращать числа $u' = (u_{k-2}, \ldots, u_0, u_{k-1})$ и $v = (v_{k-1}, \ldots, v_0)$, где u_i, v_i определяются выше.

Вероятно, наилучшая теоретическая оценка глубины схемы компрессоров, сводящей m-кратное суммирование к сложению двух чисел, $3.44 \log m + O(1)$, получена в работе [25] методом из работы [30]. Сложность такой схемы $O(mk)$. Константы под знаком O в этих оценках достаточно велики — практически, можно строить схемы глубины не более $3.71 \log m$ и сложности $5mk$ из (3,2)-компрессоров.

Обыкновенный k-разрядный сумматор можно реализовать схемой линейной сложности и глубины $\log k + O(\sqrt{\log k})$ методом В. М. Храпченко [15]. Однако, на практическом интервале значений k лучше работают другие методы. Например, метод М. И. Гринчука [24] имеет оценку глубины $1.27(\log k + 1) + 3$. Сложность схемы, непосредственно построенной по методу Гринчука, равна $O(k \log k)$, однако она может быть приведена к линейной при помощи стандартной процедуры линеаризации (см., например, [15]) с увеличением глубины на $O(\log \log k)$.

Сложение двух k-разрядных чисел по модулю $2^k - 1$ можно свести к обычному сложению $2k$-разрядных чисел. Действительно, если $a, b \leqslant 2^k - 1$, то $a + b \bmod (2^k - 1) = c + d$, где $a + b = c2^k + d$ и $c + d \leqslant 2^k - 1$.

Результат $a+b \bmod(2^k-1)$ содержится в разрядах с k-го по $(2k-1)$-й (нумерация с нуля) суммы чисел $(2^k+1)a$ и $(2^k+1)b$.

Окончательно, имеем

$$L(\Sigma_{m,k}) = O(mk), \qquad D(\Sigma_{m,k}) \leqslant 3.44 \log m + O(\log k).$$

Сложность и глубину схемы экспоненцирования (реализующей операцию Λ_k^{-1}) оценим грубо, но этого будет достаточно для наших целей. Воспользуемся двоичной записью числа b: $(b_{k-1}, b_{k-2}, \ldots, b_0)$, тогда

$$\alpha^b = \alpha^{b_0}\alpha^{2b_1} \cdot \ldots \cdot \alpha^{2^{k-1}b_{k-1}}.$$

Для вычисления каждого из сомножителей в правой части формулы достаточно k функциональных элементов ($k-1$ конъюнкций и элемент, реализующий функцию $\bar{x} \vee y$), поскольку

$$\alpha^{2^i b_i} = \overline{b_i} \cdot 1 \vee b_i \alpha^{2^i},$$

где 1 — единица поля $GF(2^k)$, а символ $\vee$ означает поразрядную дизъюнкцию. Считаем, что степени α вычислены предварительно.

Таким образом, самый простой способ вычисления α^b сводится к $k-1$ умножениям в поле, откуда следуют оценки

$$L(\Lambda_k^{-1}) \leqslant kL(M_k) = O(k^2 \log k \log\log k),$$
$$D(\Lambda_k^{-1}) \leqslant \lceil \log k \rceil D(M_k) + 1 = O(\log^2 k).$$

Теорема 2. *Пусть m — вес числа M. Тогда для операции возведения в степень M в $GF(2^n)$ выполняются оценки (при $\varepsilon > 0$):*

$$L(E_{n,m}) \lesssim \frac{\log(mn) + C_0(\varepsilon)}{\log(m^2 n)} m^2 n^2 + C_1(\varepsilon) m^{2+\varepsilon} n^{1+\varepsilon},$$
$$D(E_{n,m}) \lesssim (2+\varepsilon)\log n + 4.44 \log m + D_0(\varepsilon).$$

Доказательство. Выберем $k = \log(mn) + C_0(\varepsilon)$, такое, что справедливы оценки сложности и глубины логарифмирования из следствия 3 (см. § 4).

Рассматривая композицию линейных отображений S_{n,c_i} и $F_{k,n,p}$, $i = 1, \ldots, m$, как одно линейное отображение размерности $kmp \times n$, из метода Лупанова имеем оценку сложности для соответствующей подсхемы $(1+o(1))(kmpn/\log(kmp)) \leqslant \frac{\log(mn)+C_0(\varepsilon)}{\log(m^2 n)} m^2 n^2$. Глубина подсхемы не превосходит $\log n + 2$.

Другое линейное преобразование $B_{n,p} \cdot F_{k,p}^{-1}$ размерности $n \times kp$ реализуется подсхемой сложности $O(pkn/\log n) \sim O(mn^2)$ и глубины $\log(kp) + 2 \sim \log(mn) + D_1(\varepsilon)$.

Подсхема вычисления дискретных логарифмов во вспомогательном поле состоит из mp параллельно расположенных блоков, реализующих операции типа Λ_k. Сложность этой подсхемы (см. § 4) оценивается как $C_1(\varepsilon) mp^{1+\varepsilon} \sim C_1(\varepsilon) m^{2+\varepsilon} n^{1+\varepsilon}$. Глубина логарифмирования составляет $(\varepsilon + o(1))\log p + D_2(\varepsilon) \sim (\varepsilon + o(1))\log(mn) + D_2(\varepsilon)$.

Далее, сложность реализации p сумматоров типа $\Sigma_{m,k}$ оценивается как $O(mkp) \sim O(m^2 n \log n)$. Для глубины справедлива оценка $3.44 \log m + O(\log k) = 3.44 \log m + O(\log\log n) + D_3(\varepsilon)$.

Таким образом, сложность всей схемы определяется оценкой для первого блока линейных отображений. Асимптотика глубины складывается из глубин четырех подсхем,

$$D(\boldsymbol{E}_{n,m}) \lesssim \log n + \log(mn) + D_1(\varepsilon) + \varepsilon \log(mn) + D_2(\varepsilon) + 3.44 \log m + D_3(\varepsilon) \sim$$
$$\sim (2+\varepsilon)\log n + 4.44 \log m + D_0(\varepsilon).$$

Теорема 3. *Инвертирование в поле* $GF(2^n)$ *реализуется схемой глубины и сложности*

$$D(\boldsymbol{I}_n) \leqslant (6.44 + o(1)) \log n, \qquad D(\boldsymbol{I}_n) \leqslant (2/3 + o(1)) n^4.$$

Доказательство. Операция инвертирования соответствует возведению в степень $2^n - 2$, имеющему вес $n-1$. Действительно,

$$x^{-1} = x^{2^n-2} = x^2 x^{2^2} \cdot \ldots \cdot x^{2^{n-1}}.$$

Данная теорема является прямым следствием теоремы 2, нужно только положить $m = n-1$, а ε выбрать в пределах погрешности округления константы из работы [25] до 3.44.

Обозначим через $\boldsymbol{\Delta}_n$ операцию деления в поле $GF(2^n)$. Очевидно, что деление сводится к одному инвертированию и одному умножению в поле, однако умножение может быть интегрировано в предложенную выше схему инвертирования, так как

$$\frac{y}{x} = y x^2 x^{2^2} \cdot \ldots \cdot x^{2^{n-1}}.$$

Положим $m = n$ в алгоритме из начала параграфа. Выполняя шаг 2 для y отдельно и параллельно с совмещенными шагами 1, 2 для x, а далее схема ничем не будет отличаться от возведения в степень веса n, получаем следующий результат.

Теорема 4. *Для операции деления в* $GF(2^n)$ *справедливы оценки:*

$$D(\boldsymbol{\Delta}_n) \leqslant (6.44 + o(1)) \log n, \qquad L(\boldsymbol{\Delta}_n) \leqslant (2/3 + o(1)) n^4.$$

С практической точки зрения, предложенная схема едва ли представляет интерес. Прикидка показывает, что для значений n, порядка не выше нескольких сотен тысяч, видимо, не худшую глубину имеет стандартный метод инвертирования. При этом сложность стандартного метода всегда $O(n^3)$, а фактически для большинства полей $O(n^2)$.

§ 4. Дискретное логарифмирование

Зафиксируем α — порождающий элемент мультипликативной группы поля $GF(2^k)$ (мультипликативная группа обозначается через $GF(2^k)^*$ и состоит из всех ненулевых элементов поля). Тогда для любого элемента $\beta \in GF(2^k)^*$ единственным образом определяется число $b \in 0, \ldots, 2^k - 2$, такое, что $\beta = \alpha^b$. Оно называется *дискретным логарифмом элемента* β *по основанию* α. Оценим параметры схемы, реализующей операцию дискретного логарифмирования $\boldsymbol{\Lambda}_k$, где $\boldsymbol{\Lambda}_k(\beta) = b$.

Рассматриваемый далее способ является реализацией алгоритма Сильвера—Полига—Хеллмана (см., например, [7]) в виде схемы из функциональных элементов.

Пусть известно некоторое разложение $2^k - 1$ на взаимно простые сомножители

$$2^k - 1 = r_1 r_2 \cdot \ldots \cdot r_w.$$

Введем обозначение $q_i = (2^k - 1)/r_i$, где $i = 1 \ldots w$. Заметим, что

$$\beta^{q_i} = (\alpha^b)^{q_i} = (\alpha^{q_i})^{b \bmod r_i}.$$

Таким образом, сравнивая β^{q_i} со всевозможными степенями элемента α^{q_i} (нужно всего r_i таких сравнений), можно определить остаток $b_i = b \bmod r_i$. По набору остатков b_i, $i = 1 \ldots w$, число b восстанавливается однозначно. Рассмотрим следующую схему вычислений.

1. Вычислим все $\beta_i = \beta^{q_i}$, $i = 1, \ldots, w$.

2. Для всех i среди $j = 0, \ldots, r_i - 1$ покоэффициентным сравнением элементов β_i и α^{jq_i} (последние вычислены предварительно) найдем b_i, удовлетворяющее $\beta_i = \alpha^{b_i q_i}$.

3. Число $b = \log_\alpha \beta$ восстанавливается по своим остаткам $b_i = b \bmod r_i$.

Операцию вычисления w степеней элемента $\beta \in GF(2^k)$ обозначим через $\boldsymbol{H}_{k,w}$. Очевидные оценки глубины и сложности этой операции содержатся в следующей лемме.

Лемма 2. *Для сложности и глубины операции $\boldsymbol{H}_{k,w}$ справедливы оценки*

$$L(\boldsymbol{H}_{k,w}) \leqslant w(k-3)L(\boldsymbol{M}_k) + O(k^3/\log k),$$
$$D(\boldsymbol{H}_{k,w}) \leqslant \lceil \log(k-2) \rceil D(\boldsymbol{M}_k) + \lceil \log k \rceil + 1.$$

Доказательство. Все степени вида β^{2^l}, $l = 0, \ldots, k-1$, вычисляются схемой соответствующего линейного оператора размерности $k^2 \times k$ со сложностью и глубиной $O(k^3/\log k)$ и $\lceil \log k \rceil + 1$ соответственно.

