Известия вузов. Математика
2017, № 11, с. 30–38

http://kpfu.ru/science/nauchnye-izdaniya/ivrm/
e-mail: izvuz.matem@kpfu.ru

Ш.Т. ИШМУХАМЕТОВ, Б.Г. МУБАРАКОВ, КАМАЛЬ МААД АЛЬ-АННИ

ВЫЧИСЛЕНИЕ КОЭФФИЦИЕНТОВ БЕЗУ ДЛЯ k-АРНОГО АЛГОРИТМА НАХОЖДЕНИЯ НОД

Аннотация. Уравнением Безу называется представление наибольшего общего делителя d двух натуральных чисел A и B в виде линейной комбинации $Ax + By = d$, где x, y — целые числа, называемые коэффициентами Безу. Обычно коэффициенты Безу вычисляются с использованием расширенной версии классического алгоритма Евклида.

Разработан новый алгоритм вычисления коэффициентов Безу на основе k-арного алгоритма вычисления наибольшего общего делителя (НОД). Данная задача имеет многочисленные приложения в теории чисел и криптографии, например, для вычисления обратных элементов по умножению в модулярной арифметике.

Ключевые слова: алгоритм Евклида, расширенный алгоритм Евклида, k-арный алгоритм вычисления НОД, вычисление обратных элементов по модулю.

УДК: 511.1

Введение

Классический алгоритм Евклида используется для вычисления НОД двух заданных натуральных чисел A и B. Этот алгоритм работает с использование рекуррентного соотношения $\text{НОД}(A, B) = \text{НОД}(B, A \bmod B)$. Это соотношение используется многократно до тех пор, пока остаток $A \bmod B$ не станет равным нулю, тогда аргумент B будет равен искомому НОД исходной пары (A, B).

Расширенная версия алгоритма Евклида используется для нахождения НОД d и так называемых коэффициентов Безу, т. е. целых чисел u и v таких, что выполнено равенство $Au + Bv = d$, $d = \text{НОД}(A, B)$.

Работа расширенного алгоритма Евклида состоит из двух стадий. На первой стадии вычисляется $d = \text{НОД}(A, B)$, а на второй стадии — коэффициенты Безу u и v по формулам

$$u_n = 0, \quad v_n = 1; \quad u_i = v_{i+1}; \quad v_i = u_{i+1} - v_{i+1} \cdot [A/B]_i, \tag{1}$$

где n — номер последней итерации.

Ниже в таблице 1 приведен пример вычисления коэффициентов Безу для $A = 117$, $B = 41$.

Здесь значение n из (1) равно 5, u_5 и v_5 присваиваем значения 0 и 1, а значения u_i, v_i для $i < n$ вычисляем по формулам (1).

Расширенный алгоритм Евклида может быть использован для вычисления обратных элементов по заданному модулю. Например, для вычисления $a \bmod n$ для взаимно простых

Поступила 24.06.2016

Таблица 1. Нахождение коэффициентов Безу

i	A	B	$A \bmod B$	$[A/B]$	u	v
1	117	41	35	2	-7	20
2	41	35	6	1	6	-7
3	35	6	5	5	-1	6
4	6	5	1	1	1	-1
5	5	1	0	5	0	1

чисел a и n составляем таблицу, подобную таблице 1, полагая $A = n$, $B = a$, тогда значение v_1 будет равно искомому значению обратного элемента. В частности, из приведенного примера получим $41^{-1} \equiv 20 \pmod{117}$.

1. Бинарный и k-арный версии алгоритма Евклида

Бинарный алгоритм вычисления НОД для заданных нечетных чисел A и B использует схему $C = (A - B)/2^r$, где 2^r, $r \geq 1$, — максимальная степень двойки, делящая разность $A - B$.

На следующей итерации алгоритма рассматривается пара (B, C) или (C, B) в зависимости от того, будет ли выполняться неравенство $C < B$.

Общее количество итераций бинарного и классического алгоритмов примерно равно между собой и пропорционально длине меньшего из чисел A, B, однако число элементарных операций на одной итерации в бинарном алгоритме меньше, чем в классическом. Поэтому во многих случаях этот алгоритм работает более эффективно, чем классический.

