УДК 519.688: 004.75:004.421

І. О. Процько[1], О. В. Грищук[2]

РОЗПАРАЛЕЛЕННЯ ОБЧИСЛЕННЯ КАНОНІЧНОГО РОЗКЛАДУ ЧИСЛА НА МНОЖНИКИ

[1]Національний університет “Львівська політехніка”, [2]ТзОВ “Логіка”

Анотація. В статті розглянуто обчислення канонічного розкладу числа на множники з використанням модифікованого методу пробних ділень. Виконання операцій ділення числа розкладу на прості числа для перевірки на кратність вимагає відповідних часових затрат в сучасних комп’ютерних системах. Для їх зменшення використовується бінарне подання числа розкладу в процесі його аналізу на кратність. Для кожного розряду бінарного числа розкладу, що дорівнює одиниці, визначаються залишки його вагового коефіцієнта за модулем відповідного простого числа. Отримані значення залишків акумулюються і потім виконується перевірка накопленого значення на рівність з відповідним значенням з множини простих чисел. У випадку рівності отримуємо елемент канонічного розкладу і знову перевіряємо степені цього елемента розкладу на кратність. В протилежному випадку переходимо на наступне більше просте число для подальшої перевірки на кратність, обмежуючись значенням кореня квадратного числа розкладу. Незалежність підзавдань виконання перевірки бінарного подання числа на подільність простими числами дає можливість розпаралелювати виконання розкладу числа в багатоядерних мыкропроцесорах комп’ютерних систем. Серед рівнів паралельності можна послідовно виділити: визначення залишків вагових коефіцієнтів, акумулювання залишків для одиничних розрядів бінарного подання числа розкладу, перевірки на кратність сукупністю простих чисел. Паралельне обчислення розкладу числа досягається виконання алгоритму в багатьох потоках. Програмна реалізація на мові С++, відповідно алгоритму, розподіляє обчислення між потоками, використовуючи пул потоків. В алгоритмі розпаралелення обчислень канонічного розкладу, в залежності від введеного значення числа розкладу, визначається відповідне значення кількості простих чисел та їхніх степенів і рівномірно розподіляється між потоками для виконання аналізу на подільність. В результаті визначено залежність часу обчислення канонічного розкладу числа від кількості потоків в багатоядерних мікропроцесорах лінійки Intel Core i3/i5/i7. Для кожної комп’ютерної системи, що має певну кількість обчислювальних ядер в мікропроцесорах, існує оптимальна кількість потоків, яка забезпечує мінімальний час канонічного розкладу числа на множники.

Ключові слова: канонічний розклад, прості множники, залишки, вагові коефіцієнти, потоки, паралельне обчислення.

Аннотация. В статье рассмотрено вычисление канонического разложения числа на множители с использованием модифицированого метода пробных делений. Выполнение операций деления числа разложения на простые числа для проверки на кратность требует соответствующих временных потерь в современных компьютерных системах. Для их уменьшения используется бинарное представление числа разложения в процессе его анализа на кратность. Для каждого разряда бинарного числа разложения, что равняется единице, определяются остатки его весового коэффициента по модулю соответствующего простого числа. Полученные значения остатков аккумулируются и потом выполняется проверка накопленого значения на равенство с соответстующим значением из множества простых чисел. В случае равенства получаем элемент канонического разложения и снова проверяем степени єтого єлемента розложения на кратность. В противном случае переходим на следующее большее простое число для дальнейшей проверки на кратность, ограничиваясь значеннием квадратного кореня числа разложения. Независимость подзадач выполнения проверки бинарного представления числа на делимость простыми числами дает возможность распараллеливать выполнение разложения числа в многоядерных микропроцессорах компьютерных систем. Среди уровней параллельности можно последовательно выделить: определение остатков весовых коэффициентов, аккумулирование остатков для единичных разрядов бинарного представления числа разложения, проверки на кратность наборам из простых чисел. Программная реализация на С++, соответственно алгоритму, распределяет вычисления во многих потоках, используя пул потоков. В алгоритме распараллеливания вычислений канонического разложения, в зависимости от введеного значения числа разложения, определяется соответствующее значение количества простых чисел с их степенями и равномерно распределяется между потоками для выполнения анализа на делимость. В результате определена зависимость времени вычисления канонического разложения числа от количества потоков в многоядерных микропроцессорах линейки Intel Core i3/i5/i7. Для каждой компьютерной системы, которая имеет определенное количество вычислительных ядер в микропроцессорах, существует оптимальное количество потоков, которое обеспечивает минимальное время канонического разложения числа на множители.

