УДК 004.56.5(043.2)

Міходуй Я.М. [1]***, Глинська М.Л.*** [2]***, Івасьєв С.В.*** [1]

[1]*Західноукраїнський національний університет*

[2]*Галицький коледж імені В'ячеслава Чорновола*

ЕФЕКТИВНИЙ АЛГОРИТМ ВИЗНАЧЕННЯ ПРОСТОТИ БАГАТОРОЗРЯДНОГО ЧИСЛА

Вступ. Проблема належності заданого натурального числа до класу простих чи складених чисел є дуже актуальною не тільки в математиці, а й в комп'ютерних науках. Відрізнити просте число від складеного, а також розкласти останнє на прості множники є однією з найважливіших задач арифметики. Пошук великих простих чисел необхідний, наприклад, для забезпечення надійності систем кодування інформації з відкритим ключем. Безпека останніх базується на твердженні, що операція множення двох великих простих чисел є односторонньою функцією.

Мета: На сьогоднішній час перевірка простоти числа здійснюється на основі ймовірнісних тестів Ферма, Соловей – Штрасена, Мілера – Рабіна, які відзначаються великою обчислювальною складністю.

1. Огляд відомих рішень перевірки чисел на простоту

Основною ідеєю тесту Ферма перевірки на простоту є використання теореми Ферма згідно якої, якщо n – просте, то для довільного a, $1 \equiv a \equiv n - 1$ має місце рівність $a^{n-1} \equiv 1 \pmod n$ в іншому n не є простим [1].

У тесті на простоту Соловай – Штрасена використовується критерій Ейлера: якщо n – просте, то $a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right)$ (mod n) для всіх значень a, для яких НСД(a, n) = 1. Слід зазначити, що в даному підході потрібно перевіряти чи $\left(\frac{a}{n}\right)$ буде квадратичним лишком, тобто обчислювати символ Якобі [2], що призводить до часової складності $O(n \cdot Log_2^2 n)$.

Тест Мілера – Рабіна найбільш часто використовується на практиці та називається сильним тестом на простоту. Він базується на наступному факті: нехай n – непарне просте число, при чому $n - 1 = 2^s * r$, де r – непарне, a –натуральне число, яке взаємнопросте з n, тобто НСД(a, n) = 1. Тоді має місце одна із рівностей: $a^r \equiv 1 \pmod n$, або $a^{2^j r} \equiv -1 \pmod n$ для деякого j, $0 \equiv j \equiv s - 1$[3].

Враховуючи те, що в даному методі є операції модулярного експоненціювання, що призводить до значної обчислювальної складності $O(nLog_2{}^2 n)$.

Найпростіший метод встановлення як простоти так і складеності числа був відомий ще у давнину і називається він решетом Ератосфена. Для реалізації його потрібно виписати в ряд числа від 2 до n. Перше число в ряду є простим. Викреслюються з ряду числа, які є кратними 2. Далі взяти друге число, що стоїть в ряду і викреслити всі числа, кратні йому. І так далі брати і-те число та викреслювати кратні йому числа поки і < $\sqrt{n}$. Числа, що залишаться в ряду після операцій викреслення, є простими.

Цей метод є ефективним коли число n невелике (n < 10.000.000.000). При цьому його можна використовувати не тільки для тестування на простоту, а й для пошуку простих чисел у вказаному інтервалі та для розв’язку задачі факторизації.

2. Метод перевірки на простоту з використання векторно-модульного алгоритму модулярного множення в системі залишкових класів

Нехай маємо число *P*-*n*-розрядне. Тоді потрібно, щоб виконувалося наступне співвідношення:

$$m=2^P \, mod \, p=2 \tag{1}$$

згідно теореми Ферма, частковий випадок. Тоді $2^P = M_{(2)}$:

$$M_{(2)}=\underbrace{100000\ldots0000000}_{P\text{-розрядів}} \tag{2}$$

а $p=\sum_{i=0}^{n-1} a_i 2^i$, де $a_i=0,1$, причому p>>n. Розкладаємо $M_{(2)}$ в добуток (3):

$$M_{(2)}= \underbrace{10000..00\ldots00}_{\left[\frac{P}{n+1}\right]} \times \underbrace{10000..00\ldots000}_{\left[\frac{P}{n+1}\right]} \times \ldots \times \underbrace{10000..00..000}_{\left[\frac{P}{n+1}\right]} \times \underbrace{10000000..00\ldots000}_{P-l\left[\frac{P}{n+1}\right]} \tag{3}$$