Расширенная версия бинарного алгоритма для вычисления коэффициентов Безу была разработана М. Пенком и приведена в фундаментальной монографии Р. Крэнделла и К. Померанса ([1], с. 522).

k-арный алгоритм Евклида был разработан Дж. Соренсоном [2], [3] и улучшен независимо друг от друга К. Вебером [4] и Т. Джебелеаном [5]. В основе этого алгоритма лежит теорема, доказанная Дж. Соренсоном.

Теорема. *Пусть целые числа $A > B > 0$ заданы, и k — положительное натуральное число, взаимно простое с A и B. Существуют целые числа x, y, удовлетворяющие соотношению $|x|, |y| \leq \lceil\sqrt{k}\rceil$ такие, что*

$$Ax + By \equiv 0 \mod k, \tag{2}$$

$\lceil\sqrt{k}\rceil$ обозначает ближайшее целое число, полученное округлением вверх.

Полагая $C = (Ax + By)/k$, получим новую пару (B, C), имеющую такой же НОД, что и НОД(A, B), либо НОД, кратный НОД(A, B). Посторонние множители, появляющиеся в НОД(B, C) должны быть изъяты на последней стадии алгоритма путем дополнительного вычисления НОД исходных A и B с НОД, полученным как выход k-арного алгоритма.

Определяя $C = (Ax + By)/k$, должны проверить условие $d = \text{НОД}(C, k) \neq 1$, иначе требуется поделить C на d.

Отметим, что уравнение $Ax + By \equiv 0 \mod k$ эквивалентно уравнению $y \equiv -qx \bmod k$, где $q = AB^{-1} \bmod k$. Искомую пару x, y можно найти перебирая значения x от 1 до $\lceil\sqrt{k}\rceil$ и вычисляя два значения $y \equiv -qx \bmod k$, принадлежащие интервалу $[-\lceil\sqrt{k}\rceil, \lceil\sqrt{k}\rceil]$.

Рассмотрим пример вычисления НОД по методу Дж. Соренсона при $A = 1736704041$, $B = 1210259647$, $k = 64$:

i	A	B	x	y	C
1	1736704041	1210259647	3	-5	13143533
2	1210259647	13143533	5	-7	93113961
3	13143533	93113961	3	-7	2927143
4	93113961	2927143	6	-2	1140733
5	2927143	1140733	5	1	246507
6	1140733	246507	7	-1	30229
7	246507	30229	1	-3	89211
8	30229	89211	1	1	1081
9	89211	1081	1	3	523
10	1081	523	6	-2	85
11	523	85	2	2	19
12	85	19	1	-7	3
13	19	3	1	-1	1

На первой итерации сначала находим $q = AB^{-1} \bmod 64 = 23$. Полагая $x = 1$, вычисляем два значения $y = -23$ и $y = 41$. Оба значения y лежат вне интервала $[-8, 8]$. При $x = 3$ значения y равны -5 и 59. Первое значение принадлежит интервалу $[-8, 8]$, т. е. пара $(x, y) = (3, -5)$ удовлетворяет теореме Дж. Соренсона. Вычисляем $C = (Ax + By)/k = 368370461$. Найденное C взаимно-просто с k, поэтому можно сформировать новую пару $(B, C) = (1210259647, 368370461)$ и перейти к следующей итерации. На последней 13-й итерации найдено $C = 1$, являющееся искомым НОД.

Отметим, что поиск чисел x, y по теореме Дж. Соренсона требует одного вычисления обратного элемента по модулю k и порядка $O(\sqrt{k})$ операций с числами порядка k, поэтому общая сложность этой операции не превысит $O(k \log k)$. Вычисление можно ускорить, если составить предтаблицу обратных по модулю k элементов. Однако при k, сравнимых по размеру с длиной машинного слова, такие таблицы становятся слишком громоздкими.

2. Аппроксимирующий k-арный алгоритм

В [6] и [7] Ш. Ишмухаметовым и Р. Рубцовой был разработан аппроксимирующий k-арный алгоритм, главное отличие которого состоит в том, что коэффициенты x и y ищутся не в интервале $\lceil -\sqrt{k}, \sqrt{k} \rceil$ как в алгоритме Дж. Соренсона, а значение x выбирается из интервала $[1, k]$, а y вычисляется так, чтобы обеспечить наилучшее приближение $Ax \approx -By$. Такой подход обеспечивает взаимное сокращение слагаемых в сумму $Ax + By$ и наименьшее возможное значение $|C| = (Ax + By)/k$, что уменьшает общее число итераций, оставляя общие расходы на вычисления в ходе одной итерации примерно равными классической схеме алгоритма Дж. Соренсона. Рассмотрим работу нашего алгоритма на небольшом примере.