Ключевые слова: каноническое разложение, простые множители, остатки, весовой коэффициент, потоки, параллельное вычисление.

Abstract. The computation of the canonical factorization of a number using the modified trial divisions method has been considered. The performing operations of the division a number of factorization into prime numbers for the testing on a repetition factor demands a respective loss of the execute time in modern computer systems. To reduce them, the presentation of the number of factorization in the binary form is used for the process of analysis on repetition factors. The residuals of weighting coefficient are defined for each digit of the binary representation the number of factorization, which is equal to one. The obtained values of the residuals are accumulated and then the accumulated value is checked for the equality with the corresponding value from the set of prime numbers. In case of equality, we obtain an element of canonical factorization and again check the degrees of this element for a repetition factor. Otherwise, we proceed to the next larger prime number for further checking for a repetition factor. The independence of the subtasks to perform the check of the binary representation of a number on divisibility by prime numbers makes it possible to parallelize the execution of the factorization of a number in multi-core microprocessors of computer systems. Among the levels of the parallelism can be consistently identified: the definition of residual weighting coefficients, the accumulation of residuals for bits equel to one of the binary representation of the number of factorization, the checking for a repetition factor from the sets of primes. Software implementation in C++, according to the algorithm, schedules the computations in multithreads, using a pool of threads. In the algorithm for parallelizing the computations of the canonical factorization, depending on the entered value of the expansion number, the corresponding value of the number of primes with their powers is determined and is evenly distributed between the streams to perform an analysis of divisibility. As a result, the dependence of the run time the computation of the factorization of a number from the number of threads is defined in multi-core processors of Intel Core i3/i5/i7. For the each computer system exist the optimal number of the threads, which supports the minimal time of the canonical factorization of a number on the prime numbers.

Key words: canonical factorization, prime factors, residual, weighting factor, threads, parallel computation.
DOI: https://doi.org/10.31649/1999-9941-2019-44-1-46-51.

Вступ

Сучасні багатоядерні системи обчислювальної техніки нарощують свою продуктивність завдяки виконанню паралельних обчислень. Особливо, якщо така можливість закладена в алгоритмі та реалізована новітніми технологіями паралельного програмування.

Канонічний розклад числа, як фундаментальна теорема арифметики про однозначне представлення цілого числа [1], турбував вчених ще до нашої ери і повязаний з іменами Евкліда, Ератосфена Киренського. Канонічний розклад числа N на прості множники подано у вигляді

$$N = p_1^{s1}\, p_2^{s2} \ldots p_k^{sk}, \quad (1)$$

де $p_1^{s1}, p_2^{s2}, \ldots p_k^{sk}$ прості множники $(p_1^{s1} < p_2^{s2} < \cdots < p_k^{sk})$ та $s1, s2, \ldots, sk$ степінь їх повторюваності. В наш час задача підвищення методів та алгоритмів пошуку розкладу числа на прості множники або перевірки простоти цілих чисел вийшла на кардинально новий рівень. Це зобумовлено потребою побудови високостійких криптосистем, в яких використовуються "випадкові" прості числа з великого числа цифр [2].

Починаючи з 1977 року, коли було опубліковано в журналі «Scientific American» новий алгоритм шифрування [3] і було дано список з 42 тестових чисел різної розрядності й за розклад кожного з цих чисел була призначена премія, розроблено класичні алгоритми канонічного розкладу. До яких відносять ρ – метод, (p−1) – метод Полларда, метод еліптичних кривих, метод квадратичного решета, метод решета числового поля й інші [4]. На основі цих методів розробляються програмні модулі та модифікації класичних алгоритмів для рішення різноманітних прикладних завдань. Дослідження методів канонічного розкладу в напрямку розпаралелювання виконання обчислень показує, що основна ідея полягає у виборі випадкового значення для кожного окремого паралельного процесу, кождний з яких буде виконувати алгоритм відповідно методу. Однак одночасне виконання алгоритму декількома значними за складністю потоками для знаходження множників числа розкладу на завжди дає очікуваний результат [5].

