Записки научных
семинаров ПОМИ
Том 373, 2009 г.

Г. И. Малашонок, Ю. Д. Валеев, А. О. Лапаев

О ВЫБОРЕ АЛГОРИТМА УМНОЖЕНИЯ ДЛЯ ПОЛИНОМОВ И ПОЛИНОМИАЛЬНЫХ МАТРИЦ

1. Введение

Традиционно сложность вычислительного алгоритма оценивается по числу мультипликативных операций, при этом обычно ограничиваются указанием степени мажорирующей функции [1–2]. Такие оценки полезны главным образом для понимания того, как быстро растет сложность алгоритма на бесконечности. Для сравнения алгоритмов, применяемых в реальных задачах, и для выбора конкретного алгоритма для конкретного класса задач нужен иной подход.

В работах [3–5] предлагается оценивать вычислительную сложность алгоритмов с разреженными данными с помощью анализа степени разреженности данных в течение всего вычислительного процесса и получении точных выражений для математического ожидания числа арифметических операций. При этом, в выражении, описывающем сложность алгоритма, учитываются и мультипликативные, и аддитивные операции.

В настоящей работе развивается этот подход для получения оценок сложности алгоритмов умножения полиномов и полиномиальных матриц для разных типов полиномиальных колец.

Предполагается, что коэффициенты полиномов лежат в области, которая может быть одного из двух видов. Это либо область $\mathbb{W}$, все элементы которой могут быть записаны в одном машинном слове, например, числа с плавающей точкой или конечные поля, либо это область $\mathbb{Z}$ – целые числа, для записи которых нужно несколько машинных слов.

Для полиномиальных матриц над $\mathbb{W}$ используется три параметра: максимальная степень полинома, коэффициент разреженности полинома и размер матрицы. Для полиномиальных матриц над $\mathbb{Z}$ доба-

Ключевые слова : алгоритмы умножения полиномов, алгоритмы умножения матриц, полиномиальные матрицы, алгоритм Карацубы, алгоритм Штрассена, модулярные алгоритмы, быстрое преобразование Фурье.

вляется четвертый параметр – количество машинных слов в записи коэффициентов полиномов. В выражениях для сложности алгоритмов умножения полиномов используется соответственно 2 и 3 параметра.

Мы исследуем стандартные алгоритмы, алгоритмы Карацубы, Штрассена, БПФ и модулярные варианты, использующие китайскую теорему об остатках.

Задача состоит в том, чтобы получить выражения для сложности операций умножения полиномов и полиномиальных матриц как математического ожидания числа арифметических операций для исследуемых алгоритмов, по полученным выражениям построить таблицы для набора параметров, представляющих практический интерес, и указать самые эффективные алгоритмы в зависимости от набора параметров.

Важной задачей является экспериментальная проверка полученных оценок сложности. Поэтому мы приводим результаты экспериментов, в которых замерялось реальное время вычисления произведения полиномов и полиномиальных матриц с использованием данных алгоритмов.

2. Алгоритмы умножения в области $\mathbb{W}[x]$

Будем рассматривать случайные полиномы из кольца полиномов $\mathbb{W}[x]$ и определим понятие “тип полинома”.

Определение 1. *Назовем полиномом типа* (m, α_i) *случайный полином* $\sum_{i=1}^{m} c_i x^{i-1}$ *из* $\mathbb{W}[x]$, *коэффициент* c_i *которого отличен от нуля с вероятностью* α_i $(1 \le i \le m)$.

Степень полинома типа (m, α_i), очевидно, не может превышать $m-1$. Обозначим $\mathcal{E}(m, \alpha_i)$ математическое ожидание числа ненулевых коэффициентов полинома типа (m, α_i):

$$\mathcal{E}(m, \alpha_i) = \sum_{i=1}^{m} \alpha_i.$$

Если $\alpha_i = \alpha$ для всех i, то будем записывать тип полинома в виде (m, α), а число α будем называть *плотностью полинома*.

Будем рассматривать случайные матрицы над полиномами из $\mathbb{W}[x]$, у которых все элементы имеют один и тот же полиномиальный тип, и определим тип матрицы.

Определение 2. *Назовем матрицей типа* (n, m, α_i) *случайную матрицу размера* $n \times n$ *над* $\mathbb{W}[x]$, *у которой каждый элемент – это полином типа* (m, α_i).

Обозначим $\mathcal{E}(n, m, \alpha_i)$ математическое ожидание общего количества ненулевых коэффициентов полиномов у всех элементов матрицы типа (n, m, α_i):

$$\mathcal{E}(n, m, \alpha_i) = n^2 \sum_{i=1}^{m} \alpha_i.$$

Разбиение множества полиномов и матриц на однотипные подмножества задает отношение эквивалентности. Тип результата операций сложения, вычитания и умножения однозначно определяется типом операндов. Поэтому можно говорить об операциях над типами, имея в виду тип результата операции, когда операнды имеют заданные типы.

Теорема 1. *Для области* $\mathbb{W}$ *характеристики* 0 *справедливы следующие равенства для типов:*

$$(m, \alpha) + (m, \beta) = (m, 1 - (1 - \alpha)(1 - \beta)), \tag{1}$$

$$(n, m, \alpha) + (n, m, \beta) = (n, m, 1 - (1 - \alpha)(1 - \beta)), \tag{2}$$

$$(m, \alpha) \times (m, \beta) = (2m - 1, \pi_i^m), \tag{3}$$

$$(n, m, \alpha) \times (n, m, \beta) = (n, 2m - 1, \rho_i^{m,n}). \tag{4}$$

Здесь и далее используются обозначения:

$$t_i^m = \begin{cases} i & \text{при } 1 \leqslant i \leqslant m, \\ 2m - i & \text{при } m \leqslant i \leqslant 2m - 1, \end{cases} \tag{5}$$

$$\pi_i^m = 1 - (1 - \alpha\beta)^{t_i^m} \quad \text{при } 1 \leqslant i \leq 2m - 1, \tag{6}$$

$$\rho_i^{m,n} = 1 - (1 - \pi_i^m)^n \quad \text{при } 1 \leq i \leqslant 2m - 1. \tag{7}$$

Доказательство. Для двух чисел, отличных от нуля с вероятностью α и β, вероятность того, что они одновременно нулевые, равна $(1 - \alpha)(1 - \beta)$. Отсюда следуют равенства (1) и (2).

Коэффициенты полинома степени $m-1$ мы нумеруем числами $1,2,..m$. Результат произведения двух таких полиномов – полином степени $2m-2$. У этого полинома каждый коэффициент с номером $i=1,2,..,2m-1$ образован суммой из t_i^m слагаемых, а каждое слагаемое равно нулю с вероятностью $1-\alpha\beta$. Поэтому вероятность того, что коэффициент с номером i нулевой, равна $(1-\alpha\beta)^{t_i^m}$. Отсюда следует (3).

Каждый элемент в произведении полиномиальных матриц образован суммой из n произведений полиномов. У каждого из n полиномов-слагаемых коэффициент с номером i равен нулю с вероятностью $1-\pi_i^m$, следовательно, вероятность того, что их сумма нулевая, равна $(1-\pi_i^m)^n$. Отсюда следует (4).

Заметим, что для поля характеристики $p\neq 0$ нужно было бы учитывать, что сумма двух случайных чисел может оказаться равной нулю с вероятностью $1/p$, в то время как для поля характеристики 0 мы считаем эту вероятность равной нулю.

Будем обозначать, соответственно, $\mathcal{EA}$ и $\mathcal{EM}$ математические ожидания числа аддитивных операций и числа операций умножения в заданном алгоритме.

Будем считать одной аддитивной операцией такую операцию, в которой хотя бы один из участвующих операндов ненулевой. Число аддитивных операций при сложении или вычитании двух разреженных структур – это число ненулевых коэффициентов полиномов в результате. Поэтому для сложности суммы или разности полиномов типа (1) получим выражение

$$\mathcal{EA}^{\Sigma}_{m,\alpha,\beta}=m(1-(1-\alpha)(1-\beta)), \tag{8}$$

а для суммы или разности матриц над полиномами типа (2) получим выражение

$$\mathcal{EA}^{\Sigma}_{n,m,\alpha,\beta}=n^2m(1-(1-\alpha)(1-\beta)). \tag{9}$$

Нахождение выражений для сложности алгоритмов умножения разреженных полиномов и матриц над ними – значительно более сложная задача. Будем для каждого алгоритма умножения получать выражения для сложности, отслеживая плотности полиномов на всех этапах алгоритма.

2.1. Стандартные алгоритмы умножения полиномов и матриц

Для стандартных алгоритмов умножения полиномов и матриц над Кольцом полиномов задача подсчета числа операций умножения и сложения легко решается. Будем подсчитывать только те операции умножения, у которых оба операнда ненулевые.

Для стандартного алгоритма умножения полиномов типа (m, α) и (m, β) получим

$$\mathcal{EA}^{PS}_{m,\alpha,\beta} = \mathcal{EM}^{PS}_{m,\alpha,\beta} = m^2\alpha\beta. \tag{10}$$

Стандартный алгоритм умножения полиномиальных матриц типа (n, m, α) и (n, m, β) предполагает вычисление n^2 скалярных произведений n-векторов полиномов типа (m, α) и (m, β). Поэтому математическое ожидание числа операций умножения и сложения равно:

$$\mathcal{EM}^{MS}_{n,m,\alpha,\beta} = n^3m^2\alpha\beta, \tag{11}$$

$$\mathcal{EA}^{MS}_{n,m,\alpha,\beta} = n^3m^2\alpha\beta + n^2\sum_{i=1}^{2m-1}\sum_{s=2}^{n}\rho_i^{m,s}. \tag{12}$$

В выражении (12) первое слагаемое отвечает за операции суммирования коэффициентов полиномов, которые появляются при вычислении произведения полиномов, а второе – за последовательное суммирование полученных произведений полиномов. При этом плотность суммы нарастает в соответствии с (7), и перед s-тым суммированием плотность суммы равна $\rho_i^{m,s}$, а математическое ожидание числа операций при сложении такого полинома с полиномом плотности π_i^m равно $1-(1-\rho_i^{m,s})(1-\pi_i^m) = \rho_i^{m,s+1}$ в соответствии с (1).

