К.А. Абдикаликов
доктор технических наук, профессор
Актюбинский государственный университет имени К.Жубанова
Актобе/Казахстан

Методы вычисления модулярной арифметики

При реализации криптографических алгоритмов защиты информации основанных на методах RSA - схем или логарифмического обмена, основная вычислительная нагрузка приходится на выполнение модулярных арифметических операций над большими многоразрядными числами. В связи с этим в последнее время значительное внимание уделяется поиску вычислительно эффективных алгоритмов, выполняющих операции модулярной арифметики. Базисной операцией является модулярная арифметика - вычисление остатка от деления одного целого числа на другое.

Многоразрядные (n-словные) целые числа - это числа, требующие для своего размещения n машинных слов $(n > 2)$. В позиционном представлении с основанием B они записываются как

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

где $B = 2^p$ (p - количество разрядов в машинном слове), а W_i - i-я значащая цифра числа W в таком представлении, удовлетворяющая условию $0 \le W_i \le B-1, i = \overline{0, n-1},$ но $W_{n-1} \ne 0$.

Условие криптостойкости криптосистем диктует высокие требования к значению параметра n, которое в практических реализациях достигает нескольких десятков, а то и сотен.

$$f_Z(X) = X^e (\mathrm{mod}\, n) \qquad (2)$$

где X - положительное целое, не превосходящее $n = pq$, «потайной ход» $Z = \{p, q, e\}$, p и q - большие неравные числа, такие что $\varphi(n) = (p-1)(q-1)$ имеет большой простой множитель, а e- положительное целое, не превосходящее $\varphi(n)$, для которого $НОД(e, \varphi(n)) = 1$. Процедура для вычисления (2), помимо своего прямого назначения - шифрования, используется криптосистемами в вероятностных псевдопростых тестах при генерации ключей и должна обладать высоким быстродействием.

Проанализируем ряд алгоритмов для реализации двух последних процедур с точки зрения их быстродействия.

Классический метод. Классический алгоритм - это формализация обычного « деления в столбик », каждый шаг которого является делением $(n+1)$- цифрового числа u на n-цифровое число M и в результате дает одно цифровое частное q и цифровой остаток r. Остаток r меньше M, к нему можно приписать следующую цифру делимого так, что получается $(n+1)$-цифровое число $rB +$ (следующая цифра делимого). Это число используется как u на следующем шаге деления. Замечательным свойством, которое определяет эффективность реализации метода, является то, что достаточно точное приближение к q получается, если делить число, определяемое только первыми двумя значащими цифрами u, на цифру M_{n-1}. Более точно.

Пусть $U=\sum_{i=0}^{n} u_i B^i$, $0<u_n<B$, $0\le x_i\le B$, $i=0,1,\ldots,n-1$, $q=\left\lfloor \frac{u}{M} \right\rfloor$. Положим $\hat{q}=\min\left(\left\lfloor \frac{u_n B+u_{n-1}}{M_{n-1}} \right\rfloor, B-1\right)$. Тогда $\hat{q}\ge q$, и если $M_{n-1}\ge\left\lfloor \frac{B}{2} \right\rfloor$, то $\hat{q}-2\le q$.

При использовании цифр трех старших разрядов u и двух старших разрядов M эту оценку можно сделать почти всегда точной и отличающейся от q не более чем на единицу. Вероятность ошибки примерно равна $^2/_B$.

Классический алгоритм подробно рассмотрен в известной монографии Д. Кнута [1].

В последнее время наибольший интерес проявляется к алгоритмам, основанных на переходе к другому произвольному удобному модулю. Одной из таких популярных вычислительных схем является метод Монтгомери.

Метод Монтгомери. Обсуждаемый алгоритм [2] вычисления остатка использует понятие произведения Монтгомери в системе остаточных классов $(AR)\bmod M$, где R и M - взаимно-простые числа, которое определяется следующим образом:

$$MONTGOMERY\text{ - }PRODUCTION(A,\ B,\ M,R)=(ABR^{-1})\bmod M\ .$$

Произведение Монтгомери в новом представлении системы остаточных классов изоморфно обычному модулярному произведению в обычной системе остаточных классов.