Найростішим рішенням канонічного розкладу є метод пробних ділень. Для розкладу числа на множники метод використовує елементарної перевірки на ділення без залишку даного обсягу N на послідовний вибір дільників з множини простих чисел 2,3,5,7,9,11,13,17,…. Однак послідовне виконання операцій ділення для перевірки на кратність в сучасних комп'ютерних моделях виконується за допомогою мікропрограми або спеціалізованих арифметичних вузлів, що збільшує тривалість обчислення канонічного розкладу. Адже операція ділення реалізується з послідовного набору арифметичних операцій додавання/віднімання, незалежно від того, який з алгоритмів використовується. Для перевірки простоти числа з великою кількістю цифр метод пробних ділень з обчислювальної точки зору є достатньо ємкий.

Актуальність

Забезпечення високої швидкодії обчислення комп'ютерними системами класичної задачі факторизації цілочисельного значення на прості множники вимагає розробки ефективних алгоритмічних методів з використанням новітніх інформаційних технологій. Швидке обчислення факторизації чисел для забезпечення високої криптостійкості інформаційних даних, для переходу до багатовимірного подання одновимірних послідовностей інформаційних даних та інших застосувань є достатньо затребуваним в багатьох практичних завданнях.

Мета

Метою роботи є підвищення швидкодії обчислення канонічного розкладу числа на множники на основі вдосконаленого методу пробних ділень та розпаралелення виконання розкладу багатоядерними процесорами сучасних комп'ютерних систем. Виконання канонічного розкладу в обчислювальному середовищі на програмному або апаратному рівнях елементарної операції накопичення значень залишків значно ефективніше з точки зору їхніх величин і в порівнянні з виконанням операції ділення. Можливість подальшого підвищення швидкодії обчислення розкладу числа полягає в паралельному апаратному виконанні методу пробних ділень в аспекті перевірки накоплених значень залишків на рівність простим числам та їхнім степеням.

Задачі

1. Формулювання аналітичної моделі канонічного розкладу на основі вагових залишків бінарного числа.

2. Аналіз обчислювальної складності $O(f(n))$ визначення канонічного розкладу числа на основі вагових залишків.

3. Розробка програмної реалізації шляхом розподілення обчислення канонічного розкладу числа в багатьох потоках.

4. Тестування залежності часу виконання обчислення канонічного розкладу числа на множники від кількості потоків.

Підхід канонічного розкладу на основі вагових залишків бінарного числа

Розглянемо підхід пробних ділень з використанням залишків для кожного вагового коефіцієнта бінарного представлення числа, що визначаються операцією за модулем від простих чисел p_i та їх можливих повторень s_i [6]. Визначення подільності десяткового числа N, коли залишок дорівнює нулю, у випадку представлення в двійковій системі числення матиме вигляд:

$$N \bmod i = (a_n 2^n + a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \cdots a_1 2 + a_0) \bmod i = (a_n(2^n \bmod i) + a_{n-1}(2^{n-1} \bmod i) - a_{n-2}(2^{n-2} \bmod i) + \cdots a_1(2 \bmod i) + a_0) \bmod i, \quad (2)$$

де a_i – двійкові розряди числа, i – просте число.

Розклад числа N за формулою (2) базується на використанні залишків кожного вагового коефіцієнта ($2^k \bmod i$), де k=0,1,…n кількість двійкових розрядів числа. Ці залишки, що відображені в таблиці 1, значно ефективніше в порівнянні з арифметичною операцією ділення, запам'ятовується або визначаються [6] в обчислювальному середовищі. Пробна перевірка подільності числа виконується для послідовності простих чисел та їх степенів P = {2,3,5,7,9,11,13,17,…,i } загальною кількістю m.

Виконавши накопичення значень залишків за вибраними вагами, порівнюємо накопичену суму з простим числом i або його степенем i^s. У випадку порівняння, коли накопичене значення залишків більше – знову проводиться за формулою (2) накопичення залишків від попереднього одержаного накопиченого числа, а у випадку рівності – виводиться елемент канонічного розкладу i та виконується перехід до наступного значення з послідовності простих чисел P{i}. Отже, в результаті отримуємо набір простих множників канонічного розкладу (1) числа N.

Обчислювальна складність $O(f(n))$ визначення канонічного розкладу на основі вагових залишків бінарного числа в загальному випадку визначається кількістю операцій, які необхідно виконати. Наближену оцінку обчислювальної складності (3) для найскладнішого випадку N можна визначити, взявши на основі таблиці значень залишки для всіх n вагових коефіцієнтів з послідовності $m = N/2$ простих чисел та їх степенів.

