УДК 519.6:519.712.3:510.52 **Зінченко Я.В.**

МЕТОДИ ОБЧИСЛЕННЯ ДОБУТКУ БАГАТОРОЗРЯДНИХ ЧИСЕЛ ТА ЇХ ОПТИМІЗАЦІЯ. ЧАСТИНА І

Проведено аналіз відомих методів обчислення добутку багаторозрядних чисел, укладено їх класифікацію, наведено апріорні оцінки обчислювальної складності. Визначено області ефективного використання методів, дані рекомендації щодо їх застосування при розв'язанні прикладних задач. Здійснено пошук можливості та шляхів оптимізації методів.

Ключові слова: оптимізація, методи обчислення добутку

На даний час існує багато прикладних задач, при розв'язанні яких активно використовується арифметика багаторозрядних чисел. До них належать задачі двоключової криптографії, спектрального і кореляційного аналізу та фільтрації цифрових сигналів, аеро- та гідродинаміки, розрахунку оболонок ядерних реакторів, моделювання фізичних, хімічних (біохімічних) процесів, обробки даних біофізичного експерименту та інші.

Під m-розрядним цілим числом треба розуміти довільне ціле число, менше за b^m, де b – основа прийнятої позиційної системи, в якій представляються числа. Такі числа в цій системі записуються з використанням не більше ніж m розрядів.

Багаторозрядні числа – це числа, які записані в системі числення за основою b, де b – розмір машинного слова. Наприклад, ціле число, яке займає 10 машинних слів в пам'яті ЕОМ, розмір слова якої дорівнює $b = 10^{10}$, має 100 десяткових цифр; однак його треба розглядати як десятирозрядне число за основою 10^{10}. Такий підхід обґрунтовано тими ж розуміннями, що й перехід від двійкової системи числення до шістнадцятиричної шляхом групування бітів. Багаторозрядні цілі числа також можна називати багатослівними або s-слівними, оскільки кожне з них може бути представлене у вигляді масиву 16-, 32- або 64-бітових слів. У позиційному представленні з основою b вони записуються як:

$$W = \sum_{i=0}^{s-1} W_i b^i , \qquad (1)$$

де $b = 2^p$ (p – кількість розрядів в машинному слові), а W_i – i-а значуща цифра числа W, яка представлена так, що задовольняє умові $0 \le W_i \le b-1$, $i = \overline{0, s-1}$, але $W_{s-1} \ne 0$.

Якщо у вищенаведеному співвідношенні $s > 2$, то виконання деяких операцій, таких як множення, ділення, обчислення експоненти за модулем та ін., потребує значних витрат машинного часу і виникає необхідність в оптимізації за часом виконання на ЕОМ зазначених операцій. При цьому, як правило, розглядаються два підходи до оптимізації: один ґрунтується на використанні суттєво різних, більш ефективних алгоритмів, інший – на розробці більш ефективної структури програм, які реалізують один і той же алгоритм. В даній роботі розглядається перший підхід. Слід зауважити, що істотним може бути ефект і від оптимізації за рахунок технічних особливостей та можливостей використовуваної техніки [1].

Однією з найбільш трудомістких операцій з багаторозрядними числами є операція обчислення добутку. На сьогоднішній день існує досить велика кількість методів його обчислення, кожен з яких має свою область ефективного застосування в залежності від області значень s, моделі обчислень, програмної чи апаратної реалізації. Усі ці методи є рекурсивними і засновані на зведенні множення багаторозрядних чисел до послідовності множень чисел з меншою кількістю розрядів. Класифікацію методів обчислення добутку багаторозрядних чисел наведено на рис. 1.