Пусть даны числа $A = 1485$, $B = 793$ и $k = 16$. Выполним одну итерацию алгоритма. Будем искать целые числа x и y, для которых выполнится соотношение (2).

1. Из (2) получим $y \equiv -AB^{-1}x \bmod k = -5x \bmod 16$, откуда $y = -5x + sk$, $s \in \mathbf{Z}$.

2. Обозначим через r дробь A/B. Выполним следующие преобразования:

$$C = \left| \frac{Ax + By}{k} \right| = B \left| \frac{rx + (-qx + ks)}{k} \right| = B \left| \frac{(r - q)x}{k} + s \right|.$$

3. Представим дробь $(r-q)/k$ в виде суммы правильной дроби и целого числа

$$\frac{r-q}{k} = \frac{u}{v} + s_1, \quad \frac{u}{v} \in [0,1), \quad s_1 \in \mathbf{Z}.$$

Тогда последнее равенство можно переписать в виде

$$C = B\left|\frac{(r-q)x}{k} + s\right| = B\left|\frac{ux}{v} + s_1 x + s\right|. \tag{3}$$

4. Используя дроби Фарея [8], [9] найдем правильную дробь m/n, $0 < m < n < k$, аппроксимирующую u/v. В [7] доказано, что можно найти такие m и n, что выполнится соотношение

$$|\varepsilon| = \left|\frac{u}{v} - \frac{m}{n}\right| < \frac{3}{2k(k-1)} = O(k^{-2}).$$

Пара (m,n) может быть найдена за $O(\log_2 k)$ шагов. Фактически дробь m/n является последней из подходящих непрерывных дробей со знаменателем, ограниченным величиной $k-1$ ([10], раздел 2.6). Используя (3), получим

$$C = B\left|\frac{u}{v}x + s_1 x + s\right| = B\left|\frac{m}{n}x + \varepsilon x + s_1 x + s\right|.$$

Определим значение x, равным знаменателю n дроби m/n, а целое число s возьмем равным $s = -m - s_1 n$. Тогда

$$C = B\left|\frac{m}{n}x + \varepsilon x + s_1 x + s\right| = B\left|m + \varepsilon n + s_1 n + (-m - s_1 n)\right| = B|\varepsilon n| < \frac{3B}{2k}.$$

Для ускорения вычислений вместо исходной дроби $r = A/B$ берется $r' = A'/B'$, где A', B' получены отсеканием справа от A и B некоторого числа разрядов так, чтобы $|r - r'| < 1/2k$. В нашем примере $q = AB^{-1} \bmod k = 5$:

$$r = \frac{A}{B} = \frac{1485}{793} \approx \frac{185}{99} \quad \text{и} \quad \frac{r'-q}{k} = \frac{185/99 - 5}{16} = \frac{637}{792} - 1 \approx 0.804 - 1.$$

Наилучшей подходящей дробью для $\alpha = 0.804$ является $4/5$. Определим теперь x равным знаменателю этой дроби $x = 5$. Вычислим $s = -m - s_1 n = -4 + 5 = 1$ и $y = -qx + sk = -5 \cdot 5 + 16 = -9$. Теперь можно вычислить

$$C = \left|\frac{Ax + By}{k}\right| = \left|\frac{1485 \cdot 5 - 793 \cdot 9}{16}\right| = 18.$$

После сокращения на двойку получим $C = 9$, что дает уменьшение второго множества в $793/9 \approx 88$ раз. Приведем пример вычислений по нашему алгоритму для 33-битовых чисел A и B и $k = 64$ из предыдущего раздела:

	A	B	A/B	x	y	$C = (xA + yB)/2^r k$	$\rho = A/C$
1	1736704041	1210259647	1.4	3	-5	13143533	132
2	1210259647	13143533	92	58	-5342	276465	4378
3	13143533	276465	48	19	-903	619	21233
4	276465	619	447	22	-9826	1	276465

Вычисление НОД таких чисел выполняется по нашему алгоритму всего за четыре итерации. Сравнивая две таблицы вычислений для нашей схемы вычислений и схемы Дж. Соренсона, видим, что в случае когда отношение A/B близко к единице, результат в обоих схемах получился одинаковым, однако, по мере увеличения A/B, наша схема дает существенное более быструю сходимость алгоритма.

