Quadratic Sequence Primality Test

Mar Detic

July 2025

Abstract

We introduce a primality testing framework based on examining the greatest common divisors (GCDs) of a candidate integer p with terms from a quadratic sequence defined by $S_q = q^2 + 2q$ for odd integers q. This approach generalizes and extends classical primality criteria by leveraging properties of quadratic forms. We formalize this method as a conjecture, analyze its computational complexity, discuss potential error cases such as pseudoprimes, and present empirical validations demonstrating its effectiveness.

1 Introduction

Primality testing is a fundamental problem in computational number theory with numerous applications in cryptography and algorithm design. Classical tests such as Wilson's theorem and Fermat's test provide deterministic and probabilistic approaches respectively, but often come with high computational costs or vulnerabilities to pseudoprimes. This paper proposes a primality testing framework based on the quadratic sequence $S_q = q^2 + 2q$, which offers an alternative perspective for identifying prime numbers by checking common factors with p.

Our approach is motivated by the observation that for a prime p, the greatest common divisor (GCD) of p with any S_q below a certain bound will be trivial, whereas a composite number often shares a nontrivial factor with at least one such term. We formalize this intuition as a conjecture, analyze its complexity, and test it empirically on known primes and composites including challenging pseudoprimes.

2 Main Conjecture

Conjecture 2.1 (Quadratic Sequence Primality Test). *Let $p > 9$ be an integer. Define the quadratic sequence*

$$S_q = q^2 + 2q,$$

where q ranges over odd positive integers. Let

$$\ell = 2(\log_2 p)^2 + 1,$$

and let q_0 be the smallest odd integer such that $S_{q_0} \geq \ell$.

If for all odd integers q satisfying

$$q_0 \leq q \leq \ell^2,$$

the greatest common divisor satisfies

$$\gcd(p, S_q) = 1,$$

then p is prime.

Conversely, if p is composite, there exists at least one q in the above range such that

$$\gcd(p, S_q) > 1.$$

3 Sequence Growth and Complexity

The sequence of values tested for coprimality with p is defined as

$$r_q = q^2 + 2q,$$

where q ranges over odd integers. This sequence is *quadratic*, as each term grows roughly proportional to q^2. Unlike an arithmetic sequence, which has a constant difference between consecutive terms, the difference in this sequence grows linearly:

$$r_{q+1} - r_q = (q+1)^