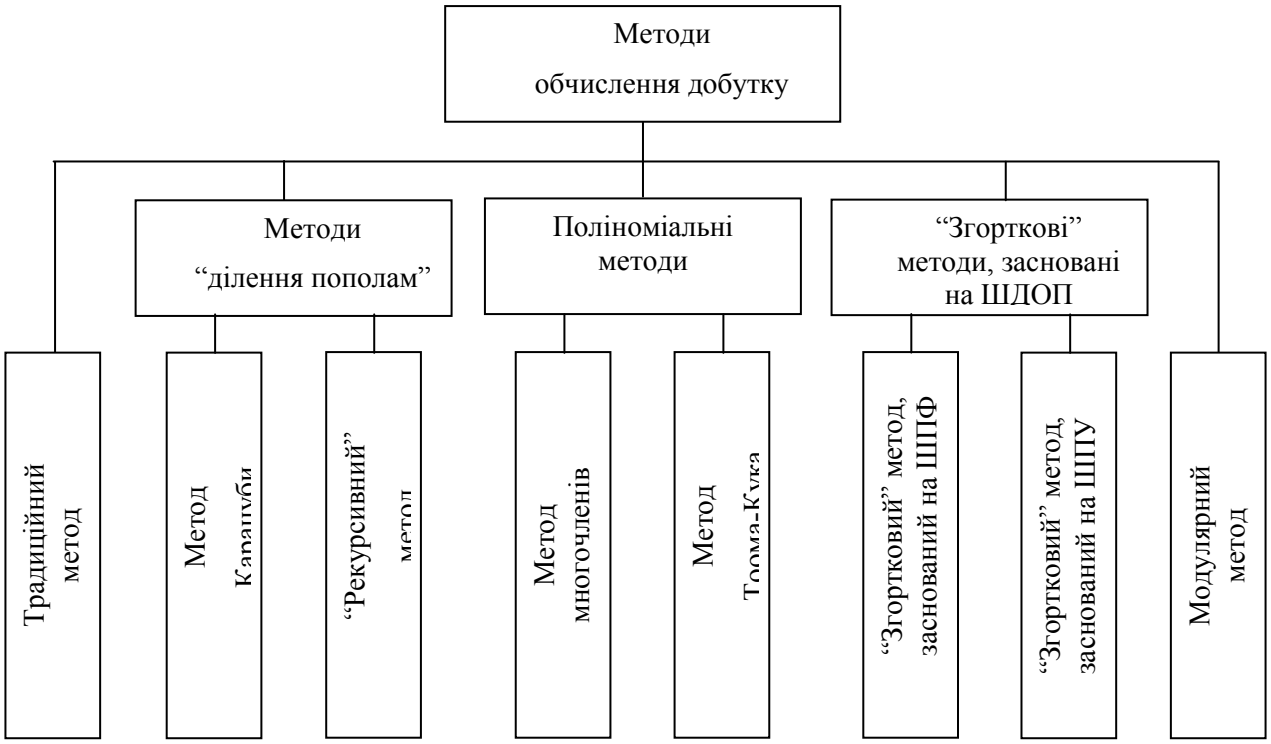

Рис. 1. Класифікація методів обчислення добутку багаторозрядних чисел

Розглянемо наведені методи множення багаторозрядних чисел по-порядку та порівняємо їх обчислювальну складність. Викладення буде вестись у відповідності з [2-10].

Традиційний метод. Нехай a і b два s-слівні числа, записані за основою w:

$$a = \left(a_{s-1}a_{s-2}\ldots a_0\right) = \sum_{i=0}^{s-1} a_i w^i \ ,$$

$$b = \left(b_{s-1}b_{s-2}\ldots b_0\right) = \sum_{i=0}^{s-1} b_i w^i \ ,$$

де a і b – цифри з проміжку $[0; w-1]$, а w може бути будь-яким цілим додатним числом. Зазвичай $w = 2^{\omega}$, ω – довжина машинного слова, наприклад, $\omega = 32$. Стандартний алгоритм добутку a і b знаходить часткові добутки, підсумовує їх і отримує кінцевий $2s$-слівний результат t. Нехай t_{ij} означає пару чисел (Перенос, Сума) – (C, S), як результат добутку $a_i \cdot b_i$. Пари t_{ij} можуть бути розміщені у вигляді таблиці наступним чином:

$$\begin{array}{cccccc} & & & a_2 & a_1 & a_0 \\ \times & & & b_2 & b_1 & b_0 \\ \hline & & & t_{02} & t_{01} & t_0 \\ & & t_{12} & t_{11} & t_{10} & \\ + & t_{22} & t_{21} & t_{20} & & \\ \hline t_5 & t_4 & t_3 & t_2 & t_1 & t_0 \end{array} .$$

Останній рядок означає суму часткових добутків і представляє собою $2s$-слівний добуток. Алгоритм по суті цифра за цифрою виконує описане множення і додавання. Для економії пам’яті використовується тільки одна змінна t для часткового добутку. Її початкове значення дорівнює 0; потім береться цифра множника b, множиться на a і підсумовується до часткового добутку t. Наприкінці обчислень ця змінна для часткових добутків містить остаточний добуток $a \cdot b$. Аналіз цього алгоритму показує, що загальне

число кроків внутрішнього добутку дорівнює s^2. Оскільки $s = m/\omega$, а ω – постійне для конкретного комп'ютера, то обчислення добутку двох m-розрядних двійкових чисел традиційним методом потребує $O_Б\left(m^2\right)$ бітових операцій [2].

Методи "ділення пополам". Використання підходу "ділення пополам", який вперше до обчислення добутку багаторозрядних чисел був застосований А. Крацубою, дозволяє зменшити асимптотичну складність до $O_Б\left(m^{1,59}\right)$ бітових операцій [3].