Таблиця 1 – Значення залишків двійкового розряду числа за модулями простих чисел та їх степенів

m/*n*	*n*	…	7	6	5	4	3	2	1	0	P{*i*}
1	$2^n \bmod 3$	…	2	1	2	1	2	1	2	1	*mod* 3
2	$2^n \bmod 5$	…	3	4	2	1	3	4	2	1	*mod* 5
3	$2^n \bmod 7$	…	2	1	4	2	1	4	2	1	*mod* 7
4	$2^n \bmod 9$	…	2	1	5	7	8	4	2	1	*mod* 9
5	$2^n \bmod 11$	…	7	9	10	5	8	4	2	1	*mod* 11
6	$2^n \bmod 13$	…	11	12	6	3	8	4	2	1	*mod* 13
…	…	…	…	…	…	…	…	…	…	…	…
m	$2^n \bmod i$	…	$2^7 \bmod i$	…	…	…	…	…	$2^1 \bmod i$	$2^0 \bmod i$	*mod i*

У цьому випадку будемо мати (n*N/2) – максимальну кількість додавань та (N/2) – мінімальну кількість порівнянь одержаної суми накопичення з простим числом i або його степенем i^s. Для накопичення залишків з мінімальною кількістю порівнянь врахуємо множник їх можливого повторення в межах k = (1,5 ÷ 3).

$$O(k(n*N/2 * N/2)) = O(N^2 * n * k) = O((p_1^{s1} p_2^{s2} \ldots p_k^{sk})^2 n k). \quad (3)$$

Отже, обчислювальна складність канонічного розкладу на основі вагових залишків бінарного числа має квадратичну залежність від значення числа розкладу $N = p_1^{s1} p_2^{s2} \ldots p_k^{sk}$ та лінійну від кількості його двійкових розрядів n. Тому розклад великих чисел на основі даного підходу ефективно може бути реалізований з використанням паралельного аналізу на подільність в багатоядерних комп'ютерних системах.

Виконання паралельного алгоритму канонічного розкладу

На сьогодні розпаралелення обчислювальних процесів вийшло на кардинально новий рівень з точки зору підвищення продуктивності. Ефективність розробки паралельних програм багато в чому залежить не тільки від паралельності алгоритму, але від наявності відповідного програмного та апаратного інструментарію.

Практично всі сучасні ОС підтримують керування потоками. Прикладна програма реалізує керування потоками за допомогою спеціальних бібліотек, що дозволяють досягнути апаратного прискорення канонічного розкладу числа. Обчислення канонічного розкладу числа на множники з використанням залишків для кожного вагового коефіцієнта в бінарному представленні числа розроблена в IDE Visual C++ 2017, але може бути скомпільована для іншого середовища.

Програма виконує наступні дії:

1. Введення числа для розкладу на множники та введення кількості потоків для паралельного обчислення.
2. Побудова таблиці залишків від ділення на прості числа.
3. Вивід на екран канонічних множників розкладу для введеного числа.
4. Тестування швидкодії виконання розкладу чисел на прості множники з виводом часу обчислення.

Паралельне обчислення розкладу числа досягається шляхом розподілення алгоритму роботи в багатьох потоках (threads). В залежності від кількості потоків визначається час виконання канонічного розкладу числа. Блок-схема (рис.1) алгоритму розподілення обчислень канонічного розкладу на множники в залежності від введеного числа розкладу N визначає відповідну кількість простих чисел та їх степенів (табл. 1) і розподіляє на потоки для аналізу на подільність. Завдяки високій степені розпаралелення задачі розкладу числа на множники з використанням залишків для кожного вагового коефіцієнта в бінарному представленні числа в плані функціональної декомпозиції і відповідної незалежності даних розподілення обчислень підзадач між потоками здійснюється пропорційно. Наприклад, розглянемо як будуть обчислюватись множники числа 262143 при використанні двох потоків. Для даного числа Таблиця 1 значення залишків двійкового розряду числа за модулями простих чисел та їх степенів буде складатись з 23000 рядків. Потоки поділять для опрацювання таблиці порівну, отже кожен потік опрацює по 11500 рядків таблиці. Після завершення обчислень, результат роботи для першого потоку буде: 3 3 3 7 19 73. В другому потоці не буде знайдено жодного простого множника.