Введем функции $F_m(\lambda)$ и $G_m(\lambda)$ натурального аргумента m и действительного аргумента λ: $F_m(1) = 2m-1$, $G_m(1) = m-1$, а для всех $\lambda \neq 1$ определим

$$F_m(\lambda) = \sum_{i=1}^{2m-1}\lambda^{t_i^m} = \frac{\lambda^{m+1}+\lambda^m-2\lambda}{\lambda-1}, \tag{13}$$

$$G_m(\lambda) = \sum_{i=1}^{m-1}\lambda^i = \frac{\lambda^m-\lambda}{\lambda-1}. \tag{14}$$

Тогда выражение (12) можно упростить:

$$\mathcal{EA}^{MS}_{n,m,\alpha,\beta} = n^3m^2\alpha\beta + n^2(n-1)(2m-1) - sn^2\sum_{s=2}^{n} F_m((1-\alpha\beta)^s). \quad (15)$$

2.2. Алгоритм Карацубы для умножения полиномов

Пусть f и g – полиномы, имеющие типы (m,β) и (m,γ), $0<\beta,\gamma\le 1$, $m=2^M$, $M>1$. Алгоритм Карацубы для вычисления произведения этих полиномов состоит в рекурсивном вычислении по формуле

$$fg = ac + (ac+bd-(a-b)(c-d))x^l + bdx^{2l}, \quad (16)$$

где $l=m/2$, $f=a+bx^l$, $g=c+dx^l$, a,b – полиномы типа (l,β), c,d – типа (l,γ).

Нужно определить плотности $\sigma(j)$ полиномов, которые появляются при погружении в рекурсию, для этого введем обозначения для функций действительного аргумента $\alpha\in[0,1]$:

$$\begin{aligned} \mu_s &= (1-\alpha)^{2^s},\\ \sigma(s) &= 1-\mu_s,\\ \pi_i^l(r,s) &= 1-(1-\sigma(r)\sigma(s))^{t_i^l},\\ \rho_i^{l,h}(r,s) &= 1-(1-\pi_i^l(r,s))^h,\\ \nu_{r,s} &= \mu_r+\mu_s-\mu_r\mu_s \end{aligned} \quad (17)$$

для всех неотрицательных целых i,r,s и натуральных h,l. Функция t_i^l определена выражением (5). Тогда справедливы равенства

$$\sigma(0)=\alpha, \quad \pi_i^l(r,s)=1-(\nu_{r,s})^{t_i^l}, \quad \rho_i^{l,h}(r,s)=1-(\nu_{r,s})^{ht_i^l},$$

и имеет место

Следствие 1 теоремы 1. *Для всех неотрицательных целых i,r,s и натуральных h,l справедливы равенства для типов:*

$$(l,\sigma(r))+(l,\sigma(r))=(l,\sigma(r+1)), \quad (18)$$

$$(l,\sigma(r))\times(l,\sigma(s))=(2l-1,\pi_i^l(r,s)), \quad (19)$$

$$\sum_{j=1}^{h}(2l-1,\pi_i^l(r,s))=(2l-1,\rho_i^{l,h}(r,s)). \quad (20)$$

Здесь в левой части равенства (20) *стоит сумма* h *однотипных слагаемых, а в правой – тип результата.*

Подчеркнем, что функцией $\sigma(t)$ удобно описывать плотность полинома, так как при сложении двух полиномов плотности $\sigma(t)$ сумма имеет плотность $\sigma(t+1)$.

Рассмотрим алгоритм Карацубы (16) для случая $\beta = \sigma(r)$, $\gamma = \sigma(s)$. Здесь r и s могут быть любыми неотрицательными целыми.

Отметим, что для вычисления выражения (16) необходимо вычислить три произведения полиномов. При этом в *двух* случаях, при вычислении ac и bd, сомножители имеют типы $(l, \sigma(r))$ и $(l, \sigma(s))$, и в *одном* случае, при вычислении произведения $(a-b)(c-d)$, сомножители имеют типы $(l, \sigma(r+1))$ и $(l, \sigma(s+1))$. При погружении в рекурсию разброс типов будет расти. Полная глубина рекурсии M. Пусть на k-том рекурсивном шаге будет d_i^k произведений полиномов, у которых сомножители имеют типы $(2^{M-k}, \sigma(r+i))$ и $(2^{M-k}, \sigma(s+i))$, $i = 0, 1, \ldots, k$. Легко показать, что

$$(2+z)^k = \sum_{i=0}^{k} d_i^k z^i$$

является производящей функцией последовательности d_i^k. Следовательно,

$$d_i^k = \binom{k}{i} 2^{k-i}.$$

Поэтому имеет место следующее

Предложение 1. *При вычислении произведения полиномов типов* $(m, \sigma(r))$ *и* $(m, \sigma(s))$, $r, s \geq 0$, $m = 2^M$, *с помощью алгоритма Карацубы на* k*-том рекурсивном шаге* $(k = 1, \ldots, M)$ *выполняется* 3^k *умножений полиномов, из них в* $\binom{k}{i} 2^{k-i}$ *случаях сомножители будут иметь типы* $(2^{M-k}, \sigma(r+i))$ *и* $(2^{M-k}, \sigma(s+i))$, $i = 0, 1, \ldots, k$.

Отметим, что коэффициенты полиномов умножаются только на последнем шаге рекурсивного погружения, а на всех промежуточных шагах происходит только суммирование. Отсюда следует

Предложение 2. *При вычислении произведения полиномов типов* $(m, \sigma(r))$ *и* $(m, \sigma(s))$, $r, s \geq 0$, $m = 2^M$, *с помощью алгоритма Карацубы математическое ожидание числа операций умножения коэффициентов полиномов будет равно:*

$$\mathcal{EM}^{PK}_{m,r,s} = \sum_{i=0}^{M} \binom{M}{i} 2^{M-i} \sigma(r+i)\sigma(s+i). \tag{21}$$

В частности, для произведения плотных полиномов типа $(m,1)$, когда $\alpha = 1$, $r = s = 0$, из (21) следует

$$\mathcal{EM}^{PK}_{m} = m^{\log_2 3}.$$

Перейдем к подсчету числа сложений. Рассмотрим один шаг алгоритма (16) и каждое из сложений, которые здесь выполняются.

1. Полиномы $a-b$ и $c-d$ имеют типы $(l, \sigma(r+1))$ и $(l, \sigma(s+1))$. Математическое ожидание числа аддитивных операций при вычислении каждого из них как разности двух полиномов будет $l\sigma(r+1)$ и $l\sigma(s+1)$, соответственно. Следовательно, математическое ожидание общего числа аддитивных операций будет равно

$$\mathcal{EA}_1 = l(2 - \mu_{r+1} - \mu_{s+1}).$$

2. Полиномы ac и bd имеют имеют одинаковый тип $(2l-1, \pi^l_i(r,s))$. Их сумма имеет тип $(2l-1, \rho^{l,2}_i(r,s))$. Так как $\rho^{l,2}_i(r,s) = 1 - \nu^{2t^l_i}_{r,s}$, то математическое ожидание числа аддитивных операций при вычислении суммы полиномов $ac+bd$ будет равно

$$\mathcal{EA}_2 = \sum_{i=1}^{2l-1} \rho^{l,2}_i(r,s) = 2l - 1 - F_l(\nu^2_{r,s}).$$

3. Полином $(a-b)(c-d)$ имеет тип $(2l-1, \pi^l_i(r+1, s+1))$. Следовательно, полином $(ac+bd)-(a-b)(c-d)$ имеет тип $(2l-1, \psi_i)$, где $\psi_i = 1 - (1-\rho^{l,2}_i(r,s))(1-\pi^l_i(r+1,s+1)) = 1 - (\nu^2_{r,s}\nu_{r+1,s+1})^{t^l_i}$. Следовательно, математическое ожидание числа аддитивных операций при вычислении этой разности полиномов будет равно

$$\mathcal{EA}_3 = \sum_{i=1}^{2l-1} \psi_i = 2l - 1 - F_l(\nu^2_{r,s}\nu_{r+1,s+1}).$$

4. Последнее сложение в правой части выражения (16) – это сложение среднего полинома $(ac+bd-(a-b)(c-d))x^l$ с полиномами

ac и bdx^{2l}. При этом его младшие $l-1$ коэффициенты складываются со старшими коэффициентами полинома ac, а старшие $l-1$ коэффициенты складываются с младшими коэффициентами полинома bdx^{2l}. Для каждого из этих двух полиномиальных сложений математическое ожидание числа сложений коэффициентов будет равно

$$\mathcal{EA}_4 = \sum_{i=1}^{l-1}(1-(1-\psi_i)(1-\pi^l_{l-i}(r,s)) = l-1-(\nu_{r,s})^l G_l(\nu_{r,s}\nu_{r+1,s+1}).$$

5. Суммируя количество аддитивных операций на одном шаге рекурсивного алгоритма (пп. 1–4), получим:

$$\begin{aligned} CP(l,r,s) &= 8l-4-l(\mu_{r+1}+\mu_{s+1}) - F_l(\nu^2_{r,s}) \\ &\quad - F_l(\nu^2_{r,s}\nu_{r+1,s+1}) - 2(\nu_{r,s})^l G_l(\nu_{r,s}\nu_{r+1,s+1}). \end{aligned}$$

Предложение 1 дает описание распределения по типам полиномов, участвующих в вычислениях на k-том рекурсивном шаге. Суммирование по всем типам для одного k-того шага, а затем по всем шагам $(k=0,1,..,M-1)$ позволяет найти искомую сложность алгоритма Карацубы по аддитивным операциям.