Нехай a і b – два m-розрядних двійкових числа; m – степінь числа 2. Розбиваючи їх записи на дві частини довжини $m/2$, отримуємо:

$$a \cdot b = \left(a_1 2^{m/2} + a_0\right)\left(b_1 2^{m/2} + b_0\right) = a_1 b_1 2^m + \left(a_1 b_0 + a_0 b_1\right)2^{m/2} + a_0 b_0 .$$

Оскільки множення на степені двійки і додавання – швидкі операції, то проблема зводиться до швидкого обчислення трьох білінійних форм $c_1 = a_1\, b_1$, $c_2 = a_1\, b_0 + a_0\, b_1$, $c_3 = a_0\, b_0$. Для цього достатньо обчислити білінійні форми c_1, c_3 і $d = \left(a_1 + a_0\right)\left(b_1 + b_0\right)$. При цьому $c_2 = d - c_1 - c_3$. Таким чином, задача множення двох m-розрядних чисел зводиться до трьох задач множення $(m/2)$-розрядних чисел (в d ще треба прибрати зайвий розряд) та декільком додаванням-віднімнням не більше ніж m-розрядних чисел. Для максимальної складності обчислення добутку двох m-розрядних двійкових чисел це дає рекурсивну нерівність для парних m:

$$L(m) \le 3L\left(\frac{m}{2}\right) + O(m),$$

із якої випливає, що $L(m) = O\left(m^{\log 3}\right) = O\left(m^{1,59}\right)$. Точна бітова складність методу Карацуби дорівнює: $M(m) = 3(m/4 - 1) + M(m/8)$; $M(4) = 25$, $M(8) = 103$, $M(l) = 3M(l/2) + 4(l-1)$.

Вищевикладену ідею рекурсивного зведення основної задачі до таких самих задач в константу разів меншого розміру в [4] названо прийомом "розділяй і володарюй". Вона використовується в "рекурсивному" методі обчислення добутку багаторозрядних чисел, складність якого аналогічна обчислювальній складності методу Карацуби.

Поліноміальні методи. В границях, де m прямує до нескінченності, час обчислення добутку багаторозрядних чисел може бути зменшений ще більше, якщо урахувати, що метод Карацуби ("рекурсивний" метод) є окремим випадком $r = 1$ більш загального методу, який називається методом многочленів і для будь-якого фіксованого r дає:

$$T\left((r+1)m\right) \le (2r+1)T(m) + km . \qquad (2)$$

Цей більш загальний метод може бути отриманий наступним чином.

Розіб'ємо числа $x = \left(x_{(r+1)m-1} \ldots x_1 x_0\right)_2$ і $y = \left(y_{(r+1)m-1} \ldots y_1 y_0\right)_2$ на $r+1$ частин:

$$x = X_r 2^{rm} + \ldots + X_1 2^m + X_0,\ y = Y_r 2^{rm} + \ldots + Y_1 2^m + Y_0,$$

де кожне X_j і кожне Y_j є m-бітовим числом. Розглянемо поліноми: $X(\hat{x}) = X_r \hat{x}^r + \ldots + X_1 \hat{x} + X_0$ і $Y(\hat{x}) = Y_r \hat{x}^r + \ldots + Y_1 \hat{x} + Y_0$. Укладемо: $Z(\hat{x}) = X(\hat{x})Y(\hat{x}) = Z_{2r} \hat{x}^{2r} + \ldots + Z_1 \hat{x} + Z_0$. Оскільки $x = X\left(2^m\right)$ і $y = Y\left(2^m\right)$, отримуємо $xy = Z\left(2^m\right)$. Очевидно, що при відомих коефіцієнтах Z_k в $Z(\hat{x})$ можна легко обчислити $x \cdot y$. Задача полягає в пошуку ефективного способу обчислення цих коефіцієнтів в $Z(\hat{x})$, який вимагає лише $2r+1$ множень m-бітових чисел і декілька наступних операцій, час виконання яких пропорційний m. Цього можна досягти за допомогою обчислення: $X(0)Y(0) = Z(0)$, $X(1)Y(1) = Z(1)$,…, $X(2r)Y(2r) = Z(2r)$.

Коефіцієнти поліному степені $2r$ можуть бути представлені у вигляді лінійної комбінації значень цього поліному в $2r+1$ різних точках. Час, який необхідний для виконання цієї операції, пропорційний m або менший. Насправді добутки $X(j)Y(j)$ не є в

строгому значенні добутками m-бітових чисел, але є добутками $(m+t)$-бітових чисел, де t – фіксоване значення, яке залежить від r. Програмі множення $(m+t)$-бітових чисел необхідне виконання лише $T(m)+c_1 m$ операцій, де $T(m)$ – кількість операцій, які необхідні для множення m розрядів, оскільки при фіксованому t два добутки t- і m-бітових чисел можна отримати за $c_2 m$ операцій. Таким чином, отримуємо метод множення, для якого виконується нерівність (2). Враховуючи цю нерівність, переходимо до нерівності: $T(m) \le c_3 m^{\log_{r+1}(2r+1)} < c_3 m^{1+\log_{r+1} 2}$.