Рисунок 1 – Блок-схема алгоритму розподілення обчислень канонічного розкладу числа

В програмі написаній на мові C++ використовуються функції:

find_prime_factors() розподіляє роботу між заданою кількістю потоків;

create_mod_prime_table() створює таблицю простих чисел і залишків від ділення на кожне просте число;

is_next_prime() визначає чи є число наступним простим числом, використовуючи відомі попередні прості числа;

is_divider() виконує за формулою (2) операції накопичення значень залишків та їх перевірку на рівність зі значенням з множини простих чисел та їх степенів. Накопичення значень залишків проводиться, коли двійкові розряди числа a_i =1;

pool() для організації використання пулу потоків і взаємодії між ними використовується функція бібліотеки CTPL [7], що є надбудовою над стандартою бібліотекою STL, яка має можливість працювати з системними потоками.

Реалізація розробленої паралельної програми здійснюється комп'ютерними системами з багатоядерними процесорами лінійки Intel Core i3/i5/i7, що можуть використовувати підтримку технологій (Intel Turbo Boost Technology 2.0, Intel Hyper-Threading Technology, Intel Smart Cache, Intel HD Graphics 3000). Створені паралельні потоки з можливістю застосування гіперпотоковості (HT) та без реалізуються повноцінними обчислювальними ядрами, що однаково суттєво впливають на продуктивність виконання.

Результати обчислення отримані в 32-бітному режимі на багатоядерних процесорах з тактовою частотою 3.7ГГц:

– Intel Core i3-6100, який має *два* фізичних ядра і кожне фізичне ядро складається з двох віртуальних (HT);

– Intel Core i5-9600, який має *шість* фізичних ядер і не

підтримує технології HT;

– Intel Core i7-2600 який має *чотири* фізичних ядра і кожне фізичне ядро складається з двох віртуальних (HT).

Наступна таблиця 2 показує залежність часу (мілісекунди) витраченого комп'ютерними системами з багатоядерними процесорами на обчислення канонічного розкладу числа на множники від введеного значення кількості потоків.

Таким чином, швидше виконується розроблена паралельна программа обчислення канонічного розкладу числа на множники на процессорах з найбільшою кількістю ядер. Подвоєння продуктивності в процесорі Intel Core i7-2600 з підтримкою гіперпотоковості (HT) у кількості 8 потоків не показало меншого часу в порівнянні з процесором Intel Core i5-9600, який має шість фізичних ядер і не підтримує технології HT. Для процесора Intel Core i3-6100 найкраща швидкодія досягається при використанні 4 потоків, при цьому найбільший приріст спостерігається при задіянні 2–х фізичних ядер процесора.

Таблиця 2 – Залежності часу обчислення канонічного розкладу числа на множники від кількості потоків

К-ть потоків	1	2	3	4	5	6	7	8	9	10	11	12
i3-6100 (HT)	3468	2817	2146	**2041**	2064	2099	2087	2211				
i5-9600	2355	1738	1307	1112	943	826	773	721	**701**	1076	2244	2564
i7-2600 (HT)	2630	2076	1522	1323	1216	1150	1069	1012	970	937	944	**903**

Висновки

1. Вдосконалено метод пробних ділень канонічного розкладу числа на множники з використанням залишків для кожного вагового коефіцієнта у бінарному представленні числа. При цьому виконання канонічного розкладу в обчислювальному середовищі на апаратному або програмному рівнях елементарної операції накопичення значень залишків значно ефективніше з точки зору їхніх менших величин і в порівнянні з виконанням операції ділення.

2. Програмне рішення канонічного розкладу числа на множники реалізовано в IDE Visual C++ 2017, яке може бути й скомпільованим для іншого середовища. Завдяки високій степені розпаралелення задачі розкладу числа на множники з використанням залишків для кожного вагового коефіцієнта в бінарному представленні числа в плані функціональної декомпозиції і відповідної незалежності даних програмне рішення виконує розподілення обчислень підзадач між паралельними потоками пропорційно.

3. Виконання програмного коду канонічного розкладу числа комп'ютерними системами з багатоядерними процесорами лінійки Intel Core i3/i5/i7 показало залежність часу обчислення від кількості сформованих потоків, що відповідно пов'язані з кількістю фізичних та віртуальних (Hyper Threading) ядер процесора.