Предложение 3. *При вычислении произведения полиномов типа* $(m,\sigma(r))$ *и* $(m,\sigma(s))$, $r,s\geq 0$, $m=2^M$, *с помощью алгоритма Карацубы математическое ожидание числа аддитивных операций будет равно*

$$\mathcal{EA}^{PK}_{m,r,s} = \sum_{k=0}^{M-1}\sum_{i=0}^{k}\binom{k}{i}2^{k-i}CP(2^{M-k-1}, r+i, s+i). \qquad (22)$$

В частности, для произведения плотных полиномов: $\alpha=1$, $r=s=0$,

$$\mathcal{EA}^{PK}_{m} = 6m^{\log_2 3} - 8m + 2.$$

Найдем математическое ожидание отношения числа всех операций в стандартном алгоритме умножения полиномов типа (m,α) и в алгоритме Карацубы: $\frac{\mathcal{EA}^{PS}_{m,\alpha,\alpha}+\mathcal{EM}^{PS}_{m,\alpha,\alpha}}{\mathcal{EA}^{PK}_{m,0,0}+\mathcal{EM}^{PK}_{m,0,0}}$. Результаты представлены

Таблица 1. Отношение числа операций сложения и умножения для стандартного алгоритма и для алгоритма Карацубы при умножения полиномов типа (m, α)

$m\backslash\%$	10	20	30	40	50	60	70	80	90	100
64	0.05	0.12	0.21	0.34	0.49	0.68	0.90	1.16	1.45	1.78
128	0.05	0.14	0.26	0.42	0.62	0.86	1.15	1.48	1.86	2.29
256	0.06	0.16	0.32	0.53	0.79	1.11	1.49	1.93	2.43	2.99
512	0.07	0.20	0.40	0.67	1.02	1.44	1.94	2.52	3.18	3.92
2^{10}	0.08	0.25	0.51	0.87	1.33	1.89	2.56	3.33	4.20	5.18
2^{11}	0.10	0.32	0.66	1.14	1.75	2.50	3.38	4.40	5.56	6.86
2^{12}	0.12	0.41	0.87	1.50	2.31	3.30	4.48	5.83	7.38	9.10
2^{13}	0.15	0.53	1.14	1.98	3.06	4.38	5.95	7.75	9.80	12.1
2^{14}	0.19	0.69	1.50	2.62	4.06	5.82	7.91	10.3	13.0	16.1
2^{15}	0.25	0.90	1.98	3.47	5.39	7.74	10.5	13.7	17.4	21.4

в Табл.1. В первой строке указана плотность полинома α в процентах, а в первом столбце – значение m.

Как показали эксперименты, в которых сравнивалось время вычисления произведения полиномов с помощью стандартного алгоритма и алгоритма Карацубы, среднее отклонение от таблицы 1 составляет 61%.

Таблица 1 примерно делится на две части своей побочной диагональю: в верхней части лучшим является стандартный алгоритм (0), в нижней – алгоритм Карацубы (1). По результатам экспериментов эта диагональ оказалась сильнее смещена вниз.

2.3. Алгоритм Штрассена для умножения матриц

Произведение матриц $A = (a_{ij})$ и $B = (b_{ij})$ второго порядка можно найти с помощью 7 операций умножения и 18 операций сложения:

$$AB = \begin{pmatrix} t_1 + t_4 - t_5 + t_7 & t_3 + t_5 \\ t_2 + t_4 & t_1 + t_3 - t_2 + t_6, \end{pmatrix}. \tag{23}$$

$t_1 = (a_{11} + a_{22})(b_{11} + b_{22})$, $t_5 = (a_{11} + a_{12})b_{22}$, $t_2 = (a_{21} + a_{22})b_{11}$, $t_6 = (a_{21} - a_{11})(b_{11} + b_{12})$. $t_3 = a_{11}(b_{12} - b_{22})$, $t_7 = (a_{12} - a_{22})(b_{21} + b_{22})$, $t_4 = a_{22}(b_{21} - b_{11})$.

Алгоритм, получаемый применением такой схемы вычислений для блочного рекурсивного умножения матриц, известен как алгоритм Штрассена [1]. Если умножаются матрицы порядка $n = 2^N$, то всего потребуется $n^{\log_2 7}$ операций умножения.

Пусть матрицы A и B имеют тип (n, m, α), $n = 2^N$, $m = 2^M$. Тогда равенства (23) будем рассматривать как блочные с блоками $a_{i,j}$ и $b_{i,j}$ размера $n/2 \times n/2$. Из 7-ми произведений в (23) в 4-х случаях один из сомножителей имеет тип $(n/2, m, \alpha)$, а другой – $(n/2, m, \sigma(1))$, и в 3-х случаях оба сомножителя имеют тип $(n/2, m, \sigma(1))$. Следовательно, на k-том рекурсивном шаге в 7^k блочных умножениях будут участвовать блоки типа $(n/2^k, m, \sigma(j))$, $j = 0, 1, 2, \ldots, k$.

Обозначим $a^k_{i,j}$ число произведений, в которых на k-том шаге рекурсии участвуют блоки размера 2^{N-k} с левым сомножителем типа $(2^{N-k}, m, \sigma(i))$ и правым сомножителем типа $(2^{N-k}, m, \sigma(j))$. В соответствии с (23) при погружении в рекурсию в *двух* случаях, при вычислении t_2, t_5, увеличена плотность первого сомножителя, в *двух* случаях, при вычислении t_3, t_4, увеличена плотность второго сомножителя и в *трех* случаях, при вычислении t_1, t_6, t_7, увеличены плотности обоих сомножителей. Поэтому производящая функция последовательности $a^k_{i,j}$ имеет вид:

$$(2y + 2z + 3yz)^k = \sum_{i=0}^{k} \sum_{j=0}^{k} a^k_{i,j} y^i z^j.$$

Следовательно,

$$a^k_{i,j} = \binom{k}{i} \binom{i}{i+j-k} 2^{2k-i-j} 3^{i+j-k}, \quad i, j \le k, \quad i + j \ge k.$$

Предложение 4. *При вычислении произведения матриц типа* $(2^N, m, \alpha)$ *с помощью алгоритма Штрассена на* k*-том рекурсивном шаге,* $k = 1, \ldots, N$*, выполняется* 7^k *умножений блоков, из них в* $\binom{k}{i}\binom{i}{i+j-k}2^{2k-i-j}3^{i+j-k}$ *случаях типы первого и второго сомножителей будут, соответственно, равны* $(2^{N-k}, m, \sigma(i))$ *и* $(2^{N-k}, m, \sigma(j))$*,* $i, j \le k$*,* $i + j \ge k$*.*

Произведения элементов матриц вычисляются на N-том рекурсивном шаге. Отсюда следует

Предложение 5. *При вычислении произведения матриц типа* $(2^N, m, \alpha)$ *с помощью алгоритма Штрассена математическое ожидание числа операций умножения будет равно*

$$\mathcal{EM}^{MSt}_{n,m,\alpha} = \sum_{i=0}^{N} \sum_{j=N-i}^{N} \binom{N}{i} \binom{i}{i+j-N} 2^{2N-i-j} 3^{i+j-N} \mathcal{EM}^{P}_{m,i,j}, \quad (24)$$

где $\mathcal{EM}^P_{m,i,j}$ *математическое ожидание числа умножений при вычислении произведения полиномов типа* $(m,\sigma(i))$ *и* $(m,\sigma(j))$ *(см.* (10) *и* (21)*).*

В частности, при $\alpha = 1$ получим:

$$\mathcal{EM}^{MSt}_{n,m} = n^{\log_2 7}\mathcal{EM}^P_m.$$

Перейдем к подсчету числа аддитивных операций. Рассмотрим один шаг алгоритма (23) и каждое из сложений, которое здесь выполняется. Пусть матрица A имеет тип $(2l, m, \sigma(r))$, а матрица B имеет тип $(2l, m, \sigma(s))$.

1. При вычислении $t_1, \ldots, t_7$ как суммы блоков матриц A и B выполняется 5 сложений матриц типа $(l, m, \sigma(r))$ и столько же – типа $(l, m, \sigma(s))$. При этом математическое ожидание числа всех аддитивных операций будет равно

$$\mathcal{EA}_1 = 5l^2m(\sigma(r+1) + \sigma(s+1)).$$

Полученные матрицы t_1, t_6, t_7 имеют тип $(l, 2m-1, \rho_i^{m,l}(r+1, s+1))$, матрицы t_2 и t_5 имеют тип $(l, 2m-1, \rho_i^{m,l}(r+1, s))$ и матрицы t_3, t_4 имеют тип $(l, 2m-1, \rho_i^{m,l}(r, s+1))$.

2. При вычислении каждой из сумм $t_2 + t_4$ и $t_3 + t_5$ и разностей $t_4 - t_5$ и $t_3 - t_2$ будет получена матрица типа $(l, 2m-1, \phi_i)$, $\phi_i = 1 - (1 - \rho_i^{m,l}(r, s+1))(1 - \rho_i^{m,l}(r+1, s))$.

При вычислении каждой из сумм $t_1 + t_7$ и $t_1 + t_6$ будет получена матрица типа $(l, 2m-1, \chi_i)$, $\chi_i = 1 - (1 - \rho_i^{m,l}(r+1, s+1))^2$.

