Some Results on Fermat's Theorem and Trial Division Method

Sajda Kareem Radi, Nagham Ali Hameed

Mechanical Eng. Dept., College of Engineering / Mechanical Eng. Dept., College of Engineering /Al-Mustansiriya University, Baghdad, Iraq Al-Mustansiriya University, Baghdad, Iraq

Receiving Date: 05-10-2010 - **Accept Date:** 29-12-2010

Abstract

An integer n>1 is called "Prime" if it has no other positive divisors than 1 and itself (within the set of integers), other wise n is said to be a" composite".

Prime numbers are very important in today's society. The methods that determine, if a particular integer is prime or composite, are called primarily testing. This paper discusses and writes the algorithms for primarily testing. Also, new theorems has been obtained for primarily tests, and been used as a test as in the "trial division" method and "Fermat's theorem". A computer program has been built, and been operating using (" MATLAB 7").

Introduction

Prime numbers are rather old objects in mathematics; however, they did not loose their fascination and importance. There have been many tests of primality and algorithms to carry out these tests and which are created throughout the years. The written history of distinguishing prime numbers from composites goes back to Sieve of Eratosthenes who came up with the first recorded algorithm for primality testing in the 3rd century BC. [1]. In the 17th century, mathematicians (Fermat, Legendre, Gauss, etc) considered primality testing and factoring to be some of the most important problems in arithmetic. Their work laid the foundation of a new age in primality testing, which began in the 1970. Miller in 1976, Solovay- Strassen in 1977,and Rabid in 1980 developed efficient algorithms for primality testing and factoring [2].

The first primality tests were not run on the computers. They were computed by hand. Today very large prime numbers are required and are nearly impossible to write down. The largest known prime number today is 243112609 – 1. So it is obviously impractical to work with these numbers without using computers.

Researchers and mathematicians have been striving to find an unconditionally deterministic polynomial – time algorithm to test for primality. The first progress was only made in 2002. The larger prime ever found without a computer was $\frac{2^{148}-1}{17}$, which has 44 digits (found by Ferrier in 1951). He used a desk calculator and a variation of Fermats Little Theorem. In the same year, using an electronic computing device, a prime with 79 digits was found by Miller and Wheeler. Today, the largest prime is over 12 million digits long [3].

In this paper, many of known primality testing methods have been discussed using proposed tests. These methods have been implemented using "Matlab 7" programming language to obtain the result which could be evaluated.

Tests for Primality

There are many different primality tests and methods which can be classified as follows:

Tests for Numbers of Special Forms:

These tests deal with numbers of special form , such as:

Mersenne numbers invented by the monk Mersenne in (1644) are defined by [1].

$$M_p = 2^p - 1$$

Example: If p = 11, M11 = 2047

Fermat numbers are defined by

$$F_n = 2^{2^n} + 1$$

These tests are fast, simple, and provide rigorous proofs that their results are correct .These tests are as follows:

Theorem 1: (Pepin's Test): For $n \geq 2$, F_n is prime if and only if

$$3^{(F_n-1)/2} \equiv -1 (\text{mod } F_n)$$

Proof: See [4]

Theorem 2: (Lucas – Lehmer's Test) : Let p > 2 be a prime, and let $M_p = 2^p - 1$ be the Mersenne number, then M_p is prime if and only if M_p divides S_{p-1} (equivalently, if and only if $S_{p-1} \equiv 0 (\text{mod} n)$) where the numbers (Sn) $n \geq 1$ are given by the following recurrence relation S1 = 4 , $S_{n+1} = S_n^2 - 2$, $n \geq 1$.

Proof : See [5].

Deterministic Tests

These methods can test any number of any forms, but these methods are more theoretical than practical. Compared with probabilistic (in section 3) primality test methods, the output results of deterministic are absolutely correct. In other words, when a positive odd number is tested, the output result has only two possible situations either this number is a prime or composite [5].

These methods are as follows:

Lucas's Test: Let n be a positive integer. If there exists an integer 1< a < n such that

$a^{n-1} \equiv 1 \pmod{n}$

and for every prime factor q of n − 1

$a^{\frac{n-1}{q}} \neq 1 \pmod{n}$ then n is prime. If no such number a exist, then n is composite [6].