Отже, для будь-якого $\varepsilon > 0$ існують така постійна $c(\varepsilon)$ і такий метод множення, що кількість бітових операцій $T(m)$, які необхідно виконати для обчислення добутку двох m-розрядних двійкових чисел, задовольняє оцінці:

$$T(m) < c(\varepsilon) m^{1+\varepsilon},$$

тобто, асимптотична бітова складність методу многочленів дорівнює $O_Б\left(m^{1+\varepsilon}\right)$.

Слід однак зауважити, що вищенаведений результат не є задовільним, оскільки при $\varepsilon \to 0$ (тобто $r \to \infty$) складність методу стрімко зростає. При малих значеннях m це призводить до швидкого збільшення $c(\varepsilon)$, тому виграш може бути отриманий тільки при дуже великих m.

Якщо припустити, що r варіюється разом з m, то, вибираючи в міру збільшення m все більші значення r, можна отримати кращий результат. Ця ідея була запропонована А. Тоомом і використовувалась для доведення того, що при зростаючому m є можливість побудувати автомат для обчислення добутку m-розрядних чисел, який би складався з невеликої кількості елементів. Згодом С. Кук показав, як застосувати ідею Тоома для прискорення роботи комп'ютерних програм. Складність методу Тоома-Кука дорівнює $O_Б\left(m 2^{\sqrt{2\log m}} \log m\right)$ бітових операцій [5].

Модулярний метод. У модулярному методі операції виконуються не з якимось конкретним числом (наприклад, u), а з його лишками $u \bmod m_1$, $u \bmod m_2, \ldots$, $u \bmod m_r$, де m_1, $m_2, \ldots$, m_r – модулі, які не містять спільних дільників, тобто вони взаємно прості. Введемо наступні позначення: $u_1 = u \bmod m_1$, $u_2 = u \bmod m_2, \ldots$, $u_r = u \bmod m_r$. Числа $(u_1, u_2, \ldots, u_r)$ обчислюються шляхом ділення числа u на прості цілі числа v_k. При цьому немає втрати інформації, оскільки знаючи $(u_1, u_2, \ldots, u_r)$, завжди можна відновити u. Числа $(u_1, u_2, \ldots, u_r)$ розглядаються як модулярне представлення цілого числа u.

Перевагою модулярного представлення є те, що операція обчислення добутку виконується досить просто:

$$(u_1, \ldots, u_r) \times (v_1, \ldots, v_r) = ((u_1 \times v_1) \bmod m_1, \ldots, (u_r \times v_r) \bmod m_r). \qquad (3)$$

Для доведення (3) треба лише показати, що для кожного модуля m_j виконується рівність: $uv \bmod m_j = (u \bmod m_j)(v \bmod m_j) \bmod m_j$. Ця рівність випливає з основного положення елементарної теорії чисел: $x \bmod m_j = y \bmod m_j$ лише тоді, коли $x \equiv y$ (по модулю m_j). Якщо $x \equiv x'$ і $y \equiv y'$, то $xy \equiv x'y'$ (по модулю m_j); звідси випливає $(u \bmod m_j) \times$ $\times (v \bmod m_j) \equiv uv$ (по модулю m_j).

Область чисел, над якими виконується операція модулярного множення, – це $m = m_1 m_2 \ldots m_r$ (добуток модулів). Якщо розмір кожного з m_j приблизно дорівнює розміру машинного слова, то можна оперувати n-розрядними числами, коли $r \approx n$. Звідси випливає, що при використанні модулярної арифметики загальний час, який витрачається на виконання операції множення з n-розрядними числами, пропорційний n (не враховуючи часу, що витрачається на перехід до модулярного представлення і навпаки).

Для обчислення многочлену (числа) r, який є добутком многочленів (чисел) p і q степені, меншої $m/2$, природно обчислити їх добуток по модулю x^m-1. Для цього необхідно спочатку обчислити лишки від ділення цих многочленів на лінійні попарно взаємно прості двочлени $x-e^k$, які є результатом розкладання многочленну x^m-1 на множники (наприклад, над полем $\mathbf{C}$), тобто обчислити значення $p\left(e^k\right)$ та $q\left(e^k\right)$, $k=0,\ldots,m-1$. Потім, попарно їх перемноживши, отримати значення $r\left(e^k\right)=p\left(e^k\right)q\left(e^k\right)$ і, застосовуючи "зворотний Китайський алгоритм", відновити многочлен r. Для спрощення обчислень m повинно дорівнювати степені числа 2 (при необхідності многочлени p і q з боку молодших коефіцієнтів додаються нулями).