Произвольную степень можно представить в виде произведения не более чем $k-2$ сомножителей вида β^{2^l}, так как очевидно, $q_i < 2^{k-1} - 1$. Для вычисления w таких произведений, каждое содержит не более $k-2$ сомножителей, требуется не более $w(k-3)$ умножений в поле.

Если воспользоваться стандартным алгоритмом умножения, то имеем

$$L(\boldsymbol{H}_{k,w}) \leqslant 2wk^3 + O(wk^3/\log k), \quad D(\boldsymbol{H}_{k,w}) < 2(\lceil \log k \rceil + 1)^2.$$

В целом, вычисление системы степеней можно выполнять более экономно, используя сведения из теории аддитивных цепочек (см., например, [3, 6]).

Произведение k многочленов степени $k-1$ реализуется схемой глубины $O(\log k)$. Для ее построения можно воспользоваться аналогичным алгоритмом для умножения чисел из работы [16] (произведение многочленов сводится к числовому произведению, см., например, [19]). Еще один способ состоит в рекурсивном применении метода настоящей работы. Таким образом, на самом деле операцию $\boldsymbol{H}_{k,w}$ можно реализовать схемой логарифмической глубины.

Введем еще несколько обозначений. Пусть $qr = 2^k - 1$, $(q, r) = 1$. На подгруппе корней r-й степени из единицы поля $GF(2^k)$ может быть корректно определена операция $\boldsymbol{\Lambda}_{k,r}$ логарифмирования по основанию порождающего элемента подгруппы, которым является α^q.

Заметим, что $\boldsymbol{\Lambda}_{k,r}(\beta^q) = b \bmod r$, где $b = \log_\alpha \beta$. Отображение $\boldsymbol{\Lambda}$ можно формально доопределить на все поле; за пределами указанной подгруппы корней из единицы определим его произвольным образом.

Отображение $R_{k,w}$ восстанавливает число, имеющее заданные остатки от деления на r_i, $i = 1, \ldots, w$, а именно

$$R_{k,w}(b_1, b_2, \ldots, b_w) = b, \qquad 0 \leqslant b < \prod r_i, \quad b = b_i \bmod r_i, \quad i = 1, \ldots, w.$$

Пользуясь введенными обозначениями, можно записать

$$\Lambda_k(\beta) = R_{k,w}(K_{k,r_1}(\beta^{q_1}), \ldots, K_{k,r_w}(\beta^{q_w})).$$

Лемма 3. *Для сложности и глубины схемы, реализующей отображение $R_{k,w}$, справедливы оценки:*

$$L(R_{k,w}) = O(k^2), \quad D(R_{k,w}) = O(\log k).$$

Доказательство. Согласно китайской теореме об остатках (см., например, [6])

$$b = b_1 c_1 + b_2 c_2 + \ldots + b_w c_w \bmod 2^k - 1,$$
$$c_i = \nu_i r_1 \cdot \ldots \cdot r_{i-1} r_{i+1} \cdot \ldots \cdot r_w = \frac{\nu_i (2^k - 1)}{r_i},$$

где нормирующий коэффициент $\nu_i \in [1, r_i - 1]$ подбирается, исходя из условия $c_i = 1 \bmod r_i$. По построению, числа c_i состоят не более чем из k двоичных разрядов.

Рассмотрим следующий способ организации вычислений (близкий к [16]). Пусть $b_i = (b_{i,j-1}, b_{i,j-2}, \ldots, b_{i,0})$ в двоичном представлении, $j = \lceil \log r_i \rceil$, тогда

$$b_i c_i = b_{i,0} c_i + 2 b_{i,1} c_i + \ldots + 2^{j-1} b_{i,j-1} c_i.$$

Вычисление слагаемых в приведенной формуле осуществляется «бесплатно» — также «бесплатно» выполняется приведение их по модулю $2^k - 1$ (старшие разряды подставляются на место младших). Поступим так с каждым из произведений $b_i c_i$, $i = 1, \ldots, w$. Количество вновь образованных слагаемых оценивается как

$$\sum_{i=1}^{w} \lceil \log r_i \rceil < w + \sum_{i=1}^{w} \log r_i = w + \log(2^k - 1) < k + w.$$

Задача сведена к суммированию не более чем $k + w$ экземпляров k-разрядных чисел по модулю $2^k - 1$ (соответствующее отображение обозначалось через $\Sigma_{k+w,k}$). Поэтому

$$L(R_{k,w}) \leqslant L(\Sigma_{k+w,k}), \quad D(R_{k,w}) \leqslant D(\Sigma_{k+w,k}),$$

после чего утверждение теоремы следует из оценок § 3 и очевидного наблюдения $w < k$.

В работе [26] предложен метод построения схемы сложности $O(k^{1+\varepsilon})$ и глубины $O(\varepsilon^{-1} \log k)$, где $\varepsilon > 0$. По всей видимости, этот метод не превосходит стандартный метод в части глубины.

Далее будет показано, что на асимптотику сложности и глубины схемы логарифмирования в целом блоки, реализующие $H_{k,w}$ и $R_{k,w}$, не оказывают существенного влияния.

Теорема 5. *Сложность и глубина отображения* $M_{k,r}$ *удовлетворяют оценкам:*

$$L(M_{k,r}) < r\left(2 + \frac{k}{\log r} \cdot \frac{\log r + 6}{\log r - \log\log r}\right), \qquad D(M_{k,r}) \leqslant \lceil \log k \rceil + \lceil \log r \rceil + 1.$$

Схема строится из подсхем, которые описаны в следующих двух пунктах.

4.1. Вычисление системы компараторов. Пусть β^q подается на входы подсхем сравнения с соответствующими α^{lq}, $l = 0 \ldots r-1$. Так как последние вычислены предварительно, то компаратор k-разрядного элемента β^q с фиксированным элементом поля есть некоторая обобщенная конъюнкция разрядов β^q (под компаратором здесь понимается схема, определяющая совпадение или несовпадение двух наборов).

Разобьем набор из k переменных (которыми кодируется β^q) на поднаборы, содержащие не более s переменных. Для каждого поднабора построим схему, реализующую все возможные обобщенные конъюнкции этой группы переменных (она называется дешифратором). Чтобы получить необходимые r конъюнкций k переменных, требуется еще не более $r(\lceil k/s \rceil - 1)$ конъюнкций, соединяющих соответствующие выходы дешифраторов. Следующая лемма фактически содержится в [9].

Лемма 4. *Сложность дешифратора* s *переменных* $L(K_s) < 2^s + 3.81 \cdot 2^{s/2}$, *а глубина* $D(K_s) \leqslant \lceil \log s \rceil + 1$.

Доказательство. Рассмотрим следующую схему. Разбиваем множество переменных на две части: они равны, когда s четно, и отличаются на 1 в нечетном случае. Пусть для них построено два дешифратора. Тогда с помощью 2^s конъюнкций объединяем всевозможными способами выходы этих подсхем.

Участвующие в этой конструкции дешифраторы меньшего порядка устроены точно так же. Дешифратор одной переменной включает в себя лишь один функциональный элемент отрицания: $L(K_1) = 1$, $D(K_1) = 1$. Оценим сложность простейших дешифраторов:

$$L(K_2) = 2^2 + 2L(K_1) = 6, \quad L(K_3) = 2^3 + L(K_1) + L(K_2) = 15,$$
$$L(K_4) = 2^4 + 2L(K_2) = 28, \quad L(K_5) = 2^5 + L(K_2) + L(K_3) = 53,$$
$$L(K_6) = 2^6 + 2L(K_3) = 94, \quad L(K_7) = 2^7 + L(K_3) + L(K_4) = 171,$$
$$L(K_8) = 2^8 + 2L(K_4) = 312.$$

В этих случаях заявленная оценка сложности выполняется; константа 3.81 получается при $s = 7$.

Проверку утверждения при $s > 8$ проведем по индукции. Отметим, что приводимая ниже выкладка корректна как для четного ($\delta = 0$), так и для нечетного случая ($\delta = 0.5$).

$$L(K_s) \leqslant 2^s + L(K_{\frac{s}{2}-\delta}) + L(K_{\frac{s}{2}+\delta}) <$$
$$< 2^s + 2^{s/2}(2^\delta + 2^{-\delta}) + 3.81 \cdot 2^{s/4}(2^{\delta/2} + 2^{-\delta/2}) <$$
$$< 2^s + \frac{3}{\sqrt{2}} 2^{s/2} + 3.81 \cdot \frac{1+\sqrt{2}}{\sqrt[4]{2}} 2^{s/4} < 2^s + 3.81 \cdot 2^{s/2},$$

если $2^{s/4} > 4.6$, что выполняется при $s \geqslant 9$.

Глубина построенной схемы дешифратора равна $\lceil \log s \rceil + 1$.

Следствие 1. *Для сложности и глубины системы* r k*-разрядных компараторов (с общим входом) справедливы оценки:*

$$L(Q_{k,r}) < \frac{kr}{\log r} \cdot \frac{\log r + 6}{\log r - \log\log r}, \quad D(Q_{k,r}) \leqslant \lceil \log k \rceil + 2.$$

Доказательство. Пользуясь доказанной леммой, оценим общую сложность схемы сравнения как

$$L(Q_{k,r}) \leqslant \frac{k}{s}(L(K_s)+r) \leqslant \frac{k}{s}\left(2^s + 3.81\cdot 2^{\frac{s}{2}}\right) + \frac{k}{s}r.$$

Выберем параметр $s = \lceil \log r - \log\log r \rceil$, тогда оценка примет вид:

$$L(Q_{k,r}) < \frac{k}{\log r - \log\log r}\left(\frac{2r}{\log r} + 3.81\sqrt{\frac{2r}{\log r}} + r\right) <$$

$$< \frac{kr}{\log r(\log r - \log\log r)}\left(2 + 3.81\sqrt{\frac{2\log r}{r}} + \log r\right) < \frac{kr}{\log r}\cdot\frac{\log r + 6}{\log r - \log\log r},$$

так как $2\log r \leqslant r$.

Глубина отдельного компаратора и, следовательно, всей схемы, не превосходит $\lceil \log k \rceil + 2$.

4.2. Реализация схемы шифратора. Следующая схема по r входам (выходы компараторов), только один из которых может принимать значение 1, вычисляет номер данного входа. Такая схема называется шифратором.

Сопоставим выходу каждой схемы сравнения β^q и α^{lq} номер l, точнее, его двоичную запись. Назовем *частной дизъюнкцией разряда* h дизъюнкцию всех входов с номерами, h-й разряд которых равен 1. Заметим, что частная дизъюнкция выходов компараторов произвольного разряда h вычисляет h-й разряд числа $b \bmod r$. Действительно, получая на входе элемент β, только один компаратор принимает значение 1, а именно тот, который отмечен номером $b \bmod r$. Если h-й разряд $b \bmod r$ равен 1, то выход соответствующего компаратора участвует в частной дизъюнкции разряда h, она, следовательно, равна 1. Иначе, если h-й разряд $b \bmod r$ равен 0, выход компаратора не подается на вход данной дизъюнкции, которая поэтому принимает значение 0. Таким образом, вычисление всего набора частных дизъюнкций дает двоичное представление $\lceil \log r \rceil$-разрядного числа $b \bmod r$. Иначе говоря, выходы шифратора реализуют частные дизъюнкции всех разрядов, вплоть до $\lceil \log r \rceil$-го.

