Identifying the Strong Pseudoprime from a Carmichael number using the Miller-Rabin primality test through Python programming

Manimaran Sambanthamoorthi,
University of Technology and Applied Sciences, Salalah, Sultanate of Oman
manimaran.sal@cas.edu.om

Abstract

A primality test is a condition to check whether a given integer is prime or not. There are some large odd integers which may pass the primality test, still it need not be a prime number. The primality test for a large composite odd integer is a time consuming process. This paper discusses the ways of finding the strong pseudoprime from the bases of a Carmichael number by using the Miller-Rabin test through Python programming.

Keyword: Primes, pseudoprime, Euler pseudoprime, Strong pseudoprime, Carmichael Number, and Python programming

1. Introduction

A prime number is a positive integer which is divisible by 1 and itself only. If a large odd number, then the task is to identify whether the given number is a prime number or a composite number. Checking an integer n, large odd number is a prime or composite takes a lot of time.

Here the objective is to discuss the two types of tests to check whether n is a strong prime or not, by using Miller-Rabin test and to discuss the Solovay-Strassen test during the process.

To check for how many number of bases $0 \leq b < n$, for which it satisfies the Fermat's Little theorem.

Before introducing the two tests, see the definition of the Fermat's Little Theorem, if n is prime then for any b, such that the greatest common divisor (gcd)(b, n)=1, one has

$b^{n-1} \equiv 1(mod\ n)$

(1.1.1)

If n is not prime, then it is still possible that (1.1.1) will be true. Such type of numbers is called pseudoprimes [3].

A pseudoprime is an odd composite number which behaves and looks like a prime number and it satisfies the condition given in the Fermat's Little Theorem. Hence it creates a challenge in choosing a prime number. Every prime number satisfies the condition (1.1.1), but the idea here is to find the composite numbers which satisfies the condition (1.1.1).

1.1 Definition [1][3]:

If n is an odd composite number and b is an integer such that gcd(b, n)=1 and (1.1.1) holds, then n is called a pseudoprime[3] to the base b.

We need to understand that if n fails the test (1.1.1) for a single base $b \in (Z/nZ)$, then n fails (1.1.1) for at least half of the possible bases $b \in (Z/nZ)$.

But in some cases, (1.1.1) holds for every base $b \in (Z/nZ)$, then we say that such integers are called Carmichael number[2][4].

1.2 Carmichael Number [2][4]:

A Carmichael number must be the product of at least three distinct primes. The smallest Carmichael number is $n = 561 = 3 \times 11 \times 17$, which is the product of three distinct prime numbers.

1.3 Solovay-Strassen Primality Test

Now the definition for the Euler pseudoprime [1] is given. Let n be an odd positive integer and let $\left(\frac{b}{n}\right)$, denote the Jacobi symbol, then, if n is a prime number, then

$b^{\frac{(n-1)}{2}} \equiv \left(\frac{b}{n}\right) (mod\ n)$, for any integer b (1.2.1)

If n is a composite number, then (1.2.1) will hold for at least 50% of all the bases $b \in (Z/nZ)$ that hold (1.2.1), then n is called an Euler pseudoprime to the base b.

2. Methodology

Here we describe the Solovay-Strassen primality tests [1].

Suppose n is a positive odd integer and to know whether n is a prime or composite number. Choose k integers $0 < b < n$ at random. For each b, first compute the both the sides of (1.2.1).

If the two sides are not congruent modulo n, then you know that n is composite, despite passing all of the tests is at most $\frac{1}{2^k}$.

Thus Solovay-Strassen primality test is a probabilistic algorithm which leads either to the conclusion that n is composite or it is probably prime.

Now to discuss the strong pseudoprime concept which is based on the Euler pseudoprime.

2.1 Miller-Rabin Test [1]

Suppose n is a large positive odd integer and $b \in (Z/nZ)$. Let n be an odd composite number and write $n - 1 = 2^s t$, with t odd. Let $b \in (Z/nZ)$. If n and b satisfy the condition either

$b^t \equiv 1(mod\ n)$ or there exists r, $0 \leq r < s$, such that $b^{2^r t} \equiv -1(mod\ n)$ (2.1.1),

then n is called a strong pseudoprime to the base b.

2.2 Proposition [1]:

Any strong pseudoprime of n to the base b is also an Euler pseudoprime.

If n is a strong pseudoprime, then n is a strong pseudoprime to the base b for at most 25% of all

$0 < b < n$.