Для обчислення значень $p\left(e^k\right)$ і $q\left(e^k\right)$, $k=0,\ldots,m-1$ спочатку обчислюються $p \bmod x^{m/2}-1$, $p \bmod x^{m/2}+1$, $q \bmod x^{m/2}-1$ і $q \bmod x^{m/2}+1$, потім обчислюються лишки за модулями $x^{m/4}+1$, $x^{m/4}-1$, $x^{m/4}+i$, $x^{m/4}-i$ і т.д., доки не будуть обчислені лишки за модулями x^2-e^{2k}, $k=0,\ldots,m/2$, і, нарешті, за модулями $x-e^k$, $k=0,\ldots,m$. Оскільки ділення многочлену степені меншої n на двочлен степені $n/2$ здійснюється "шкільним" алгоритмом зі складністю n, то точна складність всього алгоритму обчислення значень $p\left(e^k\right)$ і $q\left(e^k\right)$, $k=0,\ldots,m-1$, оцінюється як $2m\log m$ (множень та додавань, які виконуються в полі $\mathbf{C}$).

Для відновлення (інтерполяції) многочлену r по відомим його значенням $r\left(e^k\right)$, $k=0,\ldots,m-1$, природно застосувати формулу Лагранжа:

$$r=f(x)\sum_{k=0}^{m-1}\frac{r\left(e^k\right)/f'\left(e^k\right)}{x-e^k}=f(x)\sum_{k=0}^{m-1}\frac{r\left(e^k\right)e^k}{m\left(x-e^k\right)},$$

де $f(x)=x^m-1$. Оскільки при правильному виборі порядку сумування усі дроби будуть мати знаменники у вигляді двочлену, а множення многочлену степені меншої n на двочлен степені n "шкільним" методом має складність n, то додавання двох таких дробів має складність $4n$, а весь алгоритм інтерполяції – складність $2m+2m\log m$. Асимптотична бітова складність методу обчислення добутку багаторозрядних чисел, який заснований на Китайській теоремі про лишки, дорівнює $O_Б\left(m^{1,63}\right)$ операцій [6].

"Згорткові" методи. Метод обчислення добутку багаторозрядних чисел, що заснований на застосуванні дискретної згортки, вперше був запропонований А. Шенхаге і В. Штрассеном [7]. В ньому для швидкого обчислення згортки використовувалась теорема про дискретну згортку двох функцій і алгоритм швидкого обчислення дискретного перетворення Фур'є (ДПФ) – алгоритм швидкого перетворення Фур'є (ШПФ) Кулі-Т'юкі. Незалежно від авторів роботи [7] цю ж саму ідею висловлював Д. Кнут.

Дійсно, якщо розбити співмножники на блоки підходящої довжини і розглядати їх як елементи такого кільця $\mathbf{R}$, операції якого точно передають вихідні обчислення (в кільці $\mathbf{Z}$ цілих чисел) і яке, крім того, містить необхідні корені з одиниці, то обчислення добутку багаторозрядних чисел можна представити такою послідовністю дій: ДПФ послідовностей блоків для обох співмножників, покомпонентне множення перетворених послідовностей, обернене ДПФ і виконання переносів. До "малих" множень, які при цьому виникають, застосовуються аналогічні перетворення (таким чином отримуємо ітеративу процедуру). В якості $\mathbf{R}$, зазвичай, використовується поле $\mathbf{C}$ комплексних чисел, але можуть використовуватись і інші поля, наприклад, поле Галуа.

Нехай u і v – два m-розрядних двійкових цілих числа, m – степінь числа 2. Необхідно обчислити добуток $c=uv$. В результаті множення отримаємо $2m$-розрядне число. Виберемо натуральні числа l і K такі, що $lK\geq 2m$, K – степінь числа 2.

Розіб'ємо числа u і v на K блоків довжини l:

$$u = \sum_{j=0}^{K-1} u_j 2^{jl}, \qquad v = \sum_{j=0}^{K-1} v_j 2^{jl},$$

$$0 \le u_j < 2^l,\ 0 \le v_j < 2^l \text{ і } u_j = v_j = 0 \text{ при } j > K-1. \tag{4}$$

Тоді їх добуток c має вигляд:

$$c = uv = \left(\sum_{j=0}^{K-1} u_j 2^{jl} \right) \left(\sum_{j=0}^{K-1} v_j 2^{jl} \right) = \sum_{\tau=0}^{K-1} c_\tau 2^{\tau l}, \tag{5}$$

де

$$c_\tau = \sum_{\rho+\delta=\tau(\mathrm{mod}K)} u_\rho v_\delta \tag{6}$$

являє собою (при виконанні умови (4)) циклічну згортку u_ρ з v_δ. Для обчислення циклічної згортки c_τ згідно (6) використовуються теорема про дискретну згортку двох функцій і алгоритм ШПФ.