Theorem 3 (Wilson's Theorem)

This test proposed by John Wilson and published by Edward Waring in 1770 . For any positive integer p, p is prime if and only if

$$(p-1)! \equiv -1 \pmod{p}$$

Wilson's theorem is not only necessary but also sufficient for the primality test.

Proof: See [7].

Proth's Test: For $p = k * 2^n + 1$ with k odd and $2^n > k,$ if there exists an integer ~ such that

$$\sim^{(\frac{p-1}{2})} \equiv -1 \pmod{p}$$

then p is prime [5].

Pocklington's Test: The Pocklington's Lemer primality test devised by Henry Labourn Pocklington and Derrick Henry Lehmer to decide a given number N is prime which is formulated as follows:

1- q is prime $q / N-1$ and $q > \sqrt{N} - 1$

2- $a^{N-1} \equiv 1 \pmod{N}$

3- $\gcd(a^{(N-1)/q} - 1, N) = 1$

Then N is prime [8].

Probabilistic Tests

Miller- Rabin's Test: Given a positive odd integer n and let $n = 2^r s + 1$, where s is odd number. Then follow the testing numbers

Choose a random positive integer a with $1 \le a \le n-1$. If $a^s \equiv 1 (\mathrm{mod}\, n)$ or $a^{2^j S} \equiv -1 (\mathrm{mod}\, n)$ for some $0 \le j \le r-1$, then n passes the test [5].

Solovay - Strassen's Test : Let n be a positive integer n > 1 , choose at random , k >1 numbers a , 1< a< n , such that gcd (a, n) =1, and compute

$$a^{\frac{(n-1)}{2}} \equiv \left(\frac{a}{n}\right) (\mathrm{mod}\, n) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

If (1) fails to hold for some a , then n is composite [9].

Lehmann's Test: Let n be an integer, choose a random number a less than n, then calculate

$a^{(\frac{n-1}{2})}$ (mod n) ,if $a^{(\frac{n-1}{2})} \neq 1 \ or \ -1 \ (\mathrm{mod}\, n)$, n is definitely not prime [10].

Fermat's Little Primality Test: Let p be a prime number and let a be any integer which is not a multiple of p then

$$a^{p-1} \equiv 1 (\mathrm{mod}\ p)$$

In other word $(a^{p-1} - 1)$ is multiple of p for every integer a [11] .

Example:

1- P = 5, a = 3. Then 34 = 81 -1 = 80, which is a multiple of 5.

$$3^4 \equiv 1 \ (\mathrm{mod} 5)$$

2- P = 6, a = 5. Then 55 = 3125, and 3125 − 1 = 3124 , which Not a multiple of 6.

This example shows that the theorem can fail if p is not a prime number.

Results: The following theorems are the proposed tests in this study. These theorems can be used for testing as Fermat's theorem.

Theorem 4: If n is an odd prime, then

$$2^{n-2} \equiv \frac{n+1}{2} (\mathrm{mod}\, n)$$

.

Theorem 5 : If n is an odd prime, then

$$2^{\frac{n-3}{2}} \equiv (\frac{n+1}{2}) \ or \ (\frac{n-1}{2}) \ (\mathrm{mod}\ n)$$

.

In general.

Theorem 6 : If n is an odd prime, then

$$2^{\frac{n-k}{2}} \equiv \pm (\frac{n-1}{2})^{\frac{(k-1)}{2}} \mod n$$

For any prime k.

Example: Let k = 5, then

$$2^{\frac{n-5}{2}} \equiv \pm (\frac{n-1}{2})^2 \mod n$$

For $n = 19 : 2^7 = 128 \equiv -81 \mod 19$.

For $n = 21 : 2^8 = 256 \neq \pm 100 \mod 21$.

Therefore , 19 is a probable prime and 21 is a composite.

Trial Division Method: Trial division as a primality test is based upon the following theorems

Theorem 7 : An odd integer n is prime if and only if is not divisible by any prime less than or equal to $\sqrt{n}$.

Proof : See [3]

This method has the advantage of not only providing a proof of primality of a prime n, but of discovering a non trivial factorization for composite n. The main drawback of the trial division algorithm is that it takes too long if n has no small prime factors [12].