Тоді для знаходження 2^p *mod p*, обчислимо залишки кожного з множників рівняння (1) шляхом віднімання, тобто:

$$M^{12}_{(2)} = \underbrace{1000000..0000\ldots000000}_{\left[\frac{P}{n+1}\right]} \tag{4}$$

$$M^{2}_{(2)} = \underbrace{100000000..0000\ldots00000}_{P-l\left[\frac{P}{n+1}\right]} - P_{(2)} = \underbrace{100000000..0000\ldots00000}_{P-l\left[\frac{P}{n+1}\right]} \tag{5}$$

в результаті чого отримуємо *l* залишків $M^{12}_{(2)}$ і 1 залишок $M^{2}_{(2)}$, з використанням методу запропонованого в [4]..

Якщо *l* – парне, то на наступному етапі групуємо *l* залишків $M^{12}_{(2)}$ по 2, тобто $M^{12}_{(2)} * M^{12}_{(2)}$, і продовжуємо обчислення до знаходження 2^p *mod p.*

В результаті знаходження 2^p *mod p* отримаємо на кожному кроці пошук одного залишку, кількість яких буде $log_{\,2} l$ та стільки ж множень.

У випадку коли *l* – непарне, то групуємо *l-1* залишків $M^{12}_{(2)}$ по 2, тобто $M^{12}_{(2)} * M^{12}_{(2)}$, і один $M^{12}_{(2)} * M^{2}_{(2)}$ проводимо розрахунок 2^p *mod p,* для якого потрібно $log_{\,2}\, l$ кроків на кожному з яких перевіряємо на парність і непарність кількості залишків. Така процедура призводить до знаходження на кожному двох залишків, кількість яких буде $log_2 l$ та стільки ж множень.

В порівнянні з відомими, розроблений метод характеризується високою швидкодією та меншою обчислювальною складністю, що дозволяє ефективно застосовувати його при перевірці багаторозрядних чисел на простоту.

Оскільки відомі методи, переважно є ймовірнісними, то вони однозначно не вказують на необхідні умови простоти числа.

Імовірнісний алгоритм перевірки багаторозрядних чисел на простоту буде виглядати так:

1. на вході маємо число $P_{1..n}$;
2. знаходимо залишок 2^{n+1} по модулю $P_{1..n}$.;
3. знаходимо цілу частину від ділення $C_{i..k}$=P/(n+1) та залишок від ділення U;
4. обмежуєм залишок від ділення модулем P: U = U mod P;
5. ініціалізуємо змінні R=0;res=1;
6. якщо C mod 2=0 тоді R=R+1; якщо ні тоді крок 9;
7. робимо побітовий зсув змінної, для визначення кількості 0 в молодших розрядах C=C/2;
8. перехід на крок 6;
9. якщо C_i = 1 тоді res_i=1;
10. res =res *2; i=i+1;
11. якщо res більше P тоді res= res- P;
12. Якщо і менше розрядності C тоді крок 9;
13. R=R mod P;
14. res=(res*R) mod P;
15. res= (res*U) mod P;

Якщо res= 2 то число ймовірнісно просте і алгоритм повертає значення true, якщо ж ні то значення false.

В результаті таких обчислень, отримаємо значення $2^p \bmod p$, і якщо значення $2^p \bmod p = 2$, то *p*-просте число.

Розроблений експериментальний додаток на основі запропонованого алгоритм дозволяє перевіряти на простоту багаторозрядні числа та будувати числову послідовність з ймовінісно простих чисел для перевірки з існуючою множиною простих чисел.

В нашому випадку зустрічаються числа, які задовольняють умові 1 і не є простими числами. Слід зазначити, що вони зустрічаються крайньо рідко в нижня границя співпадіння в співвідношенні 1:1000, а верхня в 3:1000.

В таблиці 1 подані результати дослідження розподілу чисел, які задовольняють умову 1 і є складеними.