При вычислении каждой из сумм $t_1 + t_4 - t_5 + t_7$ и $t_1 + t_3 - t_2 + t_6$ будет получена матрица типа $(l, 2m-1, \eta_i)$, $\eta_i = 1 - (1 - \chi_i)(1 - \phi_i)$.

Математическое ожидание числа всех аддитивных операций будет равно:

$$\begin{aligned}\mathcal{EA}_2 &= l^2 \sum_{i=1}^{2m-1} (4\phi_i + 2\chi_i + 2\eta_i) \\ &= l^2(8(2m-1) - 4F_m(\phi^l_{r,s}) - 2F_m(\chi^l_{r,s}) - 2F_m(\eta^l_{r,s})),\end{aligned}$$

где

$$\phi^l_{r,s} = (\mu_r + \mu_{s+1} - \mu_r\mu_{s+1})^l(\mu_{r+1} + \mu_s - \mu_{r+1}\mu_s)^l,$$
$$\chi^l_{r,s} = (\mu_{r+1} + \mu_{s+1} - \mu_{r+1}\mu_{s+1})^{2l}, \quad \eta^l_{r,s} = \phi^l_{r,s}\chi^l_{r,s}$$

и имеют место равенства:

$\phi_i = 1-(\phi^l_{r,s})^{t^m_i}, \chi_i = 1-(\chi^l_{r,s})^{t^m_i}, \eta_i = 1-(\eta^l_{r,s})^{t^m_i} \quad (i=1,\ldots,2m-1).$

3. Суммируя результаты пп. 1 и 2, найдем математическое ожидание числа аддитивных операций на одном рекурсивном шаге при вычислении произведения матриц типа $(2l,m,\sigma(r))$ и $(2l,m,\sigma(s))$ с помощью алгоритма Штрассена:

$$C^{m,l}_{r,s} = l^2(26m-8-5m(\mu_{r+1}+\mu_{s+1})-4F_m(\phi^l_{r,s})-2F_m(\chi^l_{r,s})-2F_m(\eta^l_{r,s})).$$

Предложение 6. *При вычислении произведения матриц типа* (n,m,α), $n=2^N$, *с помощью алгоритма Штрассена математическое ожидание общего числа аддитивных операций будет равно*

$$\mathcal{EA}^{MSt}_{n,m,\alpha} = \sum_{k=0}^{N-1}\sum_{i=0}^{k}\sum_{j=k-i}^{k}\binom{k}{i}\binom{i}{i+j-k}2^{2k-i-j}3^{i+j-k}C^{m,2^{N-1-k}}_{i,j} + \sum\nolimits_{i=0}^{N}\sum_{j=N-i}^{N}\binom{N}{i}\binom{i}{i+j-N}2^{2N-i-j}3^{i+j-N}\mathcal{EA}^{P}_{m,i,j}, \tag{25}$$

где $\mathcal{EA}^{P}_{m,i,j}$ *– математическое ожидание числа сложений при вычислении произведения полиномов типа* $(m,\sigma(i))$ *и* $(m,\sigma(j))$.

В частности, при $\alpha=1$ получим:

$$\mathcal{EA}^{MSt}_{n,m} = (1/3)(26m-8)(n^{\log_2 7}-n^2)+n^{\log_2 7}\mathcal{EA}^{P}_{m}.$$

2.4. Два модулярных алгоритма умножения матриц

Рассмотрим произведение матриц типа (n,m,α), в результате которого будет получена матрица типа $(n,2m-1,\rho^{m,n}_i)$, (см. (4)). Выберем $2m-1$ различных полиномов первой степени $m_i(x)=x-h_i\in\mathbb{W}[x]$. Для каждого полинома $m_i(x)$ факторкольцо $\mathbb{W}[x]/m_i(x)\mathbb{W}[x]$ обозначим через $\mathbb{W}_i[x]$. Чтобы найти произведение матриц A и B над $\mathbb{W}[x]$, будем для каждого i, $i=1,..,2m-1$, находить образы матриц A и B при отображении $\mathbb{W}[x]\to\mathbb{W}_i[x]$: $A\to A_i$, $B\to B_i$, вычислять их произведение A_iB_i, а затем восстанавливать произведение матриц AB по его образам в $\mathbb{W}_i[x]$ по китайской теореме об остатках (КТО).

Для вычисления образа полинома по одному модулю нужно m умножений и $m\alpha$ сложений для каждого из $2n^2$ полиномов в двух исходных матрицах.

В случае применения стандартного алгоритма нужно по n^3 операций умножения и сложения для умножения матриц по каждому модулю.

В случае применения алгоритма Штрассена нужно $n^{\log_2 7}$ операций умножения и $6(n^{\log_2 7} - n^2)$ операций сложения для каждого модуля.

Для восстановления одного полинома степени $2m-1$ при использовании схемы Ньютона с предварительно вычисленными всеми необходимыми коэффициентами требуется по $(2m-1)^2$ операций умножения и сложения. Всего нужно восстановить n^2 полиномов. Найдем общее число операций.

Предложение 7. *В модулярном алгоритме умножения матриц, когда применяется стандартный алгоритм умножения матриц по каждому модулю, математические ожидания числа аддитивных операций и операций умножения равны*

$$\mathcal{A}^{crt}_{n,m} = n^2(2m-1)(n+2m(1+\alpha)-1), \quad \mathcal{M}^{crt}_{n,m} = n^2(2m-1)(n+4m-1).$$

Если применяется алгоритм Штрассена для умножения матриц по каждому модулю, то математические ожидания числа аддитивных операций и операций умножения, соответственно, равны

$$\mathcal{A}^{crtSt}_{n,m} = (2m-1)(6n^{\log_2 7} + 2mn^2(1+\alpha) - 7n^2),$$
$$\mathcal{M}^{crtSt}_{n,m} = (2m-1)(n^{\log_2 7} + 4mn^2 - n^2).$$

2.5. Сравнение алгоритмов умножения матриц над $\mathbb{W}[x]$

Будем предполагать, что время выполнения аддитивных операций и операций умножения одинаковое, а все остальные операции не будем учитывать. Приведем результаты сравнения следующих шести алгоритмов. (0). Стандартный алгоритм умножения матриц со стандартным алгоритмом умножения полиномов. (1). Стандартный алгоритм умножения матриц с алгоритмом Карацубы для умножения полиномов. (2). Алгоритм Штрассена умножения матриц со стандартным алгоритмом умножения полиномов. (3). Алгоритм Штрассена умножения матриц с алгоритмом Карацубы для умножения полиномов. (4). Модулярный алгоритм со стандартным алгоритмом умножения

матриц для каждого модуля. (5). Модулярный алгоритм, в котором применяется алгоритм Штрассена умножения матриц для каждого модуля.

В приведенных ниже таблицах для заданных значений n, m, α указан номер алгоритма (k), которой имеет наименьшую сложность, и отношение количества арифметических операций в стандартном алгоритме (0) по отношению к алгоритму (k). Плотность α приведена в процентах.

Как видно из этих таблиц, для сильно разреженных полиномов стандартный алгоритм (0) всегда является лучшим. С другой стороны, при росте размера матрицы n лучшим алгоритмом является модулярный алгоритм с умножением матриц по алгоритму Штрассена для каждого модуля (5). Промежуточное положение занимают алгоритмы (3) и (4). Отметим, что алгоритмы (1) и (2) проигрывают во всех случаях.

Таблица 2. Номера алгоритмов, имеющих наименьшую сложность при умножении матриц типа (n, m, α), и их выигрыш по отношению к стандартному алгоритму

	$n=4$				$n=16$				$n=64$			
$m\backslash\alpha$	1	10	50	100	1	10	50	100	1	10	50	100
4	0	0	0	0	0	0	0	4 1.4	0	0	0	4 2.3
16	0	0	0	3 1.3	0	0	0	4 1.8	0	0	4 1.3	4 4.4
64	0	0	0	3 2.2	0	0	0	3 2.8	0	0	4 1.7	4 6.6
256	0	0	0	3 3.8	0	0	3 1.2	3 4.9	0	0	4 1.9	4 7.6
1024	0	0	3 1.7	3 6.7	0	0	3 2.2	3 8.7	0	0	3 2.8	3 11
4096	0	0	3 3.	3 12	0	0	3 3.9	3 15	0	0	3 5.	3 20
16384	0	0	3 5.3	3 21	0	0	3 6.8	3 27	0	0	3 8.9	3 36
65536	0	0	3 9.3	3 37	0	0	3 12	3 49	0	0	3 16	3 64

	$n=256$				$n=1024$				$n=4096$			
$m\backslash\alpha$	1	10	50	100	1	10	50	100	1	10	50	100
4	0	0	4 1.	4 2.6	0	0	5 1.1	5 3.	0	0	5 1.5	5 3.9
16	0	0	4 2.1	4 7.	0	0	5 2.6	5 8.9	0	0	5 3.6	5 12
64	0	0	4 4.3	4 16	0	0	5 7.3	5 28	0	5 1.1	5 11	5 43
256	0	0	4 6.5	4 26	0	0	5 17	5 67	0	5 1.9	5 34	5 135
1024	0	0	4 7.6	4 30	0	5 1.1	5 26	5 104	0	5 3.3	5 76	5 301
4096	0	0	4 7.9	4 32	0	5 1.2	5 30	5 121	0	5 4.5	5 109	5 436
16384	0	0	3 12	3 47	0	5 1.3	5 32	5 126	0	5 4.9	5 123	5 490
65536	0	0	3 21	3 83	0	5 1.3	5 32	5 128	0	5 5.1	5 127	5 506

3. Алгоритмы умножения в области $\mathbb{Z}[x]$

Пусть P – число двоичных цифр, которые хранятся в одном машинном слове и $p = 2^P$. Будем говорить, что целое число *имеет тип* (w), если для его записи нужно w машинных слов.