Ниже рассмотрим соответствующую схему вычисления коэффициентов Безу. Наш алгоритм позволит многократно ускорить вычисления в криптографических алгоритмах, использующие модулярные вычисления. Примеры криптографических вычислений можно найти в [11].

Отметим интересный подход к задаче аппроксимации действительного числа дробями с ограниченными знаменателями в работе Ванга и Пана [12].

3. Нахождение коэффициентов Безу в случае k-арного алгоритма

Будем искать для k-арного алгоритма формулы, аналогичные (1). Рассмотрим сначала простой случай, когда НОД$(C_i, k) = 1$ и $C_i < B_i$, где $C_i = (x_iA_i + y_iB_i)/k$. В этом случае новая пара (A_{i+1}, B_{i+1}) равна $(B_i, (x_iA_i + y_iB_i)/k)$.

3.1. **Простой случай вычисления новой пары** (A, B). Пусть для пары (A_{i+1}, B_{i+1}) уже найдены искомые u_{i+1} и v_{i+1}: $u_{i+1}A_{i+1}+v_{i+1}B_{i+1} = d$. Подставим в это уравнение $A_{i+1} = B_i$, $B_{i+1} = (x_iA_i + y_iB_i)/k$:

$$u_{i+1}B_i + v_{i+1}(x_iA_i + y_iB_i)/k = d, \quad \text{откуда} \quad (v_{i+1}x_i/k)A_i + (u_{i+1} + v_{i+1} \cdot y_i/k)B_i = d.$$

Получим формулы для нахождения коэффициентов u, v, аналогичные формулам (1):

$$u_n = 0, \quad v_n = 1; \quad u_i = v_{i+1}x_i/k; \quad v_i = u_{i+1} + v_{i+1}y_i/k, \quad 1 \le i < n. \tag{4}$$

В следующей таблице выполнены вычисления коэффициентов u, v для нашего примера:

i	A	B	x	y	C	u	v
1	117	41	1	3	15	$28/k^3$	$20/k^3$
2	41	15	2	2	7	$-4/k^2$	$28/k^2$
3	15	7	2	-2	1	2/k	$-2/k$
4	7	1	-	-	-	0	1

Опять обратный ход начинается с присвоения коэффициентам последней строки значений 0 и 1. Потом поднимаемся вверх последовательно применяя формулы (4) для значений u_i и v_1 для $i = n-1, n-2, \ldots, 1$. Можно заметить, что коэффициенты u_i, v_i являются дробями со знаменателем k^{n-i}, где n — номер последней строки. Обозначим через p_i и q_i числители дробей u_i и v_i. Эти значения связаны формулами $p_i = k^{n-i}u_i$, $q_i = k^{n-i}v_i$. Тогда формулы (4) перепишутся так:

$$p_i = x_iq_{i+1}, \quad q_i = kp_{i+1} + y_iq_{i+1}. \tag{5}$$

3.2. **Вычисление обратных элементов по модулю заданного числа.** Формулы (4) дают нам дробные значения, а коэффициенты Безу по определению являются целыми числами. Значит, необходимо досчитать эти коэффициенты, заменяя значения в формулах (4) на эквивалентные по модулю целые значения. Для этого необходимо знать значения $k^{-1} \bmod A$.

Обозначим $a = A \bmod k$. Если брать k равным степени двойки $k = 2^s$, как рекомендует Т. Джебелеан [5], а значения чисел A и B представить в двоичной системе исчисления, то

a равно числу, расположенному в младших s разрядах двоичного представления A, поэтому вычисление a требует s тактов. Таблицу обратных элементов по модулю $k = 2^s$ можно составить заранее. Ее полное вычисление выполнится за $O(k \log_2 k)$ шагов. Для своей работы и хранения результатов алгоритм требует $O(k)$ ячеек памяти, что также критично при больших значениях k.

