Comparative Analysis of Primality Testing Algorithms for Enhanced Pseudoprimes Detection

Background

Modern cryptography algorithms like RSA rely on primality tests to generate large random primes.

Types of Primality Tests

Deterministic Primality Test

• Provide definite answers about the primality of a number.
• Not computationally feasible for very large numbers.
• AKS primality test (2002) is polynomial time but not efficient enough in practice.

Probabilistic Primality Test

• Do not provide definite answers about the primality of a number: Certain composite numbers are classified as primes. Such numbers are called pseudoprimes.
• Require less computational time for large numbers, often relies on sequences.

Current Predicament

A deterministic primality test that is able to handle the cases needed for modern cryptography does not exist currently. With Quantum Computing changing modern cryptography, fundamental questions, such as finding efficient deterministic primality tests, are becoming more significant.

Procedure

1. Change the parameters and the initial values for the special cases of Lucas-Type sequence.
2. For each sequence, compute the first several pseudoprimes.
3. Generate graphs that visualize the size and amount of initial pseudoprimes of Lucas-Type sequence.
4. Identify and analyze special cases that stand out from these visualizations.

Hypothesis

• Sizes of the smallest pseudoprimes is indicative of the quality of the primality tests (e.g. Dougherty-Bliss and Zeilberger [1]).
• Many pseudoprimes are expected to be divisible by 2 and 3. In general, very little structure is expected.
• Not expecting to be able to predict the pseudoprimes by simple formulas.

A Well-Known Quote

"The problem of distinguishing prime numbers from composite numbers and of resolving the latter into their prime factors is known to be one of the most important and useful in arithmetic. … Further, the dignity of the science itself seems to require that every possible means be explored for the solution of a problem so elegant and so celebrated." (Gauss, 1801)

Lucas-Type Sequence

$V_{n+2} = c_1 V_{n+1} + c_0 V_n,\ V_0 = 2,\ V_1 = c_1$

If p is prime, then we have the congruence: $V_p \equiv V_1\ (mod\ p)$

```
def lucas_type_primality_test(n, c0, c1):
    a = 2
    b = c1
    for _ in range(n - 1):
        c = (c1 * b + c0 * a) % n
        a = b
        b = c
    return (b - c1) % n == 0
```

For instance, when $c_0 = 1$ and $c_1 = 2$, the sequence is:

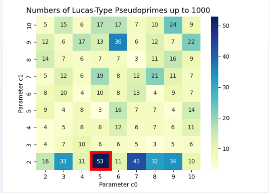
2	2	6	14	34	82	198	478
V_0	V_1	V_2	V_3	V_4	V_5	V_6	V_7

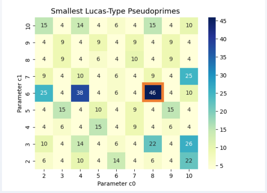
$V_3 = 14 \equiv 2 (\mathrm{mod}\ 3)$ ➡ 3 passes test (prime)

$V_6 = 198 \equiv 0 (\mathrm{mod}\ 6)$ ➡ 6 is not a prime

$V_4 = 34 \equiv 2 (\mathrm{mod}\ 4)$ ➡ 4 passes test (pseudoprime)

Probabilistic primality tests can give us two possible outputs, which are "not a prime" or "likely prime" (prime or pseudoprime).





*The case where $c_0 = 5, c_1 = 2$ has a relatively high number of pseudoprimes.

The case where $c_0 = 8, c_1 = 6$ has a relatively big smallest pseudoprime.

Analyze each case.

The least promising primality test: Lucas-Type Sequence(5, 2)

Pseudoprimes of Lucas-Type Sequence(5, 2) up to 1,000:
[4, 6, 8, 9, 12, 16, 18, 24,…… 782, 864, 972] ➡ 53 pseudoprimes
***This appears to include all numbers of the form $2^i \cdot 3^j$ except for 2, 3.**

After restricting to 2-rough pseudoprimes (those not divisible by 2):
[9, 27, 45, 81, 243, 405, 729, 759] ➡ 8 pseudoprimes

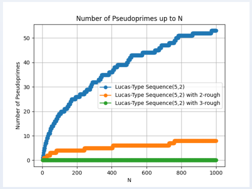
After restricting to 3-rough pseudoprimes (those not divisible by 2 or 3):
[] ➡ 0 pseudoprimes