Лемма 5. *При $s > 0$ сложность шифратора с 2^s входами $L(V_{2^s}) \leqslant \leqslant 2^{s+1} - 2s - 2$ при глубине $D(V_{2^s}) = s - 1$.*

Доказательство. Индуктивно построим схему, для которой выполняются эти оценки. В действительности, будет строиться схема, в которой вычисляются все частные дизъюнкции для групп входов с фиксированными старшими разрядами кода. При $s = 1$ входы схемы кодируются одним битом; единственная дизъюнкция совпадает в данном случае со входом, отмеченным единицей, т. е. $L(V_{2^1}) = 0$, $D(V_{2^1}) = 0$.

Рассмотрим переход от s к $s+1$. В зависимости от значения старшего $(s+1)$-го разряда кода (0 или 1) все входы можно разбить на две группы, каждая содержит по 2^s входов. Для каждой из групп реализуем систему s частных дизъюнкций младших разрядов. Соединив соответствующие выходы этих подсхем элементами дизъюнкций получим все правильные частные дизъюнкции для полного набора 2^{s+1} входов, за исключением дизъюнкции $(s+1)$-го разряда.

Частная дизъюнкция старшего разряда объединяет все входы одной из подгрупп. Для ее вычисления можно использовать результаты предшествующих построений. Заметим, что частная дизъюнкция s-го разряда для данного множества входов уже получена на глубине $s-1$; она вычисляет дизъюнкцию половины из входов рассматриваемой подгруппы. Заметим далее, что дизъюнкция половины из оставшихся входов как частная дизъюнкция $(s-1)$-го разряда группы входов с двумя фиксированными старшими разрядами 1 и 0 также уже вычислена на глубине $s-2$, и т. д.

Итак, шифратор с 2^{s+1} входами получается из двух шифраторов с 2^s входами, кроме того s элементов требуется дополнительно для вычисления частной дизъюнкции старшего разряда и по одному — для остальных частных дизъюнкций. Отсюда по индукции имеем

$$L(V_{2^{s+1}}) \leqslant 2L(V_{2^s}) + 2s \leqslant 2(2^{s+1} - 2s - 2) + 2s = 2^{s+2} - 2(s+1) - 2.$$

Параллельно проверяется, что глубина построенной схемы равна s. Глубина частной дизъюнкции $(s+1)$-го разряда равна s по построению. Глубина выходов частных дизъюнкций других разрядов на 1 больше глубины шифратора с 2^s входами, откуда следует, что их глубина также s.

Следствие 2. *Сложность шифратора с r входами $L(V_r) \leqslant 2r - 2\lceil\log r\rceil - 2$ при глубине $D(V_r) = \lceil\log r\rceil - 1$.*

Доказательство. Пусть $2^{s+1} \geqslant r = 2^s + r'$, $r' > 0$. Доказательство проведем индукцией по r.

Схема будет устроена так же, как и в частном случае. Множество входов разобьем на два: 2^s входов с нулевым старшим разрядом и r' — с единичным. Вычислим частные дизъюнкции на этих подмножествах (второе из них кодируется, вообще говоря, $\lceil\log r'\rceil$ младшими разрядами исходного кода).

Далее, $\lceil\log r'\rceil$ функциональных элементов необходимо, чтобы получить дизъюнкции младших разрядов. Еще столько же — для вычисления дизъюнкции $(s+1)$-го разряда. Это приводит к рекуррентному соотношению

$$L(V_r) \leqslant L(V_{2^s}) + L(V_{r'}) + 2\lceil\log r'\rceil \leqslant$$
$$\leqslant (2^{s+1} - 2s - 2) + (2r' - 2\lceil\log r'\rceil - 2) + 2\lceil\log r'\rceil = 2r - 2(s+1) - 2.$$

Глубина схемы равна $s = \lceil\log r\rceil - 1$.

Доказательство теоремы 5. Доказательство получается суммированием оценок из следствий, доказанных в этом и предыдущем пунктах:

$$L(\Lambda_{k,r}) \leqslant L(Q_{k,r}) + L(V_r) < \frac{kr}{\log r} \cdot \frac{\log r + 6}{\log r - \log\log r} + 2r,$$
$$D(\Lambda_{k,r}) \leqslant D(Q_{k,r}) + D(V_r) \leqslant (\lceil\log k\rceil + 2) + (\lceil\log r\rceil - 1).$$

4.3. Выбор вспомогательного поля.

Теорема 6. *Пусть $2^k - 1 = r_1 r_2 \cdot \ldots \cdot r_w$, где сомножители r_i попарно взаимно просты, w ограничено, $\rho = \log \max\limits_i r_i$. Тогда при $k \to \infty$*

$$L(\Lambda_k) \lesssim \sum_{i=1}^{w} (2 + k/\log r_i) r_i, \quad D(\Lambda_k) \leqslant \rho + O(\log^2 k).$$

Доказательство. По построению,

$$L(\Lambda_k) \leqslant \sum_{i=1}^{w} L(\Lambda_{k,r_i}) + L(H_{k,w}) + L(R_{k,w}).$$

Пусть $r_{max} = \max\limits_i r_i$. Из $r_{max} \gtrsim 2^{k/w}$ следует, что оценка порядка сложности $\Lambda_{k,r_{max}}$ из теоремы 5

$$r_{max}(2 + k/\log r_{max}) \gtrsim O(w 2^{k/w}).$$

Сложность подсхем, реализующих $H_{k,w}$ и $R_{k,w}$, согласно леммам 2 и 3, оценивается как $O(wk^3)$. Эта величина является несущественной для асимптотики, если иметь в виду оценку сложности подсхемы, реализующей $N_{k,r_{max}}$.

Аналогично проверяется оценка для глубины,

$$D(\Lambda_k) \leqslant D(N_{k,r_{max}}) + D(H_{k,w}) + D(R_{k,w}) \leqslant \rho + O(\log^2 k).$$

Используя схему логарифмирования, описанную выше, мы ставим его эффективность в зависимость от существования «гладкого» *) разложения числа $2^k - 1$ в произведение взаимно простых сомножителей. Разложение тем эффективнее, чем меньше максимальный из сомножителей.

Для интерполяции требуется поле, содержащее не менее p элементов. Практически, среди нескольких полей со степенями $k \geqslant \lceil \log p \rceil$ необходимо выбрать поле с наиболее гладким порядком мультипликативной группы.

Например, поле $GF(2^9)$ менее гладкое, чем $GF(2^{10})$, так как $2^9 - 1 = 7 \cdot 73$, а $2^{10} - 1 = 3 \cdot 11 \cdot 31$. Но еще более гладким является поле $GF(2^{12})$, поскольку $2^{12} - 1 = 5 \cdot 7 \cdot 9 \cdot 13$.

Для того, чтобы оценить эффективность операции логарифмирования, рассмотрим несколько способов выбора гладкого поля при произвольном значении p.

В первом из них выбирается наименьшее четное из подходящих значений k, $k = 2l \geqslant \lceil \log p \rceil$, и используется разложение $2^{2l} - 1$ на всегда взаимно простые сомножители $2^l - 1$ и $2^l + 1$. Заметим, что одно из этих чисел делится на 3, откуда имеем

$$2^{2l} - 1 = \begin{cases} 3^d \dfrac{2^l - 1}{3^d}(2^l + 1), & l \text{ — четно;} \\ 3^d \dfrac{2^l + 1}{3^d}(2^l - 1), & l \text{ — нечетно,} \end{cases}$$

где 3^d — максимальная из степеней тройки, на которую делится $2^{2l} - 1$. Воспользовавшись теоремой 3, получим следующие оценки для схемы вычисления дискретного логарифма в поле $GF(2^{2l})$: сложность по порядку $(8 + o(1))2^l \lesssim (16 + o(1))\sqrt{p}$, глубина — $l + o(l) \lesssim (1/2)\log p$.

Лучшая схема получается при выборе поля $GF(2^k)$, где $k = 6l \geqslant \lceil \log p \rceil$ (выбирается наименьшее из возможных значений k). Взаимно простые сомножители $2^{3l} - 1$ и $2^{3l} + 1$ допускают дальнейшее разложение: $2^{3l} \pm 1 = (2^l \pm 1)(2^{2l} \mp 2^l + 1)$. Так как $2^{2l} \pm 2^l + 1 = 3 \bmod (2^l \mp 1)$, то 3 — это единственный общий делитель, который могут иметь множители в указанном разложении.

$$2^{6l} - 1 = \begin{cases} 3^{d+1} \dfrac{2^l - 1}{3^d}(2^l + 1)\dfrac{2^{2l} + 2^l + 1}{3}(2^{2l} - 2^l + 1), & l \text{ — четно;} \\ 3^{d+1} \dfrac{2^l + 1}{3^d}(2^l - 1)\dfrac{2^{2l} - 2^l + 1}{3}(2^{2l} + 2^l + 1), & l \text{ — нечетно.} \end{cases}$$

Сложность схемы логарифмирования в поле $GF(2^{6l})$ оценивается как $(20/3 + o(1))2^{2l} \lesssim (80/3 + o(1))\sqrt[3]{p}$, глубина — $2l + o(l) \lesssim (1/3)\log p$. Для $p > 2^8$ (что отражает пример поля $GF(2^{12})$) оценки, полученные вторым способом, лучше, чем в первом случае.

Дальнейшее развитие идеи использования разложения многочленов вида $x^k - 1$ на неприводимые многочлены над $\mathbb{Z}$ позволяет добиться

*) Разложение — «гладкое», если все его сомножители малы. Гладким также называют число, допускающее гладкое разложение на множители (см., например, [7]).

оценки сложности для схемы логарифмирования $C(\varepsilon)O(p^{\varepsilon})$ и глубины $\varepsilon \log p + o(\log p)$, $\varepsilon > 0$; этому посвящен следующий пункт.

4.4. Асимптотическая оценка эффективности логарифмирования. Рассмотрим поле $GF(2^{k_v \cdot l})$, где $k_v = p_1 p_2 \cdot \ldots \cdot p_v$, $\{p_i\}$ — возрастающая последовательность простых натуральных чисел.

Теорема 7. *Пусть $l \in \mathbb{N}$. Тогда число $2^{k_v \cdot l} - 1$ представимо в виде произведения попарно взаимно простых сомножителей $r_1, r_2, \ldots, r_s$, при этом*

$$\max_i r_i \leqslant 2^{\varphi(k_v) l} e^{\varphi(k_v)/2^l},$$

где $\varphi(k)$ — функция Эйлера.

Доказательство теоремы предварим несколькими вспомогательными утверждениями. Сначала обратимся к теории круговых многочленов.