Results:

Before introducing the formula for primality testing which is obtained in this study, it is rather to state the following theorem :

Theorem 8 : (de Polignac's formula) : If n is a positive integer and p a prime , then the exponent of the biggest power of p that divides n ! is :

$$e = \sum_{k=1}^{\infty} \left[\frac{n}{p^k} \right]$$

Proof : S ee [13]

The following two theorems are the tools for the primality testing which is got in this work.

Theorem 9: If n is an integer, then

$$\sum_{k=1}^{\infty} \left[\frac{n}{p^k} \right] = \sum_{k=1}^{\infty} \left[\frac{n-1}{p^k} \right] \quad , \forall \ p \le \sqrt{n}$$

If and only if n is a prime.

The following theorem reduces the calculation of dividing n by the distinct powers of p to a single number, which is p .

Theorem 10: If n is an integer, then

If and only if n is prime.

Proof: If n is a prime , the following must be proved

$$\left[\frac{n}{p} \right] = \left[\frac{n-1}{p} \right]$$

It is known that n is prime if and only if no prime $p \le \sqrt{n}$ divides n ,

$\therefore$ $p \nmid n$, since $n = q_1 p + r_1$ *and* $n-1 = q_1 p + (r_1 - 1)$,where 0 < r1 < p and 0 < r1 -1 < p Therefore ,

$$\frac{n}{p}=\frac{q_1 p}{p}+\frac{r_1}{p} \quad and \quad \frac{n-1}{p}=\frac{q_1 p}{p}+\frac{(r_1-1)}{p}$$

$$\Rightarrow \frac{n}{p}=q_1+\frac{r_1}{p} \quad and \quad \frac{n-1}{p}=q_1+\frac{(r_1-1)}{p}$$

By the definition of the greatest integer, the following is obtained

$$\left[\frac{n}{p}\right]=q_1 \quad and \quad \left[\frac{n-1}{p}\right]=q_1$$

$$\therefore \left[\frac{n}{p}\right]=\left[\frac{n-1}{p}\right] \qquad \forall\, p \le \sqrt{n}$$

On the other hand , if

$$\left[\frac{n}{p}\right]=\left[\frac{n-1}{p}\right] \qquad \text{.....(2)}$$

then $\left[\frac{n}{p}\right]=\left[\frac{n-1}{p}\right]=0$

It is clear that :

$$\left[\frac{n}{p^k}\right]-\left[\frac{n-1}{p^k}\right]=\begin{cases}1 & if \quad p^k \mid n \\ 0 & if \quad p^k \dagger n\end{cases} \qquad \text{..... (3)}$$

From (2) and (3) it is concluded that $p \dagger n$, $\forall\, p \le \sqrt{n}$, therefore n must be a prime .

Now if an integer n is to be tested using theorem 5555 all the primes that are less than or equal to square root of n must be listed , then $\left[\frac{n}{p_1}\right]$ *and* $\left[\frac{n-1}{p_1}\right]$ are computed if they are equal continue in this manner , and compute $\left[\frac{n}{p_2}\right]$ *and* $\left[\frac{n-1}{p_2}\right]$ until we reach p_k which is less than or equal to $\sqrt{n}$, if for all these they are equal then n must be prime otherwise it is a composite .

Example: Let n = 11

$$\left[\frac{11}{2}\right]=\left[\frac{10}{5}\right]=5 \quad ,\left[\frac{11}{3}\right]=\left[\frac{10}{3}\right]=3$$

$\therefore$ 11 is a prime number .

Implementation

This section introduces the system implementation of five primality testing algorithms , which are as follows .