Примем следующую *модель вычислителя*. Полагаем, что всегда выполняются следующие равенства для типов чисел:

$$(w) + (v) = (\max(v, w)), \quad (w) \times (v) = (w + v).$$

Считаем, что для алгоритма суммирования типа $(w) + (w)$ число сложений равно w, а для стандартного алгоритма умножения $(w) \times (v)$ количество умножений и количество сложений равны vw.

Рассмотрим умножение целых чисел по алгоритму Карацубы.

Пусть f и g – числа типа (w), $w = 2l$. Алгоритм Карацубы для вычисления произведения этих чисел состоит в рекурсивном вычислении по формуле

$$fg = ac + (ac + bd - (a - b)(c - d))p^l + bdp^{2l}, \tag{26}$$

где $f = a + bp^l$, $g = c + dp^l$, a, b, c, d – числа типа (l).

Так же, как при доказательстве предложения 3, заметим, что при вычислении по формуле (26) необходимо выполнить три умножения и следующие операции сложения или вычитания: *два* вычитания $h = a - b$ и $s = c - d$ над числами типа (l) и *четыре* операции над числами типа $2l$. Это следующие операции: $q = ac + bd$, $t = q - hs$, сложение с ac и сложение с младшей частью bdp^{2l}. Рассматривая случай, когда $w = 2^W$, получим следующие выражения. Число операций умножения равно

$$N_w^{\mathcal{M}K} = w^{\log_2 3}. \tag{27}$$

Число операций сложения равно

$$N_w^{\mathcal{A}K} = \sum_{i=0}^{W-1} 3^i (2 \cdot 2^{W-1-i} + 4 \cdot 2^{W-i}) = 10(w^{\log_2 3} - w). \tag{28}$$

Для стандартного алгоритма умножения полиномов число операций сложения и умножения обозначим, соответственно, $N_w^{\mathcal{A}S}$ и $N_w^{\mathcal{M}S}$:

$$N_w^{\mathcal{A}S} = N_w^{\mathcal{M}S} = w^2. \tag{29}$$

Отношение общего числа операций сложения и умножения в алгоритме Карацубы к общему числу операций в стандартном алгоритме $\frac{N_w^{\mathcal{A}K} + N_w^{\mathcal{M}K}}{N_w^{\mathcal{A}S} + N_w^{\mathcal{M}S}}$ показывает, что алгоритм Карацубы начинает выигрывать при $w > 47$, а выигрыш в 2 раза начинается при $w > 298$.

Так же, как в $\mathbb{W}[x]$, введем в области $\mathbb{Z}[x]$ типы полиномов и матриц.

Определение 3. *Назовем полиномом типа* (m, α_i, w) *случайный полином* $\sum_{i=1}^{m} c_i x^{i-1}$ *из* $\mathbb{Z}[x]$, *коэффициент* c_i *которого отличен от нуля с вероятностью* α_i $(1 \le i \le m)$ *и все ненулевые коэффициенты имеют тип* (w).

Определение 4. *Назовем матрицей типа* (n, m, α_i, w) *случайную матрицу размера* $n \times n$ *над* $\mathbb{Z}[x]$, *у которой каждый элемент – это полином типа* (m, α_i, w).

Имеют место свойства операций над типами, аналогичные (18)–(20):

$$(l, \sigma(r), w) + (l, \sigma(r), w) = (l, \sigma(r+1), w),$$
$$(l, \sigma(r), w) \times (l, \sigma(s), w) = (2l - 1, \pi_i^l(r, s), 2w).$$

3.1. Алгоритмы умножения полиномов

Будем обозначать $\mathcal{EA}^{P}_{m,i,j,w}$ и $\mathcal{EM}^{P}_{m,i,j,w}$ – математическое ожидание числа сложений и умножений при вычислении произведения полиномов типа $(m,\sigma(i),w)$ и $(m,\sigma(j),w)$.

Так же, как в пп. 2.1 и 2.2, получим

Предложение 8. *Для стандартного алгоритма умножения полиномов, имеющих типы* $(m,\sigma(r),w)$ *и* $(m,\sigma(s),w)$, $r,s\geq 0$, *получим:*

$$\mathcal{EM}^{PS}_{m,r,s,w}=m^2\sigma(r)\sigma(s)N^{\mathcal{M}}_{w}, \tag{30}$$

$$\mathcal{EA}^{PS}_{m,r,s,w}=m^2\sigma(r)\sigma(s)(N^{\mathcal{A}}_{w}+2w). \tag{31}$$

В выражении (31) первое слагаемое отвечает за операции сложения, которые выполняются при умножении коэффициентов полиномов, а второе слагаемое отвечает за операции сложения при суммировании полученных произведений коэффициентов, которые имеют тип $(2w)$. Из (21), (22) и предшествующих рассуждений следует

Предложение 9. *При вычислении произведения полиномов типов* $(m,\sigma(r),w)$ *и* $(m,\sigma(s),w)$, $r,s\geq 0$, $m=2^M$, *с помощью алгоритма Карацубы математические ожидания числа операций сложения и числа операций умножения будут равны:*

$$\mathcal{EM}^{PK}_{m,r,s,w}=N^{\mathcal{M}}_{w}\mathcal{EM}^{PK}_{m,r,s}, \tag{32}$$

$$\mathcal{EA}^{PK}_{m,r,s,w}=N^{\mathcal{A}}_{w}\mathcal{EM}^{PK}_{m,r,s}+\sum_{k=0}^{M-1}\sum_{i=0}^{k}\binom{k}{i}2^{k-i}CP^{r+i,s+i}_{2^{M-k-1},w}, \tag{33}$$

где

$$\begin{aligned}CP^{r,s}_{l,w}=w(&14l-8-l(\mu_{r+1}+\mu_{s+1})-2F_l(\nu^2_{r,s})\\&-2F_l(\nu^2_{r,s}\nu_{r+1,s+1})-4(\nu_{r,s})^lG_l(\nu_{r,s}\nu_{r+1,s+1})),\end{aligned} \tag{34}$$

и $\mathcal{EM}^{PK}_{m,r,s}$ *определено в* (21).

В частности, для произведения плотных полиномов типа $(m,1,w)$, $\alpha=1$, $r=s=0$, когда алгоритм Карацубы применяется и для полиномов, и для чисел, получим

$$\mathcal{EM}^{PK}_{m,w}=(mw)^{\log_2 3}.$$

В следующей таблице приведены результаты сравнения четырех алгоритмов умножения полиномов из $\mathbb{Z}[x]$: (0). Стандартный алгоритм умножения чисел и полиномов. (1). Алгоритм Карацубы для умножения чисел и стандартный алгоритм умножения полиномов. (2). Стандартный алгоритм умножения чисел и алгоритм Карацубы для умножения полиномов. (3). Алгоритм Карацубы для умножения и чисел, и полиномов.

Алгоритмы сравнивались по общему числу операций умножения и сложения. Плотность полиномов α приведена в процентах. Для каждого типа полинома (m, α, w) в таблице указан номер лучшего алгоритма и выигрыш этого алгоритма по отношению к алгоритму (0).

Таблица 3. Лучшие алгоритмы для умножения полиномов типа (m, α, w) из $\mathbb{Z}[x]$ по общему числу арифметических операций и их выигрыш по отношению к стандартному алгоритму

m	$w = 4$			$w = 16$			$w = 64$			$w = 256$		
%	10	50	100	10	50	100	10	50	100	10	50	100
4	0	0	2 1.5	0	0	2 1.7	1 1.1	1 1.1	3 1.9	1 1.9	1 1.9	3 3.3
16	0	0	2 2.1	0	2 1.	2 2.7	1 1.1	3 1.3	3 3.4	1 1.9	3 2.2	3 5.8
64	0	0	2 3.3	0	2 4	2 4.7	1 1.1	3 1.9	3 5.9	1 1.9	3 3.3	3 10
256	0	2 1.6	2 5.7	0	2 2.3	2 8.2	1 1.1	3 3.	3 10	1 1.9	3 5.2	3 18
2^{10}	0	2 2.6	2 10	0	2 3.9	2 14	1 1.1	3 5.	3 18	1 1.9	3 8.8	3 32
2^{12}	0	2 4.5	2 18	0	2 6.6	2 26	1 1.1	3 8.5	3 33	1 1.9	3 15	3 58
2^{14}	0	2 7.9	2 31	0	2 12	2 46	1 1.1	3 15	3 58	1 1.9	3 26	3 102
2^{16}	0	2 14	2 56	0	2 20	2 81	3 1.3	3 26	3 104	3 2.3	3 46	3 182

Были проведены эксперименты с соответствующими четырьмя программами. В среднем различие между экспериментом и таблицей 3 составляет 18%.

3.2. Стандартный алгоритм и алгоритм Штрассена для умножения полиномиальных матриц

Предложение 10. *Для стандартного алгоритма умножения матриц над полиномами типа (n, m, α, w) математические ожидания числа операций умножения и числа аддитивных операций, соответственно, равны:*

$$\mathcal{EM}^{MS}_{n,m,\alpha,w} = n^3 \mathcal{EM}^{P}_{m,0,0,w}, \tag{35}$$

$$\begin{aligned}\mathcal{EA}^{MS}_{n,m,\alpha,w} &= n^3 \mathcal{EA}^{P}_{m,0,0,w} \\ &+2wn^2(n-1)(2m-1) - 2wn^2 \sum_{s=2}^{n} F_m((1-\alpha^2)^s).\end{aligned} \tag{36}$$

Предложения 5 и 6, полученные для математического ожидания числа операций в алгоритме Штрассена для $\mathbb{W}[x]$, позволяют сформулировать аналогичные предложения для $\mathbb{Z}[x]$. Необходимо только учесть сложность оперций умножения и сложения коэффициентов.