Пусть $d \in \mathbb{N}$. Многочлен $F_d \in \mathbb{C}[x]$ минимально возможной степени со старшим коэффициентом 1 такой, что его корнями являются все примитивные корни*) степени d из единицы, называется *d-м круговым многочленом.*

Известны следующие свойства круговых многочленов (подробнее о круговых многочленах см. в [8]):

(1) $F_d \in \mathbb{Z}[x]$;

(2) $\deg F_d = \varphi(d)$;

(3) многочлены $\{F_d\}$ попарно взаимно просты;

(4) $x^h - 1 = \prod_{d \mid h} F_d(x)$.

Лемма 6. *Пусть $x \geqslant 1$, тогда $F_d(x) \leqslant x^{\varphi(d)} e^{\varphi(d)/x}$.*

Доказательство. Корни многочлена $F_d(x)$ по модулю равны 1, обозначим их через ξ_i, тогда

$$F_d(x) = \prod_{i=1}^{\varphi(d)} (x - \xi_i) < \prod_{i=1}^{\varphi(d)} (x + |\xi_i|) = (x+1)^{\varphi(d)} = x^{\varphi(d)} \left(1 + \frac{1}{x}\right)^{\varphi(d)} \leqslant x^{\varphi(d)} e^{\varphi(d)/x}.$$

Заметим, что если $x \to \infty$, то $F_d(x) = O(x^{\varphi(d)})$. Нам понадобится еще одна лемма о делимости чисел (она приведена в [12, задача 12]).

Лемма 7. *Пусть q — простое число, $a \in \mathbb{Z}$, тогда*

$$\text{НОД}(a-1, \tfrac{a^q - 1}{a-1}) = \text{НОД}((a-1)^2, \tfrac{a^q - 1}{a-1}) = \text{НОД}(a-1, q).$$

Доказательство. Дважды разделим многочлен $x^{q-1} + \ldots + 1 = \frac{x^q - 1}{x - 1}$ на $x - 1$ с остатком:

$$x^{q-1} + x^{q-2} + \ldots + 1 =$$
$$= (x-1)^2 \left(x^{q-3} + 3x^{q-4} + \ldots + \frac{(q-1)(q-2)}{2}\right) + (x-1)\frac{q(q-1)}{2} + q.$$

Подставим a вместо x. Последующая проверка отношений делимости не представляет труда.

Доказательство основной теоремы 7. Многочлен $x^{k_v} - 1$ раскладывается в произведение круговых многочленов (свойство (4))

$$x^{k_v} - 1 = \prod_{d \mid k_v} F_d(x).$$

*) Примитивным называется корень, не являющийся корнем никакой меньшей степени.

Количество сомножителей в этом произведении равно 2^v — они соответствуют делителям числа k_v.

Если вместо переменной x подставить число 2^l, то получится разложение

$$2^{k_v l} - 1 = \prod_{d \mid k_v} F_d(2^l) \qquad (*)$$

числа $2^{k_v l} - 1$ на множители, не превосходящие $2^{\varphi(k_v)l} e^{\varphi(k_v)/2^l}$, что следует из леммы 6, так как $\varphi(k_v)$ — максимальная из степеней многочленов F_d. Исследуем далее возможность преобразования данного разложения в произведение *попарно взаимно простых* сомножителей, не превосходящих $2^{\varphi(k_v)l} e^{\varphi(k_v)/2^l}$.

Рассмотрим следующие разложения многочлена $x^{k_v} - 1$ в произведение двух сомножителей (для краткости введем обозначение $y_i = x^{k_v/p_i}$):

$$x^{k_v} - 1 = y_i^{p_i} - 1 = (y_i - 1)(y_i^{p_i - 1} + \ldots + 1). \qquad (i)$$

Из свойства (4) круговых многочленов следует, что

$$y_i - 1 = \prod_{d \mid k_v,\ p_i \nmid d} F_d(x), \qquad y_i^{p_i - 1} + \ldots + 1 = \prod_{d \mid k_v,\ p_i \mid d} F_d(x),$$

поэтому любой многочлен $F_d(x)$, $d \mid k_v$, делит какой-либо из двух сомножителей в правой части каждого из разложений (i).

Покажем, что общими делителями значений двух круговых многочленов (при подстановке 2^l) могут быть только простые числа p_i, $i = 2, \ldots, v$.

Рассмотрим произвольную пару $F_{d_1}(2^l)$ и $F_{d_2}(2^l)$, где $d_1, d_2 \mid k_v$, пусть при этом $d_1 < d_2$. Тогда обязательно найдется такое число p_i, что $p_i \mid d_2$ и $p_i \nmid d_1$. Рассмотрим разложение (i). Многочлен $F_{d_1}(x)$ делит первый из сомножителей, а $F_{d_2}(x)$ — второй. Из леммы 7 следует, что

$$\text{НОД}\left((y_i - 1)\big|_{x=2^l},\ (y_i^{p_i - 1} + \ldots + 1)\big|_{x=2^l} \right) \in \{1, p_i\}.$$

Следовательно,

$$\text{НОД}(F_{d_1}(2^l),\ F_{d_2}(2^l)) \in \{1, p_i\}.$$

Выделив в отдельные сомножители $p_i^{c_i}$, $i = 2, \ldots, v$, где c_i — кратность p_i в произведении, получим разложение на взаимно простые множители. Осталось показать, что вновь образованные множители $p_i^{c_i}$ удовлетворяют оценке $2^{\varphi(k_v)l} e^{\varphi(k_v)/2^l}$.

Если p_i делит только один из сомножителей исходного разложения $(*)$, то доказывать нечего. Рассмотрим случай, когда p_i делит два сомножителя разложения $(*)$ $F_{d_1}(2^l)$ и $F_{d_2}(2^l)$, $d_1 \neq d_2$. Из леммы 7 следует, что в любом разложении (j), где $j \neq i$, многочлены $F_{d_1}(x)$ и $F_{d_2}(x)$ делят один и тот же сомножитель в правой части (иначе p_i не может быть общим делителем). Покажем, что в разложении (i) они делят различные сомножители.

Заметим, что d, произвольный делитель k_v, однозначно определяется, если про любое число p_s, $s = 1, \ldots, v$, известно, делит оно d или нет. Поэтому и многочлен $F_d(x)$ единственным образом определяется по своей принадлежности к одному из сомножителей в каждом разложении (s).

Из этого замечания вытекает, что если бы в разложении (i) (как и во всех остальных) многочлены $F_{d_1}(x)$ и $F_{d_2}(x)$ делили один и тот же сомножитель, то они бы совпадали, что противоречило бы условию $d_1 \neq d_2$. Следовательно, в разложении (i) они делят различные сомножители.

Предположим теперь, что еще один сомножитель $F_{d_3}(2^l)$ исходного разложения (*), отличный от указанных двух, делится на p_i. Рассуждая аналогично, заключаем, что многочлен $F_{d_3}(x)$ во всех разложениях за исключением (i) делит те же сомножители, что и пара $F_{d_1}(x)$, $F_{d_2}(x)$. Но тогда, в зависимости от того, в какой из сомножителей разложения (i) он входит, он совпадает либо с $F_{d_1}(x)$, либо с $F_{d_2}(x)$, а это противоречит предположению.

Таким образом, p_i может делить одновременно не более двух чисел из набора $F_d(2^l)$, $d \mid k_v$. При этом, как следует из леммы 7, если p_i делит ровно два сомножителя в разложении (*), то один из сомножителей (а именно тот, который соответствует многочлену, делящему $y_i - 1$ в разложении (i)) делится на $p_i^{c_i - 1}$. Следовательно, выделенный множитель $p_i^{c_i}$ может, самое большее, в p_i раз превосходить любой из сомножителей $F_d(2^l)$ исходного разложения, для которого выполнено: $d \mid k_v$ и $p_i \nmid d$. При этом $\varphi(d) \leqslant \varphi(k_v)/(p_i - 1)$. Используя лемму 6, имеем

$$p_i^{c_i} \leqslant p_i \cdot \max_{d \mid k_v,\ p_i \nmid d} F_d(2^l) \leqslant p_i 2^{\varphi(k_v)l/(p_i-1)} e^{\varphi(k_v)/(2^l(p_i-1))}.$$

Покажем, что

$$p_i 2^{\varphi(k_v)l/(p_i-1)} e^{\varphi(k_v)/(2^l(p_i-1))} < 2^{\varphi(k_v)l} e^{\varphi(k_v)/2^l},$$

где $1 < i \leqslant v$, $l \geqslant 1$. Отдельно рассмотрим случай $i = v = 2$, $l = 1$ (т. е. $p_i = 3$, $k_v = 6$). После подстановки параметров в неравенство, оно принимает вид $6\sqrt{e} < 4e$, что верно. Неравенство тем более остается верным при увеличении параметров v и (или) l. Если $i > 2$, то выполнено $p_i < 2^{\varphi(k_i)/2}$, что проверяется, например, следующим образом по индукции. При $i = 3$ неравенство выполнено в силу $5 < 2^4$. Если верно, что $p_{i-1} < 2^{\varphi(k_{i-1})/2}$, то

$$p_i < 2p_{i-1} < p_{i-1}^{p_i - 1} < 2^{\varphi(k_{i-1})(p_i-1)/2} = 2^{\varphi(k_i)/2}.$$

Мы воспользовались известным неравенством $p_i < 2p_{i-1}$ (постулат Бертрана). Из доказанного промежуточного неравенства выводим

$$p_i 2^{\varphi(k_v)l/(p_i-1)} e^{\varphi(k_v)/(2^l(p_i-1))} < 2^{\varphi(k_i)/2 + \varphi(k_v)l/(p_i-1)} e^{\varphi(k_v)/2^l} <$$
$$< 2^{\varphi(k_v)l(1/2 + 1/(p_i-1))} e^{\varphi(k_v)/2^l} < 2^{\varphi(k_v)l} e^{\varphi(k_v)/2^l}.$$

Тем самым теорема полностью доказана.

Заметим, что количество сомножителей в построенном разложении не превосходит $2^v + v - 1$, где 2^v — количество сомножителей в исходном разложении (*), плюс дополнительно выносится не более $v - 1$ сомножителей.

Для иллюстрации рассмотрим пример $v = 3$, $k_3 = 30$.

$$x^{30} - 1 = F_1 F_2 F_3 F_5 F_6 F_{10} F_{15} F_{30};$$
$$F_1(x) = x - 1, \quad F_2(x) = x + 1, \quad F_3(x) = x^2 + x + 1,$$
$$F_5(x) = x^4 + x^3 + x^2 + x + 1, \quad F_6(x) = x^2 - x + 1,$$
$$F_{10}(x) = x^4 - x^3 + x^2 - x + 1, \quad F_{15}(x) = x^8 - x^7 + x^5 - x^4 + x^3 - x + 1,$$
$$F_{30}(x) = x^8 + x^7 - x^5 - x^4 - x^3 + x + 1.$$

Пусть $l = 1$. При подстановке $x = 2$ получается разложение на множители для числа $2^{30} - 1$ (порядок сомножителей сохранен)

$$2^{30} - 1 = 1 \cdot 3 \cdot 7 \cdot 31 \cdot 3 \cdot 11 \cdot 151 \cdot 331.$$

Выделим в отдельные множители вхождения в произведение двух первых нечетных простых чисел: 3 (встречается два раза, $F_2(2) = F_6(2) = 3$) и 5 (нет). Окончательно, что гарантируется теоремой, имеем удовлетворяющее всем требованиям разложение

$$2^{30} - 1 = 7 \cdot 9 \cdot 11 \cdot 31 \cdot 151 \cdot 331,$$

которое в данном случае совпадает с каноническим разложением $2^{30} - 1$ на простые множители.