Покроковий опис алгоритму обчислення добутку $c = uv$, що заснований на застосуванні дискретної згортки і ШПФ, має наступний вигляд:

Вхідні дані: два m-розрядних двійкових цілих числа u і v, m – степінь числа 2.

Вихідні дані: $2m$-розрядне двійкове ціле число c.

1. Обчислити за допомогою алгоритму ШПФ дискретні перетворення Фур'є:

$$\hat{u}_s = \sum_{\rho=0}^{K-1} u_\rho w_k^{\rho s},\ \hat{v}_s = \sum_{\delta=0}^{K-1} v_\delta w_k^{\delta s},\ s = \overline{0,\ K-1},$$

де $w_k = \exp(2\pi i / K),\ i = \sqrt{-1}$.

2. Обчислити добуток: $\hat{c}_s = \hat{u}_s \hat{v}_s,\ s = \overline{0,\ K-1}$.

3. Обчислити за допомогою алгоритму ШПФ c_τ як обернене дискретне перетворення Фур'є вектору $\hat{c}_s$: $c_\tau = \frac{1}{2^k} \sum_{s=0}^{K-1} \hat{c}_s w_k^{-s\tau},\ \tau = \overline{0,\ K-1}$.

4. Обчислити добуток $c = uv$ згідно (5) виконуючи "збірку" (зсуви і додавання l-розрядних чисел).

Точна обчислювальна складність методу Шенхаге-Штрассена множення багаторозрядних чисел при використанні моделі "неветвящиеся программы" оцінюється як $16K\log K + O(K)$. Точна складність при використанні моделі "битовые вычисления" може бути оцінена як $15m/4 + 6F(K)$, де $F(K)$ – складність алгоритму ШПФ. Асимптотичну складність методу зазвичай наводять у вигляді $O_A(K\log K)$ не уточнюючи мультиплікативну константу, іноді відмічаючи, що вона порядку 20. Асимптотична бітова складність методу дорівнює $O_Б(m\log m\log\log m)$ [7, 8].

Оскільки обчислювальна складність наведеного методу множення залежить, в основному, від складності обчислення дискретної циклічної згортки, яка, у свою чергу, обчислюється за допомогою двох прямих і одного оберненого ДПФ, то вирішальнім фактором, що визначає складність методу Шенхаге-Штрассена, є складність обчислення дискретних перетворень Фур'є. У зв'язку з цим в [9] був розвинений підхід Шенхаге і Штрассена до швидкого множення багаторозрядних чисел в частині використання для його реалізації оригінальної модифікації алгоритму ШПФ з попередньою заготовкою елементів матриці перетворення, який має особливості і переваги у порівнянні з іншими відомими алгоритмами. Його застосування дозволяє суттєво зменшити кількість операцій при обчисленні добутку багаторозрядних чисел, а складність самого методу оцінюється як:

$$T^{\times} = 3K(\log K - 1) - 16,$$

$$T^{+} = K(9\log K + 5)/2 - 9,$$

де $T^{\times}$ – кількість “елементарних” операцій множення, T^{+} – кількість “елементарних” операцій додавання. Слід зауважити, що дані оцінки, на відміну від оцінок складності Шенхаге-Штрассена [7], наведено для кількості операцій над l-розрядними числами.

Використання в методі Шенхаге-Штрассена алгоритму ШПФ пов’язане з деякими обчислювальними труднощами, оскільки даний алгоритм розроблений для поля комплексних чисел, а перемножуються цілі багаторозрядні числа. До таких труднощів варто віднести витрати машинного часу на обчислення тригонометричних функцій, а також боротьбу з помилками заокруглення при обчисленні $w_k = \exp(2\pi i / K)$, $i = \sqrt{-1}$. Позбавлення від вказаних труднощів, а також подальше зменшення точної обчислювальної складності методу множення (константи в знаці O) можливе за рахунок використання більш ефективних алгоритмів ШПФ, що реалізовані у скінчених полях (наприклад, в полі Галуа), або застосування ефективних алгоритмів обчислення дискретної циклічної (лінійної) згортки, реалізації яких виключають перехід в поле комплексних чисел. У зв’язку з цим в [10] був запропонований метод множення багаторозрядних чисел, заснований на ефективному алгоритмі обчислення дискретної циклічної згортки, реалізація якого виключає перехід в поле комплексних чисел. В основу вказаного алгоритму згортки покладено трансформацію просторів вхідних даних $K = 2^k$ в простори відповідних коефіцієнтів Уолша вимірності $2 \cdot 3^{k-1}$ і їх комбінацій у вигляді сум і різниць. Для ефективного обчислення коефіцієнтів Уолша в ньому використовується алгоритм швидкого перетворення Уолша (ШПУ).