4. Подальші дослідження пов'язані з розширенням розрядної сітки, більш ніж 32-бітний режим, для бінарного представлення числа розкладу та використання продуктивнішого апаратного інструментарію.

Список літератури

[1] И. М. Виноградов, *Основы теории чисел*, М.: Наука, 1981.

[2] В. А. Орлов, Н. В. Медведев, Н. А. Шимко, А. Б. Домрачева, *Теория чисел в криптографии* : учеб. Пособие, М.: Изд-во МГТУ им. Н. Э. Баумана, 2011.

[3] The Alternative History of Public-Key Cryptography [Електронний ресурс] – Режим доступу: http://cryptome.org/ukpk-alt.htm

[4] Ш. Т. Ишмухаметов, *Методы факторизации натуральных чисел*: учебное пособие. – Казань: Казан. ун. 2011.

[5] Параллельная реализация и сравнительный анализ алгоритмов факторизации с распределенной памятью / Макаренко А.В. Пыхтеев А.В. Ефимов С.С. [Електронний ресурс] – Режим доступу: http://cyberleninka.ru/article/n/parallelnaya-realizatsiya-i-sravnitelnyy-analiz-algoritmov-faktorizatsii-v-sistemah-s-raspredelyonnoy-pamyatyu

[6] Патент 116912 Україна, G06F7/04(2006.01), G06F17/10(2006.01). Пристрій канонічного розкладу числа на множники / І.О. Процько, В.М. Теслюк; Опубл. 25.05.2018, Бюл. №10.

[7] Бібліотека CTPL [Електронний ресурс] – Режим доступу: https://github.com/vit-vit/CTPL.

References

[1] Y. M. Vynohradov, Osnovy teoryy chysel, yzd. 9-e, pererab. – M.: Nauka, 1981. – 167 s.

[2] V. A. Orlov, N. V. Medvedev, N. A. Shymko, A. B. Domracheva, Teoryia chysel v kryptohrafyy : ucheb. posobye. – M. : Yzd-vo MHTU ym. N. Eh. Baumana, 2011. – 223 s.
[3] The Alternative History of Public-Key Cryptography [Elektronnyi resurs] – Rezhym dostupu: http://cryptome.org/ukpk-alt.htm
[4] Sh. T. Yshmukhametov, Metody faktoryzatsyy naturalnykh chysel: uchebnoe posobye. – Kazan: Ka-zan. un. 2011. – 190 s.
[5] Parallelnaia realyzatsyia y sravnytelnyi analyz alhorytmov faktoryzatsyy s raspredelennoi pamiatiu / Makarenko A.V. Pykhteev A.V. Efymov S.S. [Elektronnyi resurs] – Rezhym dostupu: http://cyberleninka.ru/article/n/parallelnaya-realizatsiya-i-sravnitelnyy-analiz-algoritmov-faktorizatsii-v-sistemah-s-raspredelyonnoy-pamyatyu
[6] Patent 116912 Ukraina, G06F7/04(2006.01), G06F17/10(2006.01). Prystrii kanonichnoho rozkladu chysla na mnozhnyky / I.O. Protsko, V.M. Tesliuk; Opubl. 25.05.2018, Biul. №10.
[7] Biblioteka CTPL [Elektronnyi resurs] – Rezhym dostupu: https://github.com/vit-vit/CTPL.
Стаття надійшла: 04.01.2019.

Відомості про авторів

Процько Ігор Омелянович, к.т.н., доцент, НУ "Львівська політехніка", кафедра автоматизованих систем управління, м. Львів, Україна.

Грищук Олександр Васильович, розробник, ТзОВ "Логіка", м. Львів, Україна.

I.O. Prots'ko[1], O.V.Gryschuk[2]

PARALLIZATION OF THE COMPUTATION OF CANONICAL SPLIT A NUMBER ON THE FACTORS

1 – Lviv National Polytechnic University , 2 – LtdC "Lohika", Lviv

И. Е. Процько[1], А. В. Грыщук[2]

РОСПАРАЛЛЕЛИВАНИЕ ВЫЧИСЛЕНИЯ КАНОНИЧЕСКОГО РАЗЛОЖЕНИЯ ЧИСЛА НА МНОЖИТЕЛИ

1 – Национальный университет "Львивська политехника", 2 – ТзОВ "Логика", Львов

ДО ВІДОМА АВТОРІВ

Найновіші правила оформлення і подання статей знаходяться на сайті журналу
http://itce.vntu.edu.ua/