Используя результат теоремы 7, можно показать, что в поле $GF(2^{30l})$ логарифмирование выполняется со сложностью $O(2^{8l})$, что при $2^{30l} > p \geqslant 2^{30(l-1)}$ соответствует оценке $O(p^{4/15})$. Однако мультипликативная константа в этом случае слишком велика.

Далее мы будем пользоваться следующим фактом:

$$\frac{c_1}{\log(v+1)} < \frac{\varphi(k_v)}{k_v} = \prod_{i=1}^{v} \frac{p_i - 1}{p_i} < \frac{c_2}{\log v} \to 0, \quad \text{при } v \to \infty.$$

Легко показать, что $c_1 > e^{-5/2}$, а $c_2 < \ln 2 = 0.693\ldots$ Более точные оценки приведены в работе [31] *).

Теорема 8. *Пусть $v \in \mathbb{N}$, $p \geqslant 2^{k_v}$. Тогда существует поле характеристики 2, содержащее не менее p элементов, в котором сложность логарифмирования асимптотически (при $p \to \infty$) не превосходит*

$$\log(v+1) 2^{\varphi(k_v)+v+4} e^{\varphi(k_v)p^{-1/k_v}} p^{\varphi(k_v)/k_v},$$

а глубина не превосходит

$$(\varphi(k_v)/k_v)\log p + 2\varphi(k_v) + O(\log^2 \log p).$$

Доказательство. Для заданного p рассмотрим поле $GF(2^{k_v l})$, где l удовлетворяет соотношению $2^{k_v(l-1)} < p \leqslant 2^{k_v l}$. Согласно теореме 7, число $2^{k_v l} - 1$ раскладывается в произведение взаимно простых сомножителей, не превосходящих $2^{\varphi(k_v)l} e^{\varphi(k_v)/2^l}$.

С помощью теоремы 6 оценим глубину схемы логарифмирования в поле $GF(2^{k_v l})$,

$$D(\Lambda_{k_v l}) \leqslant \log((2^l e^{1/2^l})^{\varphi(k_v)}) + O(\log^2(k_v l)) <$$
$$< \varphi(k_v)(l+1) + O(\log^2 \log p) < \varphi(k_v)(2 + (\log p)/k_v) + O(\log^2 \log p) <$$
$$< (\varphi(k_v)/k_v)\log p + 2\varphi(k_v) + O(\log^2 \log p).$$

Та же теорема 6 позволяет оценить порядок сложности схемы,

$$L(\Lambda_{k_v l}) \lesssim (2^v + v)(2 + k_v/\log r_{max}) r_{max} \leqslant (2^v + v)(2 + k_v/\varphi(k_v)) 2^{\varphi(k_v)l} e^{\varphi(k_v)/2^l}.$$

Справедливо неравенство:

$$(2^v + v)(2 + k_v/\varphi(k_v)) < 2^{v+4}\log(v+1),$$

---

*) Согласно теореме Мертенса это выражение имеет асимптотику $\frac{e^\gamma}{\ln v}$, где γ — постоянная Эйлера (см. также [31]).

которое проверяется при $v=1,\ldots,4$ непосредственной подстановкой, а при $v\geqslant 5$ можно воспользоваться оценкой $k_v/\varphi(k_v)<e^{5/2}\log(v+1)$, тогда

$$(2^v+v)(2+k_v/\varphi(k_v))<\log(v+1)2^v\left[\left(1+\frac{v}{2^v}\right)\left(e^{5/2}+\frac{2}{\log(v+1)}\right)\right].$$

Выражение в квадратных скобках является монотонно убывающей функцией от v, и из того, что при $v=5$ его значение меньше 16, следует заявленное неравенство.

Далее,

$$2^{\varphi(k_v)l}=2^{\varphi(k_v)}(2^{k_v(l-1)})^{\varphi(k_v)/k_v}<2^{\varphi(k_v)}p^{\varphi(k_v)/k_v}.$$

Наконец, так как $2^l\geqslant p^{-1/k_v}$, имеем

$$e^{\varphi(k_v)/2^l}\leqslant e^{\varphi(k_v)p^{-1/k_v}}.$$

Комбинируя все указанные неравенства, получаем требуемый результат.

Следствие 3. *Пусть $\varepsilon>0$. Тогда существует поле характеристики 2, содержащее не менее p элементов, в котором логарифмирование выполняется со сложностью (при $p\to\infty$), не превосходящей $C_1(\varepsilon)O(p^\varepsilon)$ и глубиной $\varepsilon\log p+C_2(\varepsilon)+O(\log^2\log p)$.*

Доказательство. Выберем v такое, что $\frac{\varphi(k_v)}{k_v}\leqslant\varepsilon$ (например, подойдет $v=\lceil 2^{0.7/\varepsilon}\rceil$), и применим доказанную теорему.

§ 5. Уточнение оценки сложности

Рассмотрим другой способ вычисления x^{-1}. Число $n-1$ представим в виде $n_1n_2+n_3$, где $0\leqslant n_i\leqslant\lceil\sqrt{n-1}\rceil$. Из этого представления выведем представление числа 2^n-2 в виде $N_1N_2+N_3$, где

$$N_1=2+2^2+\ldots+2^{n_1},\quad N_2=1+2^{n_1}+2^{2n_1}+\ldots+2^{(n_2-1)n_1},$$
$$N_3=2^{n_1n_2+1}+2^{n_1n_2+2}+\ldots+2^{n_1n_2+n_3}.$$

Вес каждого из чисел N_i равен n_i, $i=1,2,3$.

Воспользуемся формулой

$$x^{-1}=(x^{N_1})^{N_2}x^{N_3},$$

которая показывает, что инвертирование сводится к трем возведениям в степень с относительно небольшим весом и одному умножению в поле $GF(2^n)$. Более формально,

$$\boldsymbol{I}_n(x)=\boldsymbol{M}_n(\boldsymbol{E}_{n,n_2}\cdot\boldsymbol{E}_{n,n_1}(x),\boldsymbol{E}_{n,n_3}(x)).$$

Реализуя операции $\boldsymbol{E}_{n,n_i}$ алгоритмом из § 3, оценим сложность такой схемы как

$$L(\boldsymbol{I}_n)\leqslant L(\boldsymbol{M}_n)+\sum_{i=1}^{3}L(\boldsymbol{E}_{n,n_i})\lesssim\sum_{i=1}^{3}\frac{\log(n_in)}{\log(n_i^2n)}n_i^2n^2\leqslant\frac{9}{4}n^3.$$

Глубина схемы оценивается как

$$D(\boldsymbol{I}_n)\leqslant D(\boldsymbol{M}_n)+\max\{D(\boldsymbol{E}_{n,n_2})+D(\boldsymbol{E}_{n,n_1}),D(\boldsymbol{E}_{n,n_3})\}\lesssim$$
$$\lesssim 2\log n+(4+\varepsilon)\log n+4.44\log n\lesssim 10.44\log n.$$

Этот прием допускает обобщение, приводящее к следующему результату.

Теорема 9. *Пусть* $r \in \mathbb{N}$ *и* $q = \lceil \sqrt[r]{n} \rceil$. *Тогда сложность и глубина схемы инвертирования в поле* $GF(2^n)$ *оцениваются как*

$$L(\boldsymbol{I}_n) \leqslant (2r-1)L(\boldsymbol{E}_{n,q}) + (r-1)L(\boldsymbol{M}_n),$$
$$D(\boldsymbol{I}_n) \leqslant 2D(\boldsymbol{E}_{n,q}) + D(\boldsymbol{M}_n) + (r-2)\max\{D(\boldsymbol{E}_{n,q}), D(\boldsymbol{M}_n)\}.$$

Доказательство. Пусть $n-1 = [m_r, m_{r-1}, \ldots, m_1]$ в системе счисления с основанием q, т. е.,

$$n-1 = q^{r-1}m_r + q^{r-2}m_{r-1} + \ldots + qm_2 + m_1,$$

где все m_i не превосходят $\sqrt[r]{n}$.

Положим

$$N_i = 1 + 2^{q^{i-1}} + 2^{2q^{i-1}} + \ldots + 2^{(q-1)q^{i-1}} = \frac{2^{q^i}-1}{2^{q^{i-1}}-1},$$

тогда

$$N_1 N_2 \cdot \ldots \cdot N_i = 2^{q^i} - 1.$$

Пусть $M_i = 0$, когда $m_i = 0$, и

$$M_i = 2^{[m_r, \ldots, m_{i+1}, 0, \ldots, 0]+1}(1 + 2^{q^{i-1}} + \ldots + 2^{(m_i-1)q^{i-1}})$$

в противном случае. Тогда справедливо

$$(2^{q^{i-1}} - 1)M_i = 2(2^{[m_r, \ldots, m_i, 0, \ldots, 0]} - 2^{[m_r, \ldots, m_{i+1}, 0, \ldots, 0]}),$$

где в квадратных скобках записывается r-разрядное число в q-ичной системе счисления. Суммируя эти равенства по $i = 1, \ldots, r$, получаем

$$2^n - 2 = N_1 \cdot \ldots \cdot N_{r-1}M_r + N_1 \cdot \ldots \cdot N_{r-2}M_{r-1} + \ldots + N_1 M_2 + M_1,$$

где вес каждого из чисел N_i равен q, а вес M_i равен m_i.

Схему инвертирования построим, основываясь на формуле

$$x^{-1} = x^{2^n-2} = \left(\left(\ldots\left((x^{N_1})^{N_2}\right)\ldots\right)^{N_{r-1}}\right)^{M_r} \times$$
$$\times \left(\left(\ldots\left((x^{N_1})^{N_2}\right)\ldots\right)^{N_{r-2}}\right)^{M_{r-1}} \cdot \ldots\ldots \cdot (x^{N_1})^{M_2} x^{M_1}.$$

Отобразим поэтапно последовательность вычислений.

0.	x;			
1.	$x_1 = x^{N_1}$,	$y_1 = x^{M_1}$;		
2.	$x_2 = x_1^{N_2}$,	$y_2 = x_1^{M_2}$,	$z_2 = y_1$;	
$i = 3, \ldots, r-1$.	$x_i = x_{i-1}^{N_i}$,	$y_i = x_{i-1}^{M_i}$,	$z_i = z_{i-1}y_{i-1}$;	
r.		$y_r = x_{r-1}^{M_r}$,	$z_r = z_{r-1}y_{r-1}$;	
$r+1$.			$x^{-1} = z_r y_r$.	