Введемо деякі позначення. Скінченні часові ряди $u_0, \ldots, u_{K-1}$ і $v_0, \ldots, v_{K-1}$ довжини $K = 2^k$ представимо векторами-стовбцями:

$$u = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{K-1} \end{bmatrix}, \qquad v = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{K-1} \end{bmatrix}.$$

Циклічна згортка u і v – часовий ряд довжини K, який має вигляд:

$$c_\tau = \sum_{\gamma=0}^{K-1} u_\gamma v_{\gamma+\tau}, \quad \tau = \overline{0,\, K-1}. \tag{7}$$

Для спрощення позначень, згортку (7) запишемо у вигляді: $c = u * v$.

Позначимо вектори циклічного зсуву у вигляді:

$$u' = \begin{bmatrix} u_1 \\ \vdots \\ u_{K-1} \\ u_0 \end{bmatrix}, \qquad v' = \begin{bmatrix} v_1 \\ \vdots \\ v_{K-1} \\ v_0 \end{bmatrix}.$$

Набір матриць $\boldsymbol{E}$, $\boldsymbol{O}$, $\boldsymbol{P}$, $\boldsymbol{S}$, що перетворюють вектор довжини K в вектор довжини $K/2$, визначимо як:

$$Eu = \begin{bmatrix} u_0 \\ u_2 \\ \vdots \\ u_{K-2} \end{bmatrix}, \quad Ou = \begin{bmatrix} u_1 \\ u_3 \\ \vdots \\ u_{K-1} \end{bmatrix}, \quad Pu = \begin{bmatrix} u_0 + u_1 \\ u_2 + u_3 \\ \vdots \\ u_{K-2} + u_{K-1} \end{bmatrix}, \quad Su = \begin{bmatrix} u_0 - u_1 \\ u_2 - u_3 \\ \vdots \\ u_{K-2} - u_{K-1} \end{bmatrix}.$$

Відмітимо, що: $Pu = Eu + Ou,\ Su = Eu - Ou,\ Pv = Ev + Ov,\ Sv = Ev - Ov$.

Оскільки довжина вектору після перетворення будь-яким з наведених вище операторів стає вдвічі меншою, то перетворення припиняються, коли u стає одномірним вектором.

Розглянемо допоміжні (короткі) згортки a, d і f виду:

$$\begin{aligned} a &= Pu \cdot Pv = (E+O)u \cdot (E+O)v, \\ d &= Su \cdot Sv = (E-O)u \cdot (E-O)v, \\ f &= (P-S)u \cdot Ev = 2 \cdot Ou \cdot Ev. \end{aligned} \tag{8}$$

Тоді:

$$2Ec = a + d,\quad 2Oc = a - d - 2(f' - f). \tag{9}$$

Таким чином, необхідну згортку c (без множника 2) отримуємо за допомогою лінійної комбінації трьох допоміжних згорток: a, d і f.

Основною властивістю розглядуваного алгоритму (8) і (9), який являє собою скінчений ітераційний процес, є те, що на останньому його кроці із вхідних послідовностей u і v довжини $K = 2^k$ отримуємо $q_k = 2 \cdot 3^{k-1}$ одномірних векторів (чисел), які покомпонентно перемножуються. Ці числа – це коефіцієнти Уолша або їх лінійні комбінації. Для ефективного обчислення коефіцієнтів Уолша використовується алгоритм ШПУ.

Покроковий опис алгоритму обчислення добутку $c = uv$, що заснований на застосуванні дискретної згортки і ШПУ, має наступний вигляд:

Вхідні дані: два m-розрядних двійкових цілих числа u і v, m – степінь числа 2.

Вихідні дані: $2m$-розрядне двійкове ціле число c.

1. Обчислити за допомогою алгоритму ШПУ дискретні перетворення Уолша:

$$F_s^u = \sum_{j=0}^{K-1} u_j wal(s, j),\ F_s^v = \sum_{j=0}^{K-1} v_j wal(s, j),\ s = \overline{0,\ K-1},$$

де $wal(s, j)$ – функції Уолша, що впорядковані за Уолшем.

2. Обчислити допоміжний масив P_k^u, який містить коефіцієнти Уолша F_s^u і їх лінійні комбінації.

3. Обчислити допоміжний масив P_k^v, який містить коефіцієнти Уолша F_s^v і їх лінійні комбінації. При цьому оператор O замінюється на оператор E.

4. Обчислити допоміжні згортки a, d і f.

5. Обчислити необхідну згортку c_τ за допомогою лінійної комбінації трьох допоміжних згорток a, d і f.

6. Обчислити добуток $c = uv$ виконуючи "збірку" (зсуви і додавання l-розрядних чисел).