Существуют и другие способы вычисления вычетов по заданному модулю, основанные на переходе к вычислениям по другому модулю. Наиболее известен метод Баретта.

Метод Баретта. Алгоритм [3] основывается на очевидном равенстве

$$(W)\bmod M=W-\left\lfloor((W/B^{n-1})(B^{2n}/M))/B^{n+1}\right\rfloor M\ ,$$

где $\lfloor A\rfloor$ означает ближайшее многоразрядное целое, меньшее A. Поскольку все операнды - многоразрядные целые числа, то в качестве результата операции деления следует брать величину $\lfloor A/B\rfloor$, и алгоритм можно записать в виде последовательности шагов:

шаг 1: вычислить $N=\left\lfloor B^{2n}/M\right\rfloor$;

шаг 2: вычислить $W'=W-W''$, где

$$W''=\left\lfloor(\left\lfloor W/B^{n-1}\right\rfloor\times N)/B^{n+1}\right\rfloor M\ ;$$

шаг 3: найти $(W')\bmod M$.

Сравнительный анализ алгоритмов: Классического, Монтгомери и Барретта. Теоретическое быстродействие разных алгоритмов обычно определяется по количеству операций умножения и деления чисел однократной точности. Это обосновывается тем, что умножение и деление - наиболее время потребляющие операции во внутренних циклах всех трех алгоритмов, временем выполнения других операций по сравнению с умножениями и делениями можно пренебречь.

Все три алгоритма требуют предварительных предвычислений. В классическом алгоритме - это нормализация, в алгоритме Монтгомери - преобразование входных значений переменных и вычисление необходимых дополнительных параметров, в алгоритме Барретта - вычисление параметра B^{2n}/M. Кроме того, метод Монтгомери требует дополнительных поствычислений по результирующим переменным. Как видно, наибольший объем дополнительных вычислений производится в методе Монтгомери. Поэтому его применение оправдывается только в тех случаях, когда такие временные затраты компенсируются большим количеством выполнения модулярной редукции. Для одноразового использования лучше всего пользоваться классическим методом.

Теоретическая сложность трех алгоритмов приведена в табл. 1 [4]. В количество операций здесь не включены умножения и деления, требуемые для предварительных вычислений, поствычислений и преобразований аргумента. Сравнения проводятся на аргументах, которые вдвое длиннее модуля M.

Сложность алгоритмов нахождения остатка Таблица 1

	Алгоритм		
Операция	Классический	Монтгомери	Барретта
Умножение	$n(n+2.5)$	$n(n+1)$	$n(n+4)$
Деление	n	0	0
Предвычисления	Нормализация	$-M^{-1} \bmod B$	B^{2n}/M
Преобразования аргументов	нет	M - вычет	нет
Поствычисления	Денормализация	Редукция Монтгомери	нет
Ограничения	нет	$x < MB^n$, $НОД(x,M)=1$	$x < B^{2n}$

Из таблицы 1 можно сделать следующие выводы. Если рассматривать только чистую операцию нахождения остатка без предварительных и дополнительных вычислений, определяемых выбранным алгоритмом, то наиболее быстрым будет алгоритм Монтгомери. Алгоритм Барретта незначительно опережает классический алгоритм.

В таблице 2 сведены данные сравнения некоторых известных программ, предназначенных для вычисления остатка (вычета). Для этого случая мы также сохраним название коэффициент ускорения, его обозначение К(А,В) (где А – алгоритм вычисления остатка, а В – алгоритм умножения) и метод его оценки.

В столбце 2 алгоритм VI вычисляет частное и остаток по классическому алгоритму D Кнута, а IIIa – умножение по классической прямоугольной схеме. Коэффициенты ускорения для этого столбца вычислены по приведенным в [5] оценкам вычислительной сложности указанных алгоритмов.