Схема вычислений включает не более $2r-1$ подсхем, реализующих возведения в степени с весами, не превосходящими q и не более $r-1$ подсхем, реализующих умножения в поле. Глубину уровней 1 и 2 схемы можно оценить как $D(\boldsymbol{E}_{n,q})$, уровней $3, \ldots, r$ — как $\max\{D(\boldsymbol{E}_{n,q}), D(\boldsymbol{M}_n)\}$

и уровня $r+1$ — как $D(\boldsymbol{M}_n)$, откуда следует заявленная в условии теоремы оценка глубины.

Следствие 4. *Пусть $r \in \mathbb{N}$ и $q = \lceil \sqrt[r]{n} \rceil$. Тогда сложность и глубина схемы деления в поле $GF(2^n)$ оцениваются как*

$$L(\Delta_n) \leqslant (2r-1)L(\boldsymbol{E}_{n,q}) + rL(\boldsymbol{M}_n),$$
$$D(\Delta_n) \leqslant D(\boldsymbol{E}_{n,q}) + D(\boldsymbol{M}_n) + (r-1)\max\{D(\boldsymbol{E}_{n,q}), D(\boldsymbol{M}_n)\}.$$

Доказательство. Для вычисления y/x достаточно умножение на y встроить в схему, вычисляющую x^{-1}, из теоремы 9:

0.	x,	y;		
1.	$x_1 = x^{N_1}$,	$y_1 = x^{M_1}$,	$z_1 = y$;	
$i = 2, \ldots, r-1$.	$x_i = x_{i-1}^{N_i}$,	$y_i = x_{i-1}^{M_i}$,	$z_i = z_{i-1} y_{i-1}$;	
r.		$y_r = x_{r-1}^{M_r}$,	$z_r = z_{r-1} y_{r-1}$;	
$r+1$.			$y/x = z_r y_r$.	

В естественном предположении $D(\boldsymbol{E}_{n,q}) \geqslant D(\boldsymbol{M}_n)$ глубина построенных схем инвертирования и деления оценивается как

$$D(\boldsymbol{I}_n),\ D(\Delta_n) \leqslant rD(\boldsymbol{E}_{n,q}) + D(\boldsymbol{M}_n).$$

Подставляя в теорему 9 и следствие 4 оценки теоремы 2, и используя оценки $L(\boldsymbol{M}_n) = O(n^2)$, $D(\boldsymbol{M}_n) \leqslant (2+o(1))\log n$, получаем

Следствие 5. *Пусть $r \in \mathbb{N}$. Тогда для инвертирования и деления в поле $GF(2^n)$ можно построить схемы со сложностью и глубиной (при $n \to \infty$)*

$$L(\boldsymbol{I}_n),\ L(\Delta_n) \leqslant \left(2r - 3 + \frac{5}{r+2} + o(1)\right) n^{2+\frac{2}{r}},$$
$$D(\boldsymbol{I}_n),\ D(\Delta_n) \leqslant (2r + 6.44 + o(1))\log n.$$

Таким образом, построена схема для инвертирования логарифмической глубины и почти квадратичной сложности. В следующем параграфе будет показано, что можно строить схемы с глубиной $O(\log n)$ и сложностью $o(n^2)$.

§ 6. Алгоритм субквадратичной сложности

Приведем краткое описание модифицированного алгоритма возведения элемента $x \in GF(2^n)$ в степень $M = \sum_i 2^{e_i}$ веса m.

1. Вычислим все $x^{2^{e_i}} = f_i(t)$, $i = 1, \ldots, m$. Пусть $f(t) = f_1(t) \cdot \ldots \cdot f_m(t)$. Положим $p = m(n-1)+1$, выберем поле $GF(2^k)$, содержащее не менее p элементов; в нем выберем набор элементов $\alpha_1, \ldots, \alpha_p$.

2. Вычислим всевозможные $f_i(\alpha_j) \in GF(2^k)$, где $i = 1, \ldots, m$, $j = 1, \ldots, p$.

3. Для всех j вычислим произведения $f_1(\alpha_j) \cdot \ldots \cdot f_m(\alpha_j) = f(\alpha_j)$.

4. По значениям $f(\alpha_j)$, $j = 1, \ldots, p$, восстанавливается соответствующий элементу x^M многочлен $f(t) \bmod m_n(t)$, $\deg f < p$.

В данном алгоритме полностью сохраняется общая схема вычислений первоначального алгоритма из § 3 — изменяется только способ реализации. Для реализации линейных шагов 1, 2 будет применяться так называемое матричное ускорение. Будет рассмотрена другая реализация шага 4 (шаги 4, 5

исходного алгоритма). Способ реализации шага 3 останется без изменений. Напомним, что, согласно следствию 3, шаг 3 выполняется схемой сложности $O(m^{2+\varepsilon}n^{1+\varepsilon})$ и глубиной $(\varepsilon+o(1))\log(mn)$, где $\varepsilon=const>0$.

6.1. Обобщенная модулярная композиция. Как известно (см., например, [9]), оценка $O(ns/\log n)$ сложности линейного отображения размерности $n\times s$ является неулучшаемой в общем случае. Однако, она может быть улучшена для отображений, удовлетворяющих дополнительным ограничениям. Рассмотрим *гомоморфные* отображения, сохраняющие, помимо операции сложения, также операцию умножения.

Следующая лемма является простым обобщением результата Брента и Кунга [18] о сложности модулярной композиции (композиции двух многочленов по модулю третьего). Через $T_{q,r,s}$ обозначим операцию умножения двоичных матриц размера $q\times r$ и $r\times s$.

Лемма 8. *Пусть G — гомоморфизм из $GF(2)[t]$ во множество V, имеющее структуру векторного пространства размерности s над $GF(2)$ с операцией умножения. Обозначим через G_n сужение гомоморфизма G на множество многочленов степени не выше $n-1$, пусть также $rq\geqslant n$. Тогда*

$$L(G_n)\leqslant L(T_{q,r,s})+(q-1)(L(M_V)+s),\quad D(G_n)\leqslant D(T_{q,r,s})+D(M_V)+\lceil\log q\rceil,$$

где M_V — операция умножения в V.

Доказательство. Пусть $f(t)\in GF(2)[t]$, $\deg f\leqslant n-1$. Запишем,

$$f(t)=f_0(t)+f_1(t)t^r+\ldots+f_{q-1}(t)t^{(q-1)r},$$

где $\deg f_i<r$. По условию,

$$G_n(f)=G_r(f_0)+G_r(f_1)G_n(t^r)+\ldots+G_r(f_{q-1})G_n(t^{(q-1)r}),$$

где G_r — сужение отображения G на множество многочленов степени, меньшей, чем r. В частности, G_r является линейным отображением размерности $r\times s$. Вычисление всех $G_r(f_i)$, $i=0,\ldots,q-1$, соответствует умножению $q\times r$-матрицы коэффициентов многочленов $f_i(t)$ на $r\times s$-матрицу коэффициентов $G_r(t^j)\in V$.

Считая, что все $G_n(t^{ir})$, $i=1,\ldots,q-1$, вычислены предварительно (это так при схемной реализации), для вычисления $G_n(f)$ остается выполнить $q-1$ умножений и сложить q векторов из V.

Следствие 6 (Брент, Кунг, 1978). *Пусть*

$$C_{g,h}(f)=f(g(t))\bmod h(t),$$

где $g(t)$, $h(t)$ — фиксированные многочлены степени $n-1$ и n соответственно, пусть также $rq\geqslant n$. Тогда

$$L(C_{g,h})\leqslant L(T_{q,r,n})+(q-1)(L(M_n)+n),$$
$$D(C_{g,h})\leqslant D(T_{q,r,n})+D(M_n)+\lceil\log q\rceil.$$

Частным случаем модулярной композиции является операция Фробениуса: возведение элемента $x\in GF(2^n)$ в степень вида 2^l. Действительно, пусть $x=f(t)$ в полиномиальном представлении, тогда

$$f^{2^l}(t)\bmod m_n(t)=f(t^{2^l})\bmod m_n(t)=$$
$$=f(t^{2^l}\bmod m_n(t))\bmod m_n(t)=f(\xi_l(t))\bmod m_n(t),$$

где $\xi_l(t) = t^{2^l} \bmod m_n(t)$. Операция Фробениуса (обозначаем ее через $S_{n,l}$) является автоморфизмом поля $GF(2^n)$.

Лемма 9. *Операция Фробениуса в поле $GF(2^n)$ реализуется со сложностью и глубиной*

$$L(S_{n,l}) = O(n^{1.667}), \qquad D(S_{n,l}) = O(\log n).$$

Доказательство. В оценки следствия 6 подставим $q, r \sim \sqrt{n}$. Из алгоритма умножения Шёнхаге [32] имеем $L(M_n) = O(n \log n \log\log n)$ и $D(M_n) = O(\log n)$. Операция $T_{q,r,n}$, т. е. умножение матрицы размера $\sqrt{n} \times \sqrt{n}$ на матрицу размера $\sqrt{n} \times n$, выполняется, как показано в работе [27], со сложностью $O(n^{1.667})$. Кроме того, известно (см., например, [17, раздел 4.3]), что любой метод матричного умножения допускает реализацию схемой логарифмической глубины с увеличением порядка сложности на n^ε. Для завершения доказательства выберем ε в пределах погрешности округления константы из [27] до 1.667.

Из леммы 9 следует, что сложность реализации шага 1 алгоритма составляет $O(mn^{1.667})$.

Значение многочлена в фиксированной точке также получается действием гомоморфного преобразования. Пусть, как и прежде, $p = m(n-1)+1$, $m < n$, и задан набор $\{\alpha_1, \ldots, \alpha_p\} \subset GF(2^k)$. Через $C^m_{k,n}$ обозначим операцию вычисления значений m многочленов степени не выше $n-1$ в точках α_i.

Лемма 10. *Операция $C^m_{k,n}$ реализуется схемой сложности и глубины*

$$L(C^m_{k,n}) = O((mn)^{1.667}k) + O((mn)^{1.5})L(M_k), \qquad D(C^m_{k,n}) = O(\log(nk)).$$

Доказательство. Отображение $C^m_{k,n}$ можно рассматривать как объединение m отображений C_{k,n,α_i} в принятых в § 2 обозначениях. Применим лемму 8, полагая $G_n = C^m_{k,n}$ и $V = GF(2^k)^p$. Заметим, что вместо m независимых матричных умножений типа $T_{q,r,pk}$ достаточно выполнить одно умножение типа $T_{qm,r,pk}$, поскольку матрица коэффициентов $G_r(t^j)$ является одной и той же для всех матричных произведений.

