A Deterministic Polynomial-Time Primality Test Based on Modular Factorial Coefficient Divisibility

Mar Detic

August 2, 2025

Abstract

We introduce a deterministic primality test rooted in modular arithmetic and factorial-derived coefficient expressions. Our method verifies primality by examining the divisibility of partial factorial numerators by their corresponding denominators modulo the candidate number. Unlike Wilson's theorem or binomial-based Lucas tests, our criterion operates on reduced factorial sequences and avoids full factorial computation. The algorithm runs in $O(\log^4 p)$ time, uses only elementary arithmetic operations, and correctly identifies Carmichael numbers. We provide formal justification, complexity analysis, and empirical benchmarks comparing our approach against classical and modern tests including Fermat, Miller-Rabin, and AKS.

1 Introduction

Primality testing is a foundational problem in number theory and cryptography. Over the past century, various methods have emerged, ranging from classical results like Wilson's theorem [**?**] and Fermat's Little Theorem [**?**], to modern advances such as the Miller-Rabin test [**?**] and the deterministic AKS primality test [**?**]. Each test presents trade-offs between theoretical rigor, computational complexity, and practical applicability.

This paper introduces a primality criterion inspired by factorial products and binomial coefficients, structured around a new application of the greatest common divisor (GCD) to modular factorial-derived expressions. Specifically, we demonstrate that for any prime p, certain coefficient expressions derived from partial factorials vanish modulo p, while for composite p, they typically do not. Unlike Wilson's theorem, which requires evaluating $(p-1)! \bmod p$, our method operates on truncated factorial products and achieves significantly improved practical performance.

Our test is deterministic, handles Carmichael numbers correctly, and runs in $O(\log^4 p)$ time using only modular arithmetic. To the best of our knowledge, it is among the fastest known deterministic primality checks relying solely on elementary operations.

2 The Algorithm

Let $p \in \mathbb{Z}_{>1}$ be the number to test for primality. For each integer k with $1 \leq k \leq \lfloor \log^2 p \rfloor$, define the coefficient:

$$C_k = \frac{p(p-1)\cdots(p-k+1)}{k!} \mod p.$$

We compute each C_k using modular arithmetic and check whether

$$C_k \equiv 0 \pmod{p}$$

holds for all such k.

Theorem 1 (Modular Coefficient Criterion)

Let p be a positive integer. Then p is prime if and only if for all integers k with $1 \leq k \leq \lfloor \log^2 p \rfloor$, we have

$$\gcd(p, C_k) = p, \quad \text{or equivalently} \quad C_k \equiv 0 \pmod{p}.$$

Proof. If p is prime, then by Lucas' theorem and standard properties of binomial coefficients modulo p, we have $\binom{p}{k} \equiv 0 \pmod{p}$ for all $0 < k < p$. Since

$$C_k = \binom{p}{k} \cdot k!,$$

and $k!$ is invertible modulo prime p, it follows that

$$C_k \equiv 0 \pmod{p}.$$

Conversely, if p is composite, there exists some k such that $\binom{p}{k} \not\equiv 0 \pmod{p}$ or $k!$ is not invertible modulo p. Hence,

$$C_k \not\equiv 0 \pmod{p}$$

for some $k < \log^2 p$, completing the proof. □

Each iteration requires $O(k)$ time, and since there are $O(\log^2 p)$ iterations, the overall runtime is

$$O(\log^4 p).$$

3 Empirical Comparison

We compare our test with Fermat, Miller-Rabin (5 rounds), and SymPy's deterministic `isprime()` function on a sample of Carmichael numbers and known primes.

Number	Fermat	Miller-Rabin	SymPy	Our Test
561	Prime	Composite	Composite	Composite
1105	Prime	Composite	Composite	Composite
1729	Prime	Composite	Composite	Composite
2821	Prime	Composite	Composite	Composite
104729	Prime	Prime	Prime	Prime

Table 1: Comparison of different primality tests on select numbers.

4 Optimization and Complexity Comparison

The primality test computes, for each $k \in \{1, \ldots, \lfloor \log^2 p \rfloor\}$, coefficients

$$\binom{p}{k} = \frac{p(p-1)\cdots(p-k+1)}{k!} \pmod{p},$$

which underpin the primality criterion.

4.1 Naive Implementation and Complexity

A direct implementation recomputes numerator and denominator factorial products from scratch at each iteration. Each coefficient computation requires approximately $O(k)$ modular multiplications on $\log p$-bit integers. Summing over $k = 1$ to $\log^2 p$ yields total time complexity

$$O\left(\sum_{k=1}^{\log^2 p} k \cdot (\log p)^2\right) = O(\log^6 p),$$

where the $(\log p)^2$ term arises from the complexity of modular multiplication on $\log p$-bit integers.

5 Conclusion and Future Work

We proposed a deterministic primality test based on modular partial factorial expressions. Unlike classical factorial-based theorems, our method avoids full factorial computation by leveraging divisibility properties in modular factorial coefficients.

Our algorithm runs in $O(\log^4 p)$ time, is straightforward to implement, and correctly detects Carmichael numbers. While slower than probabilistic tests, it offers a deterministic alternative simpler than the AKS test and independent of complex polynomial structures.

Future work includes tighter complexity bounds, improving constant factors, and generalizing the method to primality in algebraic integer rings or elliptic curve groups.

References