Покажем далее как вычислить $k^{-1} \bmod A$. Обозначим это число через x. По определению $kx \bmod A = 1$. Отсюда найдется такое целое p, что

$$kx = 1 + pA. \tag{6}$$

Вычисляя остаток по модулю k от обоих частей (6), получим $pA \equiv -1 \pmod{k}$ или $p \equiv -A^{-1} \pmod{k}$, откуда $p \equiv -a^{-1} \pmod{k}$, $p = -a^{-1} + rk$, $r \in \mathbf{Z}$. Подставляя значение p в (4), получим

$$x = \frac{1 + (-a^{-1} + rk)A}{k} = rA + \frac{1 - a^{-1}A}{k},$$

откуда

$$k^{-1} \bmod A = x \equiv \frac{1 - a^{-1}A}{k} \pmod{A}$$

(в правой части выполняется простое деление числителя на знаменатель). Теперь можно пересчитать все коэффициенты u_i, v_i, заменяя деление на k^{n-i} умножением на $k^{i-n} \bmod A$. Отметим полученные значения штрихами:

$$u'_i = k^{i-n} p_i \bmod A, \quad v'_i = k^{i-n} q_i \bmod A, \quad 1 \le i \le n.$$

При $i = 1$ получим уравнение

$$u'_1 A + v'_1 B \equiv d \pmod{A}, \quad \text{откуда} \quad u'_1 A + v'_1 B = d + tA, \quad t \in \mathbf{Z}.$$

Из последнего уравнения получим требуемое выражение для коэффициентов Безу:

$$uA + vB = \gcd(A, B), \quad \text{где} \quad u = u'_1 - t, \quad v = v'_1, \quad u, v \in \mathbf{Z}.$$

3.3. **Пример вычисления обратных элементов.** Выполним вычисление коэффициентов u, v по формулам (4) и (5) для приведенного выше примера:

$$a^{-1} \bmod k = 117^{-1} \bmod 16 = 5^{-1} \bmod 16 = -3 \equiv 13 \pmod{16},$$

$$k^{-1} \bmod A = 16^{-1} \bmod 117 = (1 - a^{-1}A)/k = (1 + 3A)/k = (1 + 3 \cdot 117)/16 = 22,$$

$$k^{-2} \bmod A = 22^2 \bmod 117 = 16, \; k^{-3} \bmod A = 22^3 \bmod 117 = 1.$$

Вычислим коэффициенты u_i, v_i, умножая на $k^{i-n} \bmod A$:

i	A	B	x	y	C	u	v	u'	v'
1	117	41	1	3	15	$28/k^3$	$20/k^3$	28	20
2	41	15	2	2	7	$-4/k^2$	$28/k^2$	-64	97
3	15	7	2	-2	1	2/k	-2/k	44	-44
4	7	1	-	-	-	0	1	0	1

Вычислим выражение

$$u'_1 A + v'_1 B = 28 \cdot 117 + 20 \cdot 41 = 4096 = 35 \cdot 117 + 1, \quad \text{откуда} \quad -7A + 20B = 1.$$

Отметим, что нет необходимости вычисления коэффициентов u', v' на промежуточных итерациях. Достаточно, вычислить их значение для изначальной пары (A, B) при $i = 1$.

3.4. **Другие варианты значения C.** Рассмотрим более сложные варианты перехода к новой паре (A, B). Предположим сначала, что $(xA + yB)/k$ имеет общие делители с k. Для случая $k = 2^s$ это означает, что $C_{i+1} = (x_iA_i + y_iB_i)/k$ четно. В этом случае должны делить C_{i+1} на эти делители, пока НОД(k, C_{i+1}) не станет равным единице. Тогда новое значение

$$C_{i+1} = (A_ix_i + B_iy_i)/2^{r_i}, \quad \text{где} \quad 2^{r_i} \geq k. \tag{7}$$

Пересчитаем формулы (5), учитывая (7):

$$p_i = x_iq_{i+1}, \quad q_i = 2^{r_i}p_{i+1} + y_iq_{i+1}. \tag{8}$$

Нужные коэффициенты u'_1 и v'_1 можно получить, выполняя вычисление обратных по модулю A элементов: $u'_1 = 2^{-r}p_1 \bmod A$, $v'_1 = 2^{-r}q_1 \bmod A$, где $r = \sum\limits_{i<n} r_i$.