Получаем оценки

$$L(C^m_{k,n}) \leqslant L(T_{qm,r,pk}) + m(q-1)(L(M_V) + pk),$$

$$D(C^m_{k,n}) \leqslant D(T_{qm,r,pk}) + D(M_V) + \lceil \log q \rceil,$$

где M_V — покомпонентное умножение над $GF(2^k)$. Таким образом, $L(M_V) \leqslant pL(M_k)$, $D(M_V) = D(M_k)$. Положим $r \sim \sqrt{p}$, $q \sim n/r$. Рассматривая $T_{qm,r,pk}$ как выполнение k умножений матриц размера $\sqrt{mn} \times \sqrt{mn}$ на матрицы размера $\sqrt{mn} \times mn$, получаем окончательно требуемые оценки.

Из леммы 10 следует оценка сложности $O((mn)^{1.667}k)$ шага 2 алгоритма.

6.2. Модулярная интерполяция. Перейдем к шагу 4. В ранее введенных обозначениях на шаге 4 выполняется операция $B_{n,p} \cdot F^{-1}_{k,p}$, где $F^{-1}_{k,p}$ — операция восстановления коэффициентов многочлена над $GF(2)$ степени не выше $p-1$ по его значениям на наборе из p элементов поля $GF(2^k)$, а $B_{n,p}$ — приведение многочлена степени $p-1$ по модулю $m_n(t)$. Излагаемая далее конструкция фактически является модификацией алгоритма [1, п. 8.7].

Лемма 11. *Пусть* $r=\lceil p/q\rceil$, $sq\leqslant n$. *Тогда*

$$L(B_{n,p}\cdot F_{k,p}^{-1})\leqslant O\left(\frac{k^2mnq}{\log q}\right)+O(rs)L(M_{q,k})+O\left(\frac{rn}{s^2q}\right)L(M_{sq,k})+2L(M_n),$$

$$D(B_{n,p}\cdot F_{k,p}^{-1})\leqslant O(\log(kn))+D(M_{q,k})+D(M_{sq,k}),$$

где $M_{q,k}$ *— операция умножения многочленов степени* $q-1$ *над* $GF(2^k)$.

Доказательство. Согласно интерполяционной формуле Лагранжа,

$$F_{k,p}^{-1}(f(\alpha_1),\ldots,f(\alpha_p))=f(t)=\sum_{i=1}^{p}f(\alpha_i)l_i(t),$$

где $l_i(t)$ — фундаментальные многочлены Лагранжа, коэффициенты которых зависят только от постоянных величин $\alpha_1,\ldots,\alpha_p$,

$$l_i(t)=\prod_{j\neq i}\frac{t-\alpha_j}{(\alpha_i-\alpha_j)}.$$

Разобьем набор $\{\alpha_1,\ldots,\alpha_p\}$ на поднаборы A_1, ..., A_r, по q значений в каждом (за исключением, может быть, последнего — но далее для удобства будем считать, что $|A_r|=q$). Представим $f(t)$ в виде

$$f(t)=\sum_{i=1}^{r}\varphi_i(t)\lambda_i(t),$$

где

$$\varphi_i(t)=\sum_{\alpha_l\in A_i}f(\alpha_l)\frac{\prod_{j\neq l,\,\alpha_j\in A_i}(t-\alpha_j)}{\prod_{j\neq l}(\alpha_l-\alpha_j)},\qquad \lambda_i(t)=\prod_{\alpha_j\notin A_i}(t-\alpha_j).$$

Заметим, что коэффициенты любого многочлена $\varphi_i(t)$ есть линейные комбинации относительно $\{f(\beta)\mid\beta\in A_i\}$, $\deg\varphi_i\leqslant q-1$, а многочлены $\lambda_i(t)$ фиксированы.

Пусть $v=\lceil r/s\rceil$ (но для удобства примем, что $r=sv$). Положим

$$\Lambda_j(t)=\text{НОД}(\lambda_{js+1}(t),\ \lambda_{js+2}(t),\ldots,\lambda_{(j+1)s}(t)),\qquad j=0,\ldots,v-1.$$

Обозначим $\mu_{js+l}(t)=\lambda_{js+l}(t)/\Lambda_j(t)$, для всех $l=1,\ldots,s$ и $j=0,\ldots,v-1$. Очевидно, что $\deg\mu_i=(s-1)q$. Имеем

$$f(t)=\sum_{j=0}^{v-1}\Lambda_j(t)\sum_{l=1}^{s}\varphi_{js+l}(t)\mu_{js+l}(t).\qquad(\triangle)$$

Окончательно,

$$f(t)\bmod m_n(t)=\sum_{j=0}^{v-1}(\Lambda_j(t)\bmod m_n(t))\sum_{l=1}^{s}\varphi_{js+l}(t)\mu_{js+l}(t)\bmod m_n(t).$$

Рассмотрим следующую последовательность вычислений.

4.1. Вычисляются все $\varphi_i(t)$, $i=1,\ldots,r$ при помощи линейных операторов размерности $kq\times kq$.

4.2. Вычисляются произведения $\varphi_i(t)\mu_i(t)$, каждое из которых (разбив на части многочлен $\mu_i(t)$ более высокой степени) можно выполнить при помощи $s-1$ умножений многочленов степени $q-1$ (операций $\boldsymbol{M}_{q,k}$) с последующим приведением подобных.

4.3. Выполняются умножения многочленов $\Lambda_j(t) \bmod m_n(t)$ на соответствующие им суммы произведений $\sum_l \varphi_{js+l}(t)\mu_{js+l}(t)$, вычисленных на предыдущем шаге, $j=0,\ldots,v-1$. Каждое из умножений производится посредством $\lceil n/sq \rceil$ операций $\boldsymbol{M}_{sq,k}$ с последующим приведением подобных.

4.4. Складывая все вычисленные на предыдущем этапе многочлены, получаем многочлен степени не выше $2n-1$ с коэффициентами из $GF(2)$, что гарантируется условиями леммы. Приведение его по модулю $m_n(t)$ выполняется при помощи двух умножений и сложения многочленов степени $n-1$ (см. § 2.3).

Слагаемые в заявленной леммой оценке сложности соответствуют шагам этого алгоритма в том же порядке. Этими слагаемыми также поглощается сложность выполняемых на разных этапах операций сложения. То же справедливо и в отношении глубины (глубина шага 4.4 учитывается в первом члене).

Выбор параметров q, r, s зависит от алгоритма умножения многочленов над $GF(2^k)$. Применительно к построению схемы возведения в степень логарифмической глубины, необходимо ограничение на глубину алгоритма, $D(\boldsymbol{M}_{n,k}) = O(\log(kn))$ (тогда общая глубина алгоритма леммы 11 составит $O(\log(kn)))$.

Рассмотрим известные алгоритмы умножения многочленов степени $n-1$ с глубиной $O(\log n)$ над полем коэффициентов.

Очевидно, глубину $O(\log kn)$ имеет стандартный алгоритм, поскольку все умножения в нем производятся на одном уровне (а операции сложения в поле $GF(2^k)$ выполняются с единичной глубиной, как и в $GF(2)$). То же справедливо и в отношении метода Карацубы [5], равно как и для более общего метода Тоома с оценкой сложности $O(n^{\log_{l+1}(2l+1)})$ операций над $GF(2^k)$, $l \in \mathbb{N}$ (см. [13]).

Заметим, что уже применяя метод Карацубы, можно получить приемлемую оценку сложности для шага 4. Действительно, пусть $L(\boldsymbol{M}_{n,k}) = O(n^{\log 3} k^2)$ (для умножения в $GF(2^k)$ используем стандартный алгоритм). Выберем параметры, исходя из условия $q \sim sq^{\log 3 - 1} \sim n(sq)^{\log 3 - 2}$. Находим

$$q \sim n^{1/(1+(2-\log 3)(3-\log 3))} \sim n^{0.63}, \qquad s \sim n^{(2-\log 3)/(1+(2-\log 3)(3-\log 3))} \sim n^{0.26},$$

откуда следует, что шаг 4 может быть реализован со сложностью $O(mk^2 n^{1.631})$.

Выведем, однако, более сильную оценку, используя метод Шёнхаге [32]. Этот метод основан на преобразовании Фурье, из-за чего все умножения также выполняются на одном уровне, и глубина удовлетворяет оценке $O(\log kn)$.

В оценки леммы 10 подставим $\boldsymbol{M}_{n,k} = O(nk^2 \log n \log\log n)$. Из условия $q \sim s \sim n/qs$ находим, что $q, s \sim \sqrt[3]{n}$. Получаем следующий результат.

Лемма 12. *Операция $\boldsymbol{B}_{n,p} \cdot \boldsymbol{F}_{k,p}^{-1}$ реализуется со сложностью и глубиной*

$$L(\boldsymbol{B}_{n,p} \cdot \boldsymbol{F}_{k,p}^{-1}) \leqslant O(mn^{4/3} k^2 \log n \log\log n), \qquad D(\boldsymbol{B}_{n,p} \cdot \boldsymbol{F}_{k,p}^{-1}) = O(\log(kn)).$$

Замечание. Аккуратнее выбирая параметры, можно получить оценку для сложности в виде

$$O\left(m(\log\log n+\log k\log\log k)\sqrt[3]{k^4n^4\log n/\log\log n}\right).$$

Заметим также, что разложение формулы (Δ) можно итерировать, группируя многочлены $\Lambda_j(t)$ и выделяя у них общий множитель, и т. д. Так для любого натурального числа $d\geqslant 3$ можно построить схему сложности $O(mn^{1+1/d}k^2)$ и глубины $O(d\log(kn))$. Описанная конструкция фактически получается применением метода работы [26], в которой рассматривается аналогичная числовая операция, к многочленам.

Выбор $k=\log(mn)+O(1)$ и применение леммы 12 вместе с доказанными ранее следствием 3 и леммами 9 и 10 к алгоритму, приведенному в начале этого параграфа, приводит к следующему результату.

Теорема 10. *Операция возведения в степень веса m в поле $GF(2^n)$ реализуется схемой сложности и глубины*

$$L(\boldsymbol{E}_{n,m})=O((mn)^{w+\delta}\log n+m^{2+\delta}n^{1+\delta}),\qquad D(\boldsymbol{E}_{n,m})=O(\log n),$$

где w — экспонента умножения матриц размера $\sqrt{n}\times\sqrt{n}$ и $\sqrt{n}\times n$, а δ — произвольная положительная константа. В частности, $L(\boldsymbol{E}_{n,m})=O((mn)^{1.667})$.

Применение данного алгоритма в сочетании с методом теоремы 9 позволяет сформулировать главный результат этого параграфа.