The following procedure explains the ways to identify the strong pseudoprime from the given integer n.

To check the large odd composite number is prime or composite.

Let $n - 1 = 2^s t$, with t odd and choose a random integer b, such that $0 < b < n$.

By computing, $b^t mod\ n$, if the result is ± 1 from (2.1.1), for any particular b, then we go on to another b. Otherwise, square $b^t mod\ n$, until -1 is obtained, until n passes the test.

If $b^{2^{r+1}} \equiv 1(mod\ n)$, while $b^{2^r} \equiv -1(mod\ n)$, then n fails the test and we know that n is a composite number. If we try (2.1.1) for all our random choices of b.

Suppose k different bases b are checked, then that n has at most 1 out of 4^k chance of being composite.

This is because, if n is composite, then at most $\frac{1}{4}$ of the bases $0 < b < n$ satisfy (2.1.1).

Miller-Rabin test is better than Solovay-Strassen test, where the analogous estimate is a 1 out of 2^k chance[4].

3. Overview on Python Programming

Python is a programming language, it runs on multiple platforms like Windows, Mac OS X, Linux, Unix. It is a free and open source. Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to browsers to games.

In Python programming, we are going to find the strong pseudoprime by using the Miller-Rabin primality test. Python is an interesting programming language to find the strong pseudoprime.

It checks the following congruence

$b^t \equiv 1(mod\ n)$ or there exists r, $0 \leq r < s$, such that $b^{2^r t} \equiv -1(mod\ n)$

The input value of n should be given and if the results are +1 or – 1, then we conclude that the integer n is a strong pseudoprime, otherwise it is not a strong pseudoprime.

3.1 Algorithm for finding strong pseudoprime

The following algorithm gives the idea about the strong pseudoprime is identified from the range 1 to 561 modulo 561.

Step 1: Enter the value of n, say 561.

Step 2: Take all the bases $b \in (1,561)$, such that greatest common divisor of b and n is 1.

Step 3: Calculate $n - 1 = 2^s t$, where t is odd.

Step 4: Calculate $b^t \equiv 1(mod\ n)$. If it is true, then it is a strong pseudoprime.
Otherwise go to step 5.

Step 5: Calculate $b^{2^r t} \equiv -1$, where $0 \leq r < s$. If it is true, then it is a strong pseudoprime.
Otherwise it is not a strong pseudoprime.

Step 6: The process ends.

3.2 Example

Let us consider the base $b = 169$, with modulo 561. Here gcd(169, 561)=1. Let us check the first congruence $b^t \equiv 1(mod\ n)$ of the Miller-Rabin test, here $t = 35$.

$(169)^{35} \equiv 1\ (mod\ 561)$.

Here the first congruence of the Miller-Rabin test is true for the base $b = 169$. Therefore we are able to conclude that the base $b = 169$ is a strong pseudoprime to the modulo 561.

Also consider an another example, with base $b = 256$, with modulo 561. Here gcd(256,561)=1.

Let us consider the base $b = 256$, with modulo 561. Here gcd(256, 561)=1. Let us check the first congruence $b^t \equiv 1(mod\ n)$ of the Miller-Rabin test, here $t = 35$.

$(256)^{35} \equiv 1\ (mod\ 561)$.

Here the first congruence of the Miller-Rabbin test is true for the base $b = 256$. Therefore we are able to conclude that the base $b = 256$ is a strong pseudoprime to the modulo 561.

4. Conclusion

From this research paper, the idea is to conclude that the identification of the strong pseudoprime by using the Miller-Rabin primality tests and by using the application of the Python programming. Here we are able to find the strong pseudoprime from the bases of the Carmichael number $n = 561$. Similarly the Euler pseudoprime from the bases of the number $n = 561$ can be found. Every strong pseudoprime is a Euler pseudoprime and every Euler pseudoprime is a pseudoprime and the converse need not be true. Also all the bases of a Carmichael number are a pseudoprime. We can choose any odd composite positive non-Carmichael number and all the strong pseudoprimes in their bases can be found.

5. REFERENCES

[1] A Course in Number Theory and Cryptography by Neal Koblitz.

[2] H. Cohen and H.W. Lenstra, Jr., “Primality Testing and Jacobi sums”, Math. Comp. 42(1984), 297- 330.

[3] J.D. Nixon, “Factorization and primality tests”, American Math. Monthly 91 (1984), 333-352.

[4] C. Pomerance, “Recent developments in primality testing”, The Math. Intelligencer 3 (1981), 97-105.