Предложение 11. *При вычислении произведения матриц типа $(2^N, m, \alpha, w)$ с помощью алгоритма Штрассена получим:*

$$\mathcal{EM}^{MSt}_{n,m,\alpha,w} = \sum_{i=0}^{N} \sum_{j=N-i}^{N} \binom{N}{i} \binom{i}{i+j-N} 2^{2N-i-j} 3^{i+j-N} \mathcal{EM}^{P}_{m,i,j,w}, \tag{37}$$

$$\begin{aligned}\mathcal{EA}^{MSt}_{n,m,\alpha,w} &= \sum_{k=0}^{N-1} \sum_{i=0}^{k} \sum_{j=k-i}^{k} \binom{k}{i} \binom{i}{i+j-k} 2^{2k-i-j} 3^{i+j-k} C^{m,2^{N-1-k}}_{i,j,w} \\ &+ \sum_{i=0}^{N} \sum_{j=N-i}^{N} \binom{N}{i} \binom{i}{i+j-N} 2^{2N-i-j} 3^{i+j-N} \mathcal{EA}^{P}_{m,i,j,w}.\end{aligned} \tag{38}$$

Здесь

$$\begin{aligned}C^{m,l}_{r,s,w} &= wl^2(42m - 16 - 5m(\mu_{r+1} + \mu_{s+1}) \\ &\quad - 8F_m(\phi^l_{r,s}) - 4F_m(\chi^l_{r,s}) - 4F_m(\eta^l_{r,s})).\end{aligned}$$

В выражении $C^{m,l}_{r,s,w}$ учтено, что в матрицах $t_1, \ldots, t_7$ коэффициенты полиномов имеют тип $(2w)$, а в матрицых $a_{i,j}$ и $b_{i,j}$ – тип (w).

В частности, при $\alpha = 1$, когда алгоритм Карацубы применяется и для полиномов, и для чисел, получим

$$\mathcal{EM}^{MSt}_{n,m,1,w} = n^{\log_2 7}(mw)^{\log_2 3}.$$

3.3. Два модулярных алгоритма умножения матриц

Рассмотрим произведение матриц A и B типа (n, m, α, w). Будем считать, что $r = \lceil \log_h mn+2w \rceil$ наибольших простых чисел $p_1, \ldots, p_r$ типа (1) дают в произведении число, мажорирующее все числовые коэффициенты в произведении матриц. Сначала применим КТО для чисел: найдем образы матриц A и B при отображении $\mathbb{Z} \to \mathbb{Z}_{p_i} = \mathbb{Z}/p_i\mathbb{Z}$. Затем применим КТО для полиномов: найдем образы матриц при отображении $\mathbb{Z}_{p_i}[x] \to \mathbb{Z}_{p_i}[x]/m_i(x)\mathbb{Z}_{p_i}[x]$, выбирая, как и в п.2.4., $2m-1$ различных полиномов первой степени $m_i(x) = x - h_i \in \mathbb{Z}_{p_i}[x]$. Всего получим $(2m-1)r$ различных образов. Восстановление полученных произведений по КТО дает искомое произведение матриц. Найдем число арифметических операций.

(a) Для отображения $\mathbb{Z} \to \mathbb{Z}_{p_i}$ одного числа типа (w) требуется w операций деления с остатком и столько же вычитаний. Поэтому математическое ожидание числа делений и вычитаний для двух матриц по всем простым p_i, $i = 1, .., r$ составит $\mathcal{E}D_1 = \mathcal{E}A_1 = 2rn^2m\alpha w$.

(b) Для отображения $\mathbb{Z}_{p_i}[x] \to \mathbb{Z}_{p_i}[x]/m_i(x)\mathbb{Z}_{p_i}[x]$ одного полинома степени $m-1$, содержащего $m\alpha$ мономов, нужно m умножений и $m\alpha$ сложений, поэтому математическое ожидание числа умножений и числа сложений для r матриц по всем полиномам $m_i(x)$, $i = 2m-1$, составит $\mathcal{E}M_2 = 2rn^2(2m-1)m$ и $\mathcal{E}A_2 = \alpha\mathcal{E}M_2$.

(c) Для вычисления произведения каждой из $(2m-1)r$ пар матриц в случае применения стандартного алгоритма умножения требуется n^3 операций сложения и умножения, всего $\mathcal{E}A^S_3 = \mathcal{E}M^S_3 = rn^3(2m-1)$. В случае применения алгоритма Штрассена нужно $n^{\log_2 7}$ операций умножения и $6(n^{\log_2 7} - n^2)$ операций сложения. Всего $\mathcal{E}M^{St}_3 = rn^{\log_2 7}(2m-1)$ и $\mathcal{E}A^{St}_3 = 6r(n^{\log_2 7} - n^2)(2m-1)$.

(d) Для восстановления каждого из rn^2 полиномов по $2m-1$ модулям первого порядка $x - q_i$ при использовании схемы Лагранжа с предварительно вычисленными всеми необходимыми коэффициентами требуется по $(2m-1)^2$ операций умножения и сложения. Всего $\mathcal{E}A_4 = \mathcal{E}M_4 = rn^2(2m-1)^2$.

(e) Для восстановления в $\mathbb{Z}$ $2m-1$ коэффициентов n^2 полиномов по r простым модулям p_i по схеме Лагранжа с предварительно вычисленными всеми необходимыми коэффициентами требуется $\mathcal{E}M_5 = r^2(2m-1)n^2$ операций умножения и $\mathcal{E}A_5 = 2\mathcal{E}M_5$ операций сложения.

Найдем общее число операций. При этом будем учитывать, что для каждой операции сложения и умножения в $\mathbb{Z}_{p_i}$ требуется дополнительная операции деления с остатком, поэтому в пп. (b), (c), (d) нужно добавить $\mathcal{E}D_i = \mathcal{E}A_i + \mathcal{E}M_i$, $i = 2, 3, 4$.

Предложение 12. *Математические ожидания числа операций умножения, сложения и деления в модулярном алгоритме произведения полиномиальных матриц над $Z[x]$ в случае применения стандартного алгоритма умножения матриц по каждому модулю будут равны*

$$\begin{aligned}
\mathcal{EM}^{crtS}_{n,m,\alpha,w} &= rn^2(2m-1)(4m+n+r-1),\\
\mathcal{EA}^{crtS}_{n,m,\alpha,w} &= rn^2(2m\alpha w + (2m-1)(2m(1+\alpha)+n+2r-1)),\\
\mathcal{ED}^{crtS}_{n,m,\alpha,w} &= rn^2(2m\alpha w + (2m-1)(2m(3+\alpha)+2n-2)),
\end{aligned}$$

а в случае применения алгоритма Штрассена – будут равны

$$\begin{aligned}
\mathcal{EM}^{crtSt}_{n,m,\alpha,w} &= rn^2(2m-1)(4m+n^{\log_2 7-2}+r-1),\\
\mathcal{EA}^{crtSt}_{n,m,\alpha,w} &= rn^2(2m\alpha w + (2m-1)(2m(1+\alpha)+6n^{\log_2 7-2}+2r-7)),\\
\mathcal{ED}^{crtSt}_{n,m,\alpha,w} &= rn^2(2m\alpha w + (2m-1)(2m(3+\alpha)+7n^{\log_2 7-2}-8)).
\end{aligned}$$

3.4. Сравнение алгоритмов умножения матриц над $\mathbb{Z}[x]$

Полученные аналитические выражения для математического ожидания числа операций в рассмотренных алгоритмах умножения матриц позволяют провести сравнение этих алгоритмов для заданного типа (n, m, α, w) матриц и сделать прогноз для реальных расчетов.

Приведем результаты сравнения следующих десяти алгоритмов. (0). Стандартный алгоритм умножения матриц со стандартными алгоритмами умножения полиномов и чисел. (1). Стандартный алгоритм умножения матриц со стандартным алгоритмом умножения полиномов и алгоритмом Карацубы для умножения чисел. (2). Стандартный алгоритм умножения матриц с алгоритмом Карацубы для умножения полиномов и стандартными алгоритмами умножения чисел. (3). Стандартный алгоритм умножения матриц с алгоритмом Карацубы для умножения полиномов и чисел. (4). Алгоритм Штрассена

умножения матриц со стандартными алгоритмами умножения полиномов и чисел. (5). Алгоритм Штрассена умножения матриц со стандартным алгоритмом умножения полиномов и алгоритмом Карацубы для умножения чисел. (6). Алгоритм Штрассена умножения матриц с алгоритмом Карацубы для умножения полиномов и стандартными алгоритмами умножения чисел. (7). Алгоритм Штрассена умножения матриц с алгоритмом Карацубы для умножения полиномов и чисел. (8). Модулярный алгоритм со стандартным алгоритмом умножения матриц для каждого модуля. (9). Модулярный алгоритм, в котором применяется алгоритм Штрассена умножения матриц для каждого модуля.

В приведенных ниже таблицах для заданных значений n, m, α, w указан номер алгоритма (N) $(0 \le N \le 9)$, которой имеет наименьшую сложность, и отношение количества арифметических операций в стандартном алгоритме (0) по отношению к алгоритму (N). При этом суммируется число операций сложения, умножения и деления, причем число операций деления предварительно умножается на 10, так как примерно такое соотношение по времени эти операции показывают в экспериментах.

Таблица 4. Лучшие алгоритмы для умножения матриц типа (n, m, α, w) над $\mathbb{Z}[x]$ по числу арифметических операций и их выигрыш по отношению к стандартному алгоритму (0).