1 - Algorithm for the Miller – Rabin Primality Test

1- Input n >1 an odd integer to be tested for primality , t >1

2- let n -1 =2S . r such that r is odd

3 - Repeat from 1 to t

4- Choose a random integer a , $2 \leq a \leq n-2$

5- Compute x = ar mod n

6- If $\neq 1$, and $x \neq n-1$ then j = 1

7- While $j \leq s-1$ and $x \neq n-1$ do the following

8- compute x = x2 mod n

9- If x = 1, then print (" composite ") : j = j + 1

10- If $x \neq n-1$ then print (" composite ")

11 - Print (" Prime")

End

Run : n =19 , a =2 ,s = 1 , r = 9

19 prime

2- Algorithm for the Proth's Test

1- Input positive integer n

2- Input odd integer $k < 2n$

3- Input positive integer a

4 - Let p = k . 2n +1

5- If $a^{\frac{p-1}{2}} \equiv -1 (\mathrm{mod}\, p)$ for some a then output " p is prime "

6- Output P is not prime.

Run : is p = 13 prime ?

n = 2 , k = 3 , a = 15626

13 prime

3– Algorithm for the Lucas Test

1- Input odd integer n > 2 to be tested for primality ,k

2- Factor n-1 to its prime

3- L oop1: repeat k times

4- Choose a random integer in the range [2, n – 1]

5- If $a^{n-1} \neq 1 \pmod n$ then print composite otherwise

6 Loop 2 : for all prime factors q of n-1:

7- If $a^{(n-1)/q} \neq 1 \pmod n$

8- If we did not check this equality for all prime factors of n – 1 then do next loop2

9- Otherwise print prime otherwise do next loop1print composite .

Run : is 5 prime ?

a =3 , q = 2

5 prime

4 - Algorithm for the Lucas- Lehmer Test

1- Input : a Mersenne number p

2- calculate n = 2P-1

3- Use trial division to check if s has any factors between 2 and sqrt of p. If it does, then print (composite)-

4- Let u = 4

5 - For k = 1 to p – 2 do the following

6- Compute u =(u2 – 2) mod n

7- If u = 0 then print (“ prime ”)

8- print " composite "

Run : To test n =7

p = 3 , u =((4* 4) -2) mod 7 = 0

7 prime

5- Algoithm for the Solovay- Strassen Test

1-Input an odd integer n and t

2- For I = 1 to t do the following:

3-Choose a random integer a , $2 \le a \le n - 2$

4- Compute r =a(n-1)/2 (mod n)

5- If $r \neq 1$, and $r \neq n - 1$, then print (" composite")

6- Compute the Jacobi symbol s = (a/ n)

7- If $r \neq s \pmod n$ then print (" composite")

8- Print (" prime ")

Run : is n=7 prime ?

a = 4 , r =1 , s = 1

r = s =1 , 7 prime

References

1. R.Slezeviciene,J.Steuding,S.Turskiene,"Recent Breakthrough in Primality Testing", Nonlinear Analysis : Modelling and Control", vol.9, No.2, p(171-184), 2004.
2. Shafi Goldwasser and Joe Kilian.,"Primality Testing using Elliptic Curves",Journal of ACM,vol.46,No.4,1999.
3. Chelsea Arrington, "Primality Testing ", 2010. http://primes.utm.edu/notes/by_year.html
4. David M. Bressoud ," Factorization and Primality Testing " , Springer- Verlag , New York,1989.
5. Chia-Longwn, Der- Chyuan Lou, Te-Jen Chang , "Computational Reduction of Wilson's Primality Test for Modern Cryptosystems ",Informatica ,vol.33,p(453-458),2009.
6. Song Y.Yan, "Number Theory for Computing", Springer-Verlag, Berlin, 2000.
7. David M.Burton , " Elementary Number Theory". Second Edition, Web Publishers, 1989.
8. Koblrtz, Neal, "A course in Number Theory and Cryptography" , 2nd Ed., Springer, 1994.
9. Al-Shamari A. S.S.," Analytical Study of Some Public-key Cryptosystems Depending on Some Evaluation Paramaters",M.Sc.Thesis,Univ. of Tech.,Baghdad,1995.
10. Hamdoon A.," Evaluation of Primality Testing Algorithms " , in proceeding 5th Annual

Al- Raffidiain University College , 1999 .

11. Darren Glass, "Primality Testing " ,2004.
12. Carl Pomerance, "Recent Development in Primality Testing" , The Mathematical Intelligencer. Vol.3, No.3, PP.97-105, 1981.
13. I.Niven . , " An Introduction to the Theory of Number " , Third edition , John Wiley and Sons,New York , 1972 .