Теорема 11. *Пусть $r=const\in\mathbb{N}$. Тогда операции инвертирования и деления в поле $GF(2^n)$ реализуются схемами со сложностью и глубиной*

$$L(\boldsymbol{I}_n),L(\boldsymbol{\Delta}_n)=O(rn^{w+(w+1)/r}\log n+n^{1+3/r}),\qquad D(\boldsymbol{I}_n),D(\boldsymbol{\Delta}_n)=O(r\log n).$$

В частности,

$$L(\boldsymbol{I}_n),L(\boldsymbol{\Delta}_n)=O(n^{1.667}),\qquad D(\boldsymbol{I}_n),D(\boldsymbol{\Delta}_n)=O(\log n).$$

Мультипликативные постоянные при главном члене, которые могут быть указаны для последней оценки, очень велики (по меньшей мере, десятки тысяч), поэтому практического значения предложенный алгоритм не имеет. Впрочем, умножая матрицы методом Штрассена [33] (см. также [6, п. 4.6.4]) с показателем 1.904 в оценке сложности и применяя теорему 9 с параметром $r=20$, получаем схему сложности $O(n^{1.999})$ и глубины $O(\log n)$ с мультипликативными постоянными при главном члене порядка нескольких сотен. Однако, и такой алгоритм неэффективен в имеющих прикладное значение полях.

§ 7. Замечания

7.1. О методах Литоу—Давида и фон цур Гатена. Инвертирование по методу [29], основанное на матричном представлении элементов поля, включает восстановление коэффициентов многочлена степени n по его корням, которые кодируются $O(n^2)$ битами, что означает вычисление элементарных симметрических функций от n чисел, содержащих $O(n^2)$ разрядов каждое. В частности, вычисляется произведение этих n чисел.

Метод [21], также использующий матричное представление, предназначен для конечного поля наиболее общего вида $GF(q^n)$. В случае $q=2$ элементарный алгоритм инвертирования можно непосредственно вывести из результата работы [19]. Как и выше, запишем

$$x^{-1} = x^{2^n-2} = x^2 x^{2^2} \cdot \ldots \cdot x^{2^{n-1}}.$$

Произведение соответствующих элементам x^{2^i} многочленов $f_i(t)$ сводится, согласно [19], к вычислению числового произведения

$$f_1(2^L) f_2(2^L) \cdot \ldots \cdot f_{n-1}(2^L),$$

где L приблизительно равно $n \log n$. Таким образом, требуется перемножить $n-1$ чисел, содержащих порядка $n^2 \log n$ разрядов.

Как в [29], так и в [19] предлагается использовать результат из [16]. Сложность схемы, выполняющей с логарифмической глубиной перемножение n чисел, кодирующихся n битами, в работе [16] оценивается как $O(n^5 \log^2 n)$. Мультипликативный коэффициент в оценке глубины не приводится, но элементарный анализ показывает, что он не меньше 15.

7.2. Об инвертировании в нормальных базисах. Корни неприводимого двоичного многочлена степени n в поле $GF(2^n)$ образуют нормальную систему $\{\alpha, \alpha^2, \ldots, \alpha^{2^{n-1}}\}$, которая, если ее элементы линейно независимы, является базисом поля. Такие базисы называются *нормальными*. Известно, что нормальных базисов достаточно много — по крайней мере, один такой базис имеется в любом поле (см. [3, 8, 28]).

Нормальные базисы изучались еще в 19-м веке, однако практический интерес к ним появился только в конце 20-го, с развитием криптографии конечных полей. В пользу применения нормальных базисов говорит исключительная простота реализации операций Фробениуса, которые вообще не требуют схемных затрат, так как сводятся к циклическому сдвигу коэффициентов (это видно прямо из определения). Однако, сложнее дело обстоит с умножением. Стандартный алгоритм Месси—Омура (см., например, [3, 28]) имеет теоретическую оценку сложности $O(n^3)$, которая даже в случае оптимального с точки зрения данного алгоритма выбора базиса имеет порядок n^2. Для некоторых конкретных видов базисов оценка сложности понижается до $O(n \log n \log\log n)$ (см. [20]), однако такие базисы существуют не во всех полях.

Поэтому практический интерес приобретает задача перехода к стандартному базису, в котором умножение выполняется сравнительно просто — теоретически не сложнее, чем за $O(n \log n \log\log n)$ операций, как следует из [32]. Переход между базисами является линейным преобразованием координат и может быть выполнен методом О. Б. Лупанова со сложностью $O(n^2/\log n)$, что автоматически ведет к понижению оценки сложности умножения в нормальных базисах. Оказывается, что переход выполняется особенно просто для тех же базисов, для которых хорошо работает алгоритм Месси—Омура. И с использованием арифметики стандартных базисов сложность умножения в них оказывается почти линейной (см. [2, 3]). Для гауссовых нормальных базисов применяется переход к стандартному базису в расширении поля (см., например, [3, 20]).

Так как в алгоритме инвертирования из § 3 на начальном и заключительном этапе выполняются линейные преобразования, размерность которых не изменяется при композиции с преобразованием координат из одного базиса в другой, то оценки теорем 3 и 4 справедливы для вычислений вообще в любом, а не только нормальном или стандартном, базисе. В оценку теоремы 11 следует внести поправку на выполнение перехода, тогда оценки сложности и глубины инвертирования и деления в нормальном базисе примут вид $O(n^2/\log n)$ и $O(\log n)$.

7.3. О возведении в произвольную степень. В современных работах [20, 23], посвященных экспоненцированию в конечных полях, оценки сложности возведения в произвольную степень в поле $GF(2^n)$ приводятся в виде $O(n^2 \log\log n)$ для стандартных базисов и нормальных базисов линейной сложности (для нормального базиса вообще справедлива та же самая оценка, потому что переход к полиномиальному базису и обратно всегда выполняется не сложнее, чем за $O(n^2/\log n)$ операций). Глубина всех перечисленных методов не меньше $O(\log^2 n)$. В работе [21] построена схема логарифмической глубины, однако сложность такой схемы слишком велика. Вопрос о реализации возведения в степень хотя бы со сложностью $O(n^2)$, по-видимому, остается открытым.

Автор благодарен своему научному руководителю С. Б. Гашкову за постановку задачи, многочисленные идеи, обсуждения и поддержку при ее решении.

СПИСОК ЛИТЕРАТУРЫ

1. Ахо А., Хопкрофт Д., Ульман Д. Построение и анализ вычислительных алгоритмов. — М.: Мир, 1979.
2. Болотов А. А., Гашков С. Б. О быстром умножении в нормальных базисах конечных полей // Дискретная математика. — 2001. — Т. 13, вып. 3. — С. 3–31.
3. Болотов А. А., Гашков С. Б., Фролов А. Б., Часовских А. А. Элементарное введение в эллиптическую криптографию: алгебраические и алгоритмические основы. — М.: КомКнига, 2006.
4. Гашков С. Б., Хохлов Р. А. О глубине логических схем для операций в полях $GF(2^n)$ // Чебышевский сборник. — 2003. — Т. 4, вып. 4(8). — С. 59–71.
5. Карацуба А. А., Офман Ю. П. Умножение многозначных чисел на автоматах // Докл. АН СССР. — 1961. — Т. 145(2). — С. 293–294.
6. Кнут Д. Искусство программирования. Т. 2. Получисленные алгоритмы. — М.: Вильямс, 2004.
7. Коблиц Н. Курс теории чисел и криптографии. — М.: ТВП, 2001.
8. Лидл Р., Нидеррайтер Х. Конечные поля. — М.: Мир, 1988.
9. Лупанов О. Б. Асимптотические оценки сложности управляющих систем. — М.: Изд-во МГУ, 1984.
10. Лупанов О. Б. О вентильных и контактно-вентильных схемах // Докл. АН СССР. — 1956. — Т. 111(6). — С. 1171–1174.
11. Сергеев И. С. Об инвертировании в конечных полях характеристики 2 с логарифмической глубиной // Вестник МГУ. Математика. Механика. — 2007. — № 1. — С. 26–31.
12. Серпинский В. 250 задач по элементарной теории чисел. — М.: Просвещение, 1968.
13. Тоом А. Л. О сложности схемы из функциональных элементов, реализующей умножение целых чисел // Докл. АН СССР. — 1963. — Т. 150(3). — С. 496–498.
14. Хохлов Р. А. Реализация логическими схемами операций умножения и инвертирования в конечных полях характеристики два. — Канд. дисс., МГУ, 2005.
15. Храпченко В. М. Об асимптотической оценке времени сложения параллельного сумматора // Проблемы кибернетики. Вып. 19. — М.: Наука, 1967. — С. 107–120.
16. Beame P., Cook S., Hoover H. Log depth circuits for division and related problems // SIAM J. Comput. — 1986. — V. 15, № 4. — P. 994–1003.
17. Bini D., Pan V. Polynomial and matrix computations. V. 1. — Boston: Birkhäuser, 1994.
18. Brent R., Kung H. Fast algorithms for manipulating formal power series // J. ACM. — 1978. — V. 25, № 4. — P. 581–595.
19. Eberly W. Very fast parallel polynomial arithmetic // SIAM J. Comput. — 1989. — V. 18, № 5. — P. 955–976.
20. Gao S., von zur Gathen J., Panario D., Shoup V. Algorithm for exponentiation in finite field // J. Symb. Comput. — 2000. — V. 29. — P. 879–889.
21. von zur Gathen J. Inversion in finite fields using logarithmic depth // J. Symb. Comput. — 1990. — V. 9. — P. 175–183.
22. von zur Gathen J., Gerhard J. Modern computer algebra. — Cambridge University Press, 1999.
23. von zur Gathen J., Nöcker M. Polynomial and normal bases for finite fields // J. Crypt. — 2005. — V. 18. — P. 337–355.
24. Grinchuk M. I., Bolotov A. A. Process for designing comparators and adders of small depth // US patent application. — 2006. — № 7020865.
25. Grove E. Proofs with potential. — Ph. D. thesis, U. C. Berkeley, 1993.

26. Hastad J., Leighton T. Division in $O(\log n)$ depth using $O(n^{1+\epsilon})$ processors. Неопубликованная работа. — 1986. — www.nada.kth.se/~yohanh/paraldivision.ps.
27. Huang X., Pan V. Fast rectangular matrix multiplication and applications // J. Complexity. — 1998. — V. 14. — P. 257–299.
28. Jungnickel D. Finite fields: structure and arithmetics. — Wissenschaftsverlag, Mannheim, 1995.
29. Litow B., Davida G. $O(\log n)$ parallel time finite field inversion // Proc. Aegean Workshop on Computing, Lecture Notes in Computer Science 319. — Berlin, 1988. — P. 74–80.
30. Paterson M., Pippenger N., Zwick U. Optimal carry save networks // LMS Lecture Notes Series. — V. 169. Boolean function Complexity. — Cambridge University Press, 1992. — P. 174–201.
31. Rosser J., Schoenfeld L. Approximate formulas for some functions of prime numbers // Ill. J. Math. — 1962. — V. 6 — P. 64–94.
32. Schönhage A. Schnelle multiplikation von polynomen über körpern der charakteristik 2 // Acta Inf. — 1977. — V. 7. — P. 395–398.
33. Strassen V. Gaussian elimination is not optimal. // Numer. Math. — 1969. — B. 13, № 4. — P. 354–356. [Русский перевод: Штрассен Ф. Алгоритм Гаусса не оптимален. // Кибернетич. сб. Новая серия. Вып. 7. — М.: Мир, 1970. — С. 67–70.]

Поступило в редакцию 09 XI 2005