$n = 4$	$w = 4$				$w = 16$				$w = 64$			
$m\backslash\%$	1	10	50	100	1	10	50	100	1	10	50	100
4	0	0	0	6 1.5	0	0	0	6 2.	1 1.1	1 1.1	1 1.1	7 2.5
16	0	0	0	6 2.5	0	0	2 1.	6 3.4	1 1.1	1 1.1	3 1.3	7 4.3
64	0	0	6 1.1	6 4.2	0	0	6 1.6	6 6.	1 1.1	1 1.1	7 2.	7 7.6
256	0	0	6 1.9	6 7.3	0	0	6 2.7	6 11	1 1.1	1 1.1	7 3.5	7 14
1024	0	0	6 3.3	6 13	0	0	6 4.8	6 19	1 1.1	1 1.1	7 6.1	7 24
4096	0	0	6 5.8	6 23	0	0	6 8.4	6 33	1 1.1	1 1.1	7 11	7 43

$n = 16$	$w = 4$				$w = 16$				$w = 64$			
$m\backslash\%$	1	10	50	100	1	10	50	100	1	10	50	100
4	0	0	0	6 1.9	0	0	0	6 2.6	1 1.1	1 1.1	1 1.1	7 3.2
16	0	0	0	6 3.1	0	0	8 1.5	8 4.7	1 1.1	1 1.1	8 2.3	8 6.6
64	0	0	6 1.4	6 5.4	0	0	8 2.5	8 9.	1 1.1	1 1.1	8 5.9	8 19
256	0	0	6 2.4	6 9.5	0	0	6 3.5	6 14	1 1.1	1 1.1	8 9.9	8 36
1024	0	0	6 4.2	6 17	0	0	6 6.2	6 24	1 1.1	1 1.1	8 12	8 46
4096	0	0	6 7.5	6 30	0	0	6 11	6 44	1 1.1	1 1.1	7 14	7 56

$n = 64$	$w = 4$				$w = 16$				$w = 64$			
$m\backslash\%$	1	10	50	100	1	10	50	100	1	10	50	100
4	0	0	8 1.	8 2.9	0	0	8 1.8	8 5.3	1 1.1	1 1.1	8 2.5	8 7
16	0	0	8 1.9	8 6.7	0	0	8 4.7	8 16	1 1.1	1 1.1	8 8.3	8 24
64	0	0	8 2.9	8 11	0	0	8 9.	8 33	1 1.1	8 1.1	8 22	8 72
256	0	0	8 3.4	8 13	0	0	8 12	8 46	1 1.1	8 1.7	8 39	8 140
1024	0	0	6 5.5	6 22	0	0	8 13	8 51	1 1.1	8 2.	8 48	8 185
4096	0	0	6 9.8	6 39	0	0	6 14	6 57	1 1.1	8 2.	8 50	8 200

$n = 256$	$w = 4$				$w = 16$				$w = 64$			
$m\backslash\%$	1	10	50	100	1	10	50	100	1	10	50	100
4	0	0	8 1.5	8 4.5	0	0	8 3.6	8 12	1 1.	1 1.1	8 7.2	8 22
16	0	0	8 3.6	8 13	0	0	8 10	8 37	1 1.	8 1.4	8 24	8 77
64	0	0	8 7.8	8 30	0	8 1.3	8 26	8 96	1 1.	8 3.5	8 70	8 237
256	0	0	8 12	8 46	0	8 1.8	8 42	8 162	1 1.1	8 6.1	8 139	8 509
1024	0	0	8 14	8 54	0	8 2.	8 49	8 196	1 1.1	8 7.6	8 184	8 714
4096	0	0	8 14	8 56	0	8 2.1	8 52	8 207	1 1.1	8 8.1	8 200	8 795

$n = 1024$	$w = 4$				$w = 16$			$w = 64$		
$m\backslash\%$	1	10	50	100	1	10	100	1	10	100
4	0	0	9 1.8	9 5.7	0	0	9 18	1 1.	1 1.1	9 49
16	0	0	9 4.9	9 18	0	9 1.1	9 61	1 1.	9 2.6	9 173
64	0	0	9 15	9 57	0	9 2.4	9 195	1 1.	9 7.5	9 590
256	0	9 1.5	9 33	9 130	0	9 5.	9 460	1 1.1	9 18	9 1532
1024	0	9 2.	9 48	9 191	0	9 7.1	9 698	1 1.1	9 27	9 2562
4096	0	9 2.2	9 54	9 217	0	9 8.1	9 802	1 1.1	9 31	9 3080

$n = 4096$	$w = 4$				$w = 16$			$w = 64$		
$m\backslash\%$	1	10	50	100	1	10	100	1	10	100
4	0	0	9 2.4	9 7.7	0	0	9 26	1 1.	9 1.6	9 87
16	0	0	9 6.9	9 26	0	9 23	9 90	1 1.	9 82	9 310
64	0	9 1.5	9 23	9 91	0	9 3.9	9 326	1 1.	9 13	9 1136
256	0	9 3.2	9 70	9 278	0	9 11	9 1007	1 1.	9 39	9 3598
1024	0	9 6.	9 145	9 580	0	9 22	9 2125	1 1.1	9 82	9 7915
4096	0	9 8.1	9 199	9 797	0	9 30	9 2943	9 1.2	9 115	9 11316

Результаты экспериментов с соответствующими десятью программами показали результаты по времени вычислений, которые в среднем отличались на 25% от тех, что представлены в первой половине таблицы 4.

Как видно по таблице 4, асимптотически лучшим для плотных матриц является модулярный алгоритм (9), в котором применяется умножение по алгоритму Штрассена для каждого модуля. Для большого класса задач лучшими является или алгоритм (8) – модулярный алгоритм со стандартным умножением матриц, или алгоритмы (6) и (7), в которых применяется алгоритм умножения Штрассена для матриц и алгоритм умножения Карацубы для полиномов, при этом в алгоритме (6) числа умножаются по стандартному алгоритму, а в алгоритме (7) по алгоритму Карацубы.

Остальные четыре алгоритма (2)–(5) практически не имеют преимуществ ни для какого типа матриц.

4. Алгоритмы для плотных полиномов и матриц над $\mathbb{Z}[x]$

4.1. Алгоритмы умножения плотных полиномов

Отдельный интерес представляют плотные полиномы. В рассмотренных выше алгоритмах умножения сложность существенно зависела от степени разреженности полиномов (α). Теперь мы сравним эти алгоритмы отдельно в случае плотных полиномов. А также

рассмотрим еще один модулярный алгоритм, использующий быстрое преобразование Фурье. Сложность последнего алгоритма практически не зависит от степени разреженности полиномов, поэтому мы исследуем его в настоящем разделе.

Будем сравнивать следующие пять алгоритмов.

(0). Стандартный алгоритм умножения чисел и полиномов (PSS). (1). Алгоритм Карацубы для умножения чисел и стандартный алгоритм умножения полиномов (PSK). (2). Стандартный алгоритм умножения чисел и алгоритм Карацубы для умножения полиномов (PKS). (3). Алгоритм Карацубы для умножения как чисел, так и полиномов (PKK). (4). Модулярный алгоритм умножения полиномов, использующий дискретное преобразование Фурье [7] (PF), который рассмотрим ниже.

4.2. Модулярный алгоритм умножения полиномов, использующий дискретное преобразованием Фурье

Рассмотрим произведение полиномов f и g типа $(m, 1, w)$ в $Z[x]$. Будем считать, что $r = \lceil \log_h m + 2w \rceil$ простых чисел $p_1, \ldots, p_r$ типа (1) дают в произведении число, мажорирующее все числовые коэффициенты в произведении полиномов.

Найдем образы полиномов f и g при отображении $\mathbb{Z} \to \mathbb{Z}_{p_i} = \mathbb{Z}/p_i\mathbb{Z}$, $i \in \{1, \ldots, r\}$. Затем перемножим полученные полиномы, используя дискретное преобразование Фурье, и восстановим по КТО искомое произведение полиномов.

Так как степень произведения fg равна $2m-2$, то количество точек для ДПФ будет равно $N = 2^{\lceil \log_2(2m-1) \rceil}$ – ближайшей к $2m-1$ степени числа 2.

Пусть $\hat{f} = F^N_{p_i}(f)$ и $\hat{F}^N_{p_i}(\hat{f})$ – N-мерные векторы прямого и обратного дискретных преобразований Фурье для полинома f в поле Z_{p_i} на N точках. Здесь f рассматривается как вектор коэффициентов полинома. Легко показать, что для любого f выполняется равенство $Nf = \hat{F}^N_{p_i}(F^N_{p_i}(f))$ и произведение полиномов f и g в поле Z_{p_i} вычисляется по формуле $\hat{F}^N_{p_i}(F^N_{p_i}(f) \cdot F^N_{p_i}(g))/N$, где операция "$\cdot$" обозначает поэлементное умножение двух векторов длины N.

Найдем число арифметических операций на каждом этапе.

(a) Вычисление образов полиномов f и g при отображении $\mathbb{Z} \to \mathbb{Z}_{p_i}$, $i \in \{p_1, \ldots, p_r\}$: по $2mrw$ операций деления и вычитания.

(b) Вычисление ДПФ для полиномов f и g в $Z_{p_i}[x]$ $i = 1, \ldots, r$ по алгоритму Cooley–Tukey [7] на $N = 2^{\lceil \log_2(2m-1) \rceil}$ точках: по $2Nr \log_2 N$

операций сложения и умножения и вдвое большее число операций деления.

(c) Поэлементное умножение ДПФ-образов полиномов f и g в каждом поле Z_{p_i}: по rN операций умножения и деления.