Точна обчислювальна складність вищенаведеного методу множення багаторозрядних чисел, що заснований на застосуванні дискретної згортки і ШПУ, оцінюється як:

$$Q_{FFW}^{\times} = 2 \cdot 3^{k-1}, \tag{10}$$

$$\begin{aligned} Q_{FFW}^{+} = Q_{FFW_{ЗГ}}^{+} + Q_{FFW_{ЗБ}}^{+} &= 13 \cdot 3^{k-1} + 2^{k+1}(k - 2{,}75) + (2^k - 1) = \\ &= 13 \cdot 3^{k-1} + 2^{k+1}(k - 2{,}25) - 1 \end{aligned}, \tag{11}$$

де $Q_{FFW}^{\times}$ – кількість "елементарних" операцій множення, Q_{FFW}^{+} – кількість "елементарних" операцій додавання. При цьому, $Q_{FFW_{ЗГ}}^{+} = 13 \cdot 3^{k-1} + 2^{k+1}(k - 2{,}75)$ – кількість операцій, необхідних для обчислення згортки, $Q_{FFW_{ЗБ}}^{+} = (2^k - 1)$ – кількість операцій в "збірці".

Порівняння обчислювальної складності методів множення багаторозрядних чисел.

Аналіз розглянутих вище методів обчислення добутку багаторозрядних чисел показав,

що кожен з них має свою область ефективного застосування, яка залежить від області значень m (довжин багаторозрядних чисел, що перемножуються) та моделі обчислень. В табл. 1 порівнюється їх обчислювальна складність. В якості показника складності використовується кількість "елементарних" операцій множення $Q^{\times}$, які необхідні для обчислення добутку. В таблиці порівнюються найбільш часто використовувані методи: Карацуби ($Q^{\times}_{KAR}$), з застосуванням ШПУ ($Q^{\times}_{FFW}$) та з застосуванням ШПФ ($Q^{\times}_{FFT}$).

Таблиця 1

Порівняння обчислювальної складності методів множення багаторозрядних чисел

k	$K=2^k$	m (біт)	$Q^{\times}_{KAR}$	$Q^{\times}_{FFW}$	$Q^{\times}_{FFT}$
11	2048	16384	331776	118098	77752
10	1024	8192	82944	39366	34752
9	512	4096	20736	13122	15304
8	256	2048	5184	4374	6608
7	128	1024	1296	1458	2776
6	64	512	324	486	1120
5	32	256	81	162	424

З таблиці видно, що при $m \le 1024$ для обчислення добутку багаторозрядних чисел слід використовувати метод Карацуби або "рекурсивний" метод, оскільки в цьому діапазоні довжин чисел вони мають найменшу складність. При $1024 < m < 8192$ треба застосовувати метод, заснований на ШПУ, а при $m \ge 8192$ – метод, заснований на ШПФ.

ЛІТЕРАТУРА

1. Кнут Д. Искусство программирования. В 3 т. Т.2. Получисленные алгоритмы / Д. Кнут. – М. : Изд. дом "Вильямс", 2001. – 832 с.

2. Задірака В. Комп'ютерна арифметика багаторозрядних чисел / В. Задірака, О. Олексюк. – К. : Наук. видання, 2003. – 246 с.

3. Карацуба А.А. Умножение многозначных чисел на автоматах / А.А. Карацуба, Ю.П. Офман // ДАН СССР. – 1962. – Т. 145. – С. 293-294.

4. Ахо А. Построение и анализ вычислительных алгоритмов : пер. с англ. / А. Ахо, Д. Хопкрофт, Д. Ульман ; под ред. Ю.В. Матиясевича. – М. : Мир, 1979. – 536 с.

5. Тоом А.Л. О сложности схемы из функциональных элементов, реализующей умножение целых чисел / А.Л. Тоом // ДАН СССР. – 1963. – Т. 150. – С. 496-498.

6. Schenhage A. Multiplikation groser Zahlen // Computing. – 1966. – № 1. – P. 182-196.

7. Шенхаге А. Быстрое умножение больших чисел / А. Шенхаге, В. Штрассен // Кибернетический сборник. – 1973. – Вып. 2. – С. 87-98.

8. Болотов А.А. О сложности алгоритмов построения неприводимых трехчленов и пятичленов над конечными полями / А.А. Болотов, С.Б. Гашков, Р.А. Хохлов // Интеллектуальные системы. – 1999. – Т. 4, № 3-4. – С. 12-34.

9. Задирака В.К. Быстрое умножение многоразрядных чисел с использованием БПФ / В.К. Задирака, С.С. Мельникова // Кибернетика и системный анализ. – 1996. – № 3. – С. 63-67.

10. Задирака В.К. Анализ сложности алгоритма умножения сверхбольших чисел на основе коэффициентов Уолша / В.К. Задирака, С.С. Мельникова // Кибернетика и системный анализ. – 2001. – № 6. – С. 99-110.

Надійшла: 01.12.2013р.

Рецензент: д.т.н., професор Козелков С.В.