Сравнение программ для вычисления остатка Таблица 2

n	K(VI, IIIa)	K(DIL,MD)	K(MOD, MD)	K(K,Б)	K(B,Б)	K(M,Б)
1	2	3	4	5	6	7
8	1.81	1.95	1.73	1.53	1.71	1.120
16	1.82	1.61	1.49	1.38	1.38	1.056
32	1.78	1.38	1.33	1.31	1.21	1.029
48	1.73	1.36	1.31	1.26	1.15	1.027
64	1.71	1.29	1.26	1.25	1.12	1.023

Программы DIL и MOD реализуют схему деления в столбик: DIL вычисляет частное и остаток, а MOD — только остаток. Коэффициенты ускорения, приведенные в графах 5, 6 и 7 взяты из работы [4]. Они вычислены для программ, предназначенных для вычисления остатка соответственно по алгоритмам Кнута, Барретта и Монтгомери.

При применении данных методов для вычисления экспоненциальной модулярной функции метод Монтгомери дает лучшие временные показатели, но для малых аргументов метод Барретта предпочтительнее и поэтому он рекомендуется для выполнения теста Рабина проверки простоты чисел.

Литература:

1. Кнут Д.Е. Искусство программирования для ЭВМ. Т.2. -М.: Мир, 1977. -724 с.
2. Montgomery P.L. Modular Multiplication Without Trial Division // Math. Comp. -1985. -44. -№ 170. -P.519-521.

3. Barrett P.D. Implementing the Rivest Shamir and Adleman Public Key Encryption algorithm on a Standard Digital Signal Processor // Advances in Cryptology. EUROCRYPT `86. Lect. Notes Comput. Sci. -1987. -№ 263.- P.311-323.
4. Bosselaers A., Govaerts R., Vandawalle J. Comparison of Three Modular Reduction Function // Advances in Cryptology. EUROCRYPT ` 94. Lect. Notes Comput. Sci. -1994. -№ 773. -P. 175-186.
5. Задирака В.К., Абдикаликов К.А. Элементы современной криптологии и методы защиты банковской информации. –Алматы.: Гылым, 1999. -336с.

Түйін

Мақалада көпразрядты бүтін сандармен жасалатын операцияларда қолданылатын модулярлы арифметиканы есептеудің тиімді әдістері ұсынылған.

Resume

In the paper we propose effective methods for calculating the modular arithmetic used in operations with multi-digit whole numbers.

Özet

Yazıda çok basamaklı tam sayılar ile operasyonlarda kullanılan modüler aritmetik hesaplamak için etkili yöntemler öneriyoruz.

Sh. N. Kuttykozhaeva
Doctor of physical and mathematical science
Kokshetau State University named after Sh. Ualikhanov
Kokshetau /Kazakhstan

Mathematical modelling of the tasks of hydrodynamics by the method of fictitious areas

The fictitious domain method is one of the known methods of approximate solutions of boundary value problems in mathematical physics. In general fictitious domain method is justified for the linear boundary value problems of mathematical physics. In this work is devoted to substantiation of the fictitious domain method for nonlinear elliptic equations. A new method of obtaining best possible rate of convergence of solutions in the method of fictitious domains.

Let's consider the boundary value problem for nonlinear elliptic equations in $\Omega \subset R^3$ area with border area S

$$\Delta v - v^3 = f, \tag{1}$$

$$v|_S = 0. \tag{2}$$

As method of fictitious domains for the continuation of the lower coefficient in the auxiliary area $D \supset \Omega$ with border $S_1, S_1 \cap S = \emptyset$, solving equation with small parameter

$$\Delta v^\varepsilon - (v^\varepsilon)^3 - \frac{\xi(x)}{\varepsilon} v^\varepsilon = f, \tag{3}$$

$$v^\varepsilon \big|_{S_1} = 0, \tag{4}$$

where f - continued with zero out of Ω and $\xi(x) = \begin{cases} 0, & x \in \Omega, \\ 1, & D_1 = D \setminus \Omega. \end{cases}$

Next one notations used are from the monograph