Таблиця 1 – Розподіл складених чисел, які задовольняють умову $2^p \bmod p = 2$

Номер	Діапазон чисел	Складені числа, які задовольняють умову 2^p mod p = 2
1	[1..1000]	341
2	[1000..2000]	1105
3	[1000..2000]	1387
4	[1000..2000]	1729
5	[2000..3000]	2047

Результати чисельного експерименту показують, що використання даного методу дозволить однозначно визначати прості числа з врахуванням таблиці 2.1, тобто тих чисел які є винятками.

В результаті таких обчислень, отримаємо значення $2^p \bmod p$, і якщо значення $2^p \bmod p = 2$, то p-просте число. Порівняння обчислювальних складностей розробленого та існуючих методів приведено на рисунку 1.

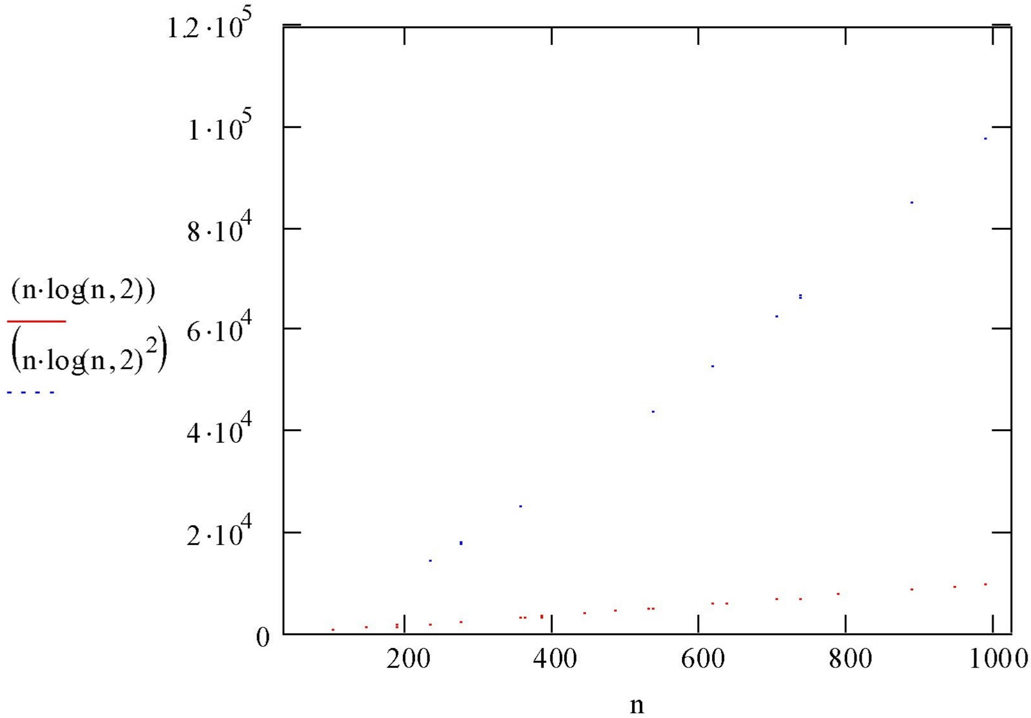

Рисунок 1 - Поріняння складностей алгоритмів перевірки на простоту чисел

Часова складність розробленого методу перевірки натуральних чисел на простоту складає $O(n\,Log_2 n)$ з врахуванням векторно-модульного алгоритму модулярного множення.

Висновок. Розроблений ймовірнісний метод перевірки на простоту багаторозрядних чисел, який на відміну від відомих характеризується низькою обчислювальною складністю $O(n\,Log_2 n)$ та незначною складністю реалізації алгоритму, що дозволяє ефективно застосовувати його при перевірці багато розрядних чисел на простоту.

Перелік використаних джерел.

1. Agrawal M. PRIMES is in P / M.Agrawal, N.Kayal, N.Saxena.– Annals of Mathematics.– 2004, v.160, p. 781–793.

2. Buhler J. Algorithmic Number Theory: Proc. ANTS-III / J.P. Buhler(ed.).– Portland, OR, v.1423, Lect.Not.Comp.Sci. Springer–Verlag, 1998, 640 p.

3. Venturi D. Lecture Notes on Algorithmic Number Theory./ D. Venturi. – Springer-Verlag, New-York, Berlin, 2009, 217 p.

4. Ишмухаметов Ш.Т. Методы факторизации натуральных чисел: учебное пособие /Ш.Т. Ишмухаметов.– Казань: Казан. ун. 2011.– 190 с.