Последним рассмотрим случай $C_{i+1} > B_i$. В этом случае операнды p_{i+1}, q_{i+1} в формулах (8) поменяются местами:

$$p_i = x_ip_{i+1}, \quad q_i = 2^{r_i}q_{i+1} + y_ip_{i+1}.$$

Заметим, что этот случай возможен только для алгоритма Дж. Соренсона, поскольку в аппроксимирующем алгоритме C_{i+1} всегда меньше B_i.

4. Окончательный вариант алгоритма вычисления уравнения Безу

Даны два нечетных числа A и B, $1 < B < A$, и $k = 2^s$. Требуется решить уравнение $xA + yB = d$, где $d =$ НОД(A, B).

1. Полагаем $A_1 = A$, $B_1 = B$. Выполняем прямой ход k-арного алгоритма, вычисляя $C_i = (x_iA_i + y_iB_i)/2^{r_i}$, $r_i \geq s$, и сохраняя значения x_i, y_i. Накапливаем сумму $r_1 + r_2 + \cdots + r_i$.

2. Вычисление выполняется до тех пор, пока на каком-нибудь шаге i не получим $C_i < k$. Закончим прямое вычисление, полагая $n = i + 1$, $A_n = B_i$, $B_n = C_i$. Определим $d_1 = B_n$ — текущее значение НОД(A, B). Если $d_1 = 1$, то переходим к шагу 4. Возможно d_1 содержит посторонние множители. Проверим их наличие на следующем шаге.

3. Вычислим $d_2 =$ НОД(A_1, d), используя классическую схему Евклида. Если $d_2 = 1$, переходим к шагу 4. Иначе вычисляем $d_3 =$ НОД(B_1, d_2). Если $d_3 > 1$, то определим $d =$ НОД(A, B) равным d_3. Во всех остальных случаях определим $d = 1$.

4. Переходим к обратному ходу. Пусть n — номер последней итерации. Полагаем $p_n = 0$, $q_n = 1$. Выполняем последовательное вычисление p_1, q_i по формулам (8).

5. Пусть $r = r_1 + r_2 + \cdots + r_n$. Получено уравнение $pA + qB = d_1 \cdot 2^r$ или $pA + qB = d \cdot w2^r$, где w — посторонний множитель.

6. Вычислим $D = (w2^r)^{-1} \bmod A$.

7. Вернемся к основному уравнению $uA + vB = d$. Коэффициент v найдем из сравнения $v \equiv qD \bmod A$. Коэффициент при u найдем, подставляя в основное уравнение v: $u = (d - vB)/A$. Уравнение Безу решено.

Пример. Найти решение уравнения Безу для $A = 7619$, $B = 2795$. Прямой ход алгоритма выполним по классической схеме k-арного алгоритма при $k = 16$:

i	A	B	x	y	r	C	p	q
1	7619	2795	2	-2	4	603	38	-86
2	2795	603	1	-1	4	137	-3	19
3	603	137	1	-3	6	3	1	-3
4	137	3	-	-	-	-	0	1

В ходе вычисления появляется посторонний множитель $w = 3$, который может быть найден путем вычисления $w = d_2/d_1 = 3/\text{НОД}(7619, 3) = 3$.

Сумма всех r_i равна $s = 4 + 4 + 6 = 14$, откуда получим уравнение $38A - 86B = 3 \times 2^{14}$. Необходимо сократить обе части уравнения на множитель $3 \cdot 2^{14}$. Вычислим $D = (3 \times 2^{14})^{-1} \bmod A = 5281$. Теперь можно найти коэффициенты u и v:

$$v = qD \bmod A = (-86 \cdot 5281) \bmod 7619 = -4645,$$
$$u = (d - vB)/A = (1 + 4645 \cdot 2795)/7619 = 1704.$$

Значит наше уравнение имеет вид $1704A - 4645B = 1$. Если в качестве v взять положительное решение сравнения $v \equiv qD \bmod A$, а именно $v = 2974$, то соответствующее $u = -1091$ дает другое решение уравнения Безу

$$-1009A + 2974B = 1.$$

5. Заключение

Нами разобрана схема расширенного k-арного алгоритма, позволяющая вычислять коэффициенты уравнения Безу, а значит, вычислять обратные по модулю элементы быстрее, чем в классической схеме алгоритма Евклида. Общая асимптотическая оценка данного алгоритма остается той же, т. е. равна $O(n^2)$, однако практический алгоритм при k, не превышающем размера машинного слова, работает значительно быстрее своего классического аналога.