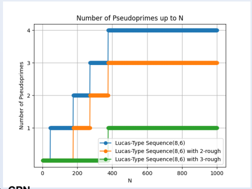
A promising primality test: Lucas-Type Sequence(8, 6)

Pseudoprimes of Lucas-Type Sequence(8, 6) up to 1000:
[46, 177, 273, 377] ➡ 4 pseudoprimes

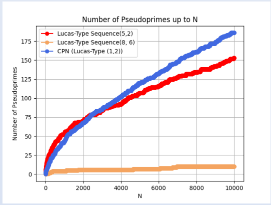
After restricting to 2-rough pseudoprimes:
[177, 273, 377] ➡ 3 pseudoprimes

After restricting to 3-rough pseudoprimes:
[377] ➡ 1 pseudoprimes

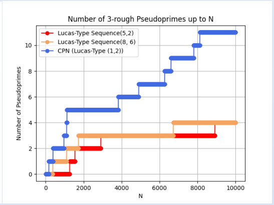




***A surprising comparison with a special case, Companion Pell Number**

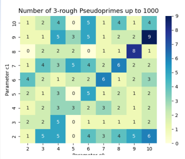


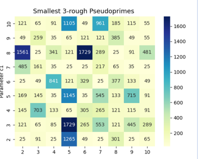
***Lucas-Type (5, 2) vs. Lucas-Type (8, 6) vs. CPN up to 10,000 (Restricting to 3-rough)**

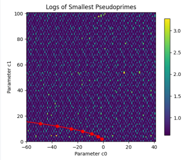


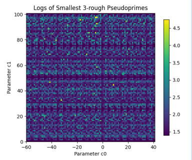
Findings

• After restricting to 3-rough pseudoprimes, the Lucas-Type(5, 2) looks comparable with Lucas-Type(8, 6).
• A lot of pseudoprimes of these three special cases are multiples of 2 and 3.
• Dougherty-Bliss and Zeilberger have proved that $2^i \cdot 3^j$ with i ≥ 3, $j \geq 0$ are always pseudoprimes for CPN.



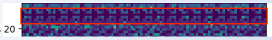






*Red curve highlights the special case of Lucas-Type Sequence, the well-known Fermat's primality test (in this case: $V_n = 2\alpha^n, c_0 = -\alpha^2,\ c_1 = 2\alpha\ , \alpha \in \mathbb{Z}.$)
➡ Smallest pseudoprimes become larger after restricting to 3-rough numbers.

*Logs of Smallest 3-rough Pseudoprimes.



➡ There seem to be certain patterns among pseudoprimes for different choices of parameters.

Special Case: Fermat's Primality Test

Lucas Type sequence can also be written as:

$$V_n = \alpha_1^n + \alpha_2^n \qquad (c_0 = -\alpha_1\alpha_2,\ c_1 = \alpha_1 + \alpha_2)$$

If $\alpha = \alpha_1 = \alpha_2$, the congruence $V_p \equiv V_1\ (mod\ p)$ turns into $2\alpha^p \equiv 2\alpha (mod\ p)$.
This is essentially Fermat's Little Theorem:

$$\alpha^p \equiv \alpha (mod\ p)$$

Conclusion

• It's not straightforward to determine the quality of primality tests. For instance, only looking at smallest pseudoprimes can be misleading.
• For certain sequences, there are some patterns among these pseudoprimes. For instance, all numbers of $2^i \cdot 3^j$ within Lucas-Type(5, 2) are pseudoprimes.
• One primality test that looks worse than the other might actually be a competitive test in large numbers.
• Consider the divisibility by 2 and 3 when it comes to comparing primality tests because otherwise we might discard tests because of a "flaw" that could be easily fixed.

Future Direction

• Write the code to generate higher-order recursion to examine if they will have a big difference in numbers of pseudoprimes.
• Study stronger congruences similar to how Miller Rabin is stronger than Fermat's primality test.
• Implement a version of binary exponentiation when computing the sequences to test large numbers and higher-order recursion.

Reference

[1] Robert Dougherty-Bliss and Doron Zeilberger. "Lots and Lots of Perrin-Type Primality Tests and Their Pseudo-Primes." *arXiv preprint arXiv:2307.16069* (2023).