(d) Вычисление обратного преобразования Фурье в каждом поле Z_{p_i}: по $rN\log_2 N$ операций сложения и умножения и вдвое большее число операций деления.

(e) Восстановление коэффициентов результата по КТО: $r^2(2m-1)$ операций умножения, вдвое больше число операций сложения.

4.3. Сравнение алгоритмов умножения плотных полиномов над $\mathbb{Z}[x]$

В предыдущих разделах были получены выражения сложности для алгоритмов $0-3$ с учетом разреженности входных данных. Приведем в таблице 5 эти оценки для случая $\alpha = 1$. Будем обозначать число операций сложения – $\mathcal{A}$, умножения – $\mathcal{M}$ и деления – $\mathcal{D}$:

Таблица 5. Число операций сложения, умножения и деления при умножении плотных полиномов для пяти алгоритмов

0	$\mathcal{A}$	m^2w^2
PSS	$\mathcal{M}$	$m^2(w^2+2w)$
1	$\mathcal{A}$	$10m^2(w^{\log_2 3}-w)$
PSK	$\mathcal{M}$	$m^2w^{\log_2 3}$
2	$\mathcal{A}$	$w^2(10m^{\log_2 3}-14m+4)$
PKS	$\mathcal{M}$	$m^{\log_2 3}w^2$
3	$\mathcal{A}$	$w(10m^{\log_2 3}-14m+4)+(mw)^{\log_2 3}$
PKK	$\mathcal{M}$	$(mw)^{\log_2 3}$
4	$\mathcal{A}$	$r(2mw+3N\log_2 N+2r(2m-1))$
PF	$\mathcal{M}$	$r(3N\log_2 N+N+r(2m-1))$
	$\mathcal{D}$	$r(2mw+6N\log_2 N+N)$

Для отдельных значений m и w можно составить таблицу.

Эксперименты с соответствующими программами показали в среднем отличие от таблицы 6 на 36%.

4.4. Модулярный алгоритм умножения матриц, использующий дискретное преобразование Фурье

Рассмотрим способ вычисления произведения матриц над $\mathbb{Z}[x]$, использующий дискретное преобразования Фурье (MFC). Он заключается в следующем. Сначала применяется КТО для чисел: находятся

Таблица 6. Номера лучших алгоритмов при умножении плотных полиномов над $\mathbb{Z}$ и их выигрыш по отношению к стандартному алгоритму

m	$w=4$	$w=16$	$w=64$
16	2/2.1	2/2.7	3/3.4
64	2/3.3	2/4.7	3/5.9
256	2/5.7	2/8.2	3/10.4
1024	2/10.0	2/14.5	4/20.4
4096	2/17.6	4/43.9	4/78.2
16384	4/61.8	4/160.4	4/300.0

образы матриц при отображении $\mathbb{Z} \to \mathbb{Z}_{p_i} = \mathbb{Z}/p_i\mathbb{Z}$ как в алгоритме в п. 2.4. Затем для каждого полинома в $\mathbb{Z}_{p_i}[x]$ находится ДПФ-образ. Затем вычисляется произведение матриц с использованием этих образов элементов матриц. Результат восстанавливается сначала обратным преобразованием Фурье и затем по КТО.

Найдем количество арифметических операций в этом алгоритме. В данном алгоритме шаги (a) и (e) такие же, как в модулярных алгоритмах в разделе 3.3. Для трех средних шагов (b), (c), (d) найдем число арифметических операций.

(b) Вычисление ДПФ для $2rn^2$ полиномов требует $2rn^2N\log_2 N$ операций сложения и умножения.

(c) Вычисление произведения r пар матриц для ДПФ-образов в каждой из N точек требует rn^3N операций сложения и умножения.

(d) Вычисление обратного преобразования Фурье для всех элементов всех r матриц требует $rn^2N\log_2 N$ операций сложения и умножения.

Так как операции сложения и умножения Z_{p_i} требуют дополнительно деления с остатком, то всего для трех этих шагов нужно $6rn^2N\log_2 N + 2rn^3N$ операций деления.

Предложение 13. *Математические ожидания числа операций умножения, сложения и деления в модулярном алгоритме произведения полиномиальных матриц над $Z[x]$, использующем дискретное преобразование Фурье, будут равны, соответственно,*

$$\mathcal{EM}^{crt}_{n,m,\alpha,w} = rn^2(r(2m-1) + 3N\log_2 N + nN),$$
$$\mathcal{EA}^{crt}_{n,m,\alpha,w} = rn^2(2m\alpha w + 2r(2m-1) + 3N\log_2 N + nN),$$
$$\mathcal{ED}^{crt}_{n,m,\alpha,w} = rn^2(2m\alpha w + 6N\log_2 N + 2nN).$$

Полученные выражения показывают, что данный алгоритм имеет небольшое преимущество перед алгоритмом (8), когда размеры матриц ограничены, а числовые коэффициенты в полиномах растут. Для сравнения модулярного алгоритма (8) с данным алгоритмом составим таблицу для отношения числа арифметических операций.

Таблица 7. Отношение количества арифметических операций в модулярном алгоритме (8) к количеству арифметических операций в модулярном алгоритме, использующем дискретное преобразование Фурье, для матриц типа $(n, m, 1, w)$

$w = 8$					
m	$n = 4$	$n = 8$	$n = 16$	$n = 32$	$n = 64$
4	2.33	2.14	1.87	1.58	1.32
8	2.52	2.33	2.06	1.75	1.46
16	2.94	2.73	2.41	2.03	1.66
32	3.84	3.55	3.12	2.58	2.05
64	5.63	5.2	4.53	3.68	2.8
128	9.09	8.38	7.27	5.83	4.3

$w = 16$					
m	$n = 4$	$n = 8$	$n = 16$	$n = 32$	$n = 64$
4	2.81	2.63	2.35	1.99	1.63
8	2.94	2.77	2.5	2.14	1.77
16	3.23	3.05	2.76	2.37	1.95
32	3.88	3.66	3.31	2.82	2.28

Эксперименты, в которых вычислялось отношение времени вычисления произведения соответствующих матриц показали различие с таблицей 28 в среднем на 25%.

5. Заключение

Известно большое число матричных и полиномиальных алгоритмов умножения и их асимптотические оценки сложности. В настоящей

работе сделана попытка получить реальные оценки для сложности некоторых алгоритмов, которые были бы применимы на практике для выбора лучшего алгоритма для заданного набора параметров.

С этой целью был проведен уточненный анализ алгоритмов в ходе всего вычислительного процесса и получены математические ожидания для количества арифметических операций в каждом алгоритме. Чтобы перевести полученные выражения для числа операций во время, необходимое для вычисления по заданному алгоритму, была принята модель, в которой операции сложения и умножения выполняются за одинаковое время, а деление с остатком требует в 10 раз больше времени.

Для проверки результатов для всех алгоритмов были написаны программы и замерялось время решения задач. Среднее расхождение между экспериментальными и теоретическими результатами для разных наборов алгоритмов колебались от 18% до 61%. Это можно считать очень хорошим результатом для столь простой модели пересчета числа арифметических операций во время вычислений.

Полученные выражения для математического ожидания числа арифметических операций в рассмотренных алгоритмах и составленные на их основе таблицы лучших алгоритмов для заданных типов полиномиальных матриц можно рассматривать как некоторые рекомендации для применения этих алгоритмов.

Дальнейшие исследования в этом направлении могут быть связаны с выбором более точной модели, которая определяет зависимость времени вычислений от количества арифметических операций в алгоритме, возможно с привязкой к конкретной модели компьютера. Это позволит получить более точный прогноз для времени вычислений, и на этой основе можно будет описать процедуру для выбора лучшего алгоритма для заданного набора параметров задачи и заданной модели компьютера.

Литература

1. V. Strassen, *Gaussian Elimination is not optimal.* — Numer. Math. **13** (1969), 354–356.
2. А. Ахо, Дж. Хопкрофт, Дж. Ульман, *Построение и анализ вычислительных алгоритмов.* Мир, М., 1979.
3. Д. Э. Кнут, *Исскуство программирования.* Т. 2. Получисленные алгоритмы, 3-е изд., Издательский дом “Вильямс”, М., 2001.
4. Г. И. Малашонок, Е. С. Сатина, *Быстрое умножение и разреженные структуры.* — Программирование **2** (2004), 1–5.

5. G. I. Malaschonok, *Complexity Considerations in Computer Algebra.* — In: Computer Algebra in Scientific Computing, CASC 2004. Techn. Univ. Munchen, Garching, Germany (2004), pp. 325–332.
6. Г. И. Малашонок, *Сложность быстрого умножения на разреженных структурах.* — В кн.: Алгебра, логика и кибернетика (Материалы международной конференции). Изд-во ГОУ ВПО "ИГПУ", Иркутск (2004), 175–177.
7. П. Ноден, К. Китте, *Алгебраическая алгоритмика (с упражнениями и решениями).* Пер. с франц. Мир, М., 1999.
8. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms.* MIT Press, 2002.

Malaschonok G. I., Valeev Yu. D., Lapaev A. O. On the choice of multiplication algorithm for polynomials and polynomial matrices.

The multiplication algorithm for dense and sparse polynomials and polynomial matrices of different numerical domains are investigated. The expressions for complexity of multiplication operations of polynomials and polynomial matrices are obtained. Each of these expressions is an average of distribution for machine arithmetic operation numbers. Expressions of complexity for a set of parameters which has a practical interest are presented. The results of experiments with the respective programs are demonstrated.

Тамбовский государственный
университет им. Г. Р. Державина
ул. Интернациональная, 33,
392622 Тамбов, Россия

E-mail: malaschonok@ya.ru

Поступило 30 ноября 2009 г.