Литература

[1] Крэнделл Р., Померанс К. *Простые числа: криптографические и вычислительные аспекты* (УРРС, М., 2011).
[2] Sorenson J. *The k-ary GCD algorithm*, University of Wisconsin-Madison, Techn. Report, 1–20 (1990).
[3] Sorenson J. *Two fast GCD algorithms*, J. Algorithms **16** (1), 110–144 (1994).
[4] Weber K. *The accelerated integer GCD algorithm*, ACM Trans. Math. Software **21** (1), 1–12 (1995).
[5] Jebelean T. *A generalization of the binary GCD algorithm*, Proc. of Intern. Symp. on Symb. and Algebr. Comp. (ISSAC'93), 111–116 (1993).
[6] Ishmukhametov S.T., Rubtsova R.G. *A fast algorithm for counting GCD of natural numbers*, Proc. Intern. conf. Algebra, Analysis and Geometry, Kazan, KFU (2016), 52–53.
[7] Ishmukhametov S.T. *An approximating k-ary GCD algorithm*, Lobachevskii J. Math. **37** (6), 722–728 (2016).
[8] Hardy G.H., Wright E.M. *An introduction to the theory of numbers*. 4th Ed. (Oxford, Clarendon Press, 1959).
[9] Грэхем Р., Кнут Д., Паташник О. *Конкретная математика. Математические основы информатики.* 2-е испр. изд. (Вильямс, 2015).
[10] Ишмухаметов Ш.Т. *Методы факторизации* (КФУ, Казань, 2011).
[11] Ишмухаметов Ш.Т., Рубцова Р.Г. *Математические методы защиты информации*, электронное учебное пособие (КФУ, Казань, 2012) http://kpfu.ru/docs/F366166681/mzi.pdf
[12] Wang X., Pan V. *Acceleration of Euclidian algorithm and rational nimber reconstruction*, Siam J. Comp. **32** (2), 548–556 (2003).

Ш.Т. Ишмухаметов
Казанский федеральный университет,
ул. Кремлевская, д. 18, Казань, 420008, Россия,
e-mail: Shamil.Ishmukhametov@kpfu.ru

Б.Г. Мубараков
Казанский федеральный университет,
ул. Кремлевская, д. 18, Казань, 420008, Россия,
e-mail: mubbulat@mail.ru

Камаль Маад Аль-Анни

Университет Страсбурга,
ул. Блеза Паскаля, д. 4, г. Страсбург, 67081, Франция,

e-mail: maadk_anni@live.com

Sh.T. Ishmukhametov, B.G. Mubarakov, and Kamal Maad Al-Anni

Calculation of Bezout's coefficients for k-ary algorithm of greatest common divisor

Abstract. Bezout's equation is a representation of the greatest common divisor d of two integers A and B as a linear combination $Ax+By=d$, where x and y are integers called Bezout's coefficients. Usually Bezout's coefficients are caclulated using the extended version of the classical Euclidian algorithm.

We elaborate a new algorithm for calculating Bezout's coefficients based on the k-ary GCD algorithm. This problem has numerous applications in the number theory and cryptography, for example, for calculation of multiplicative inverse elements in modular arithmetic.

Keywords: Euclidian algorithm, extended Euclidian algorithm, k-ary GCD algorithm, calculation of inverse elements by module.

Sh.T. Ishmukhametov

Kazan Federal University,
18 Kremlyovskaya str., Kazan, 420008 Russia,

e-mail: Shamil.Ishmukhametov@kpfu.ru

B.G. Mubarakov

Kazan Federal University,
18 Kremlyovskaya str., Kazan, 420008 Russia,

e-mail: mubbulat@mail.ru

Kamal Maad Al-Anni

University of Strasbourg,
4 Rue Blaise Pascal, Strasbourg, 67081 France,

e-mail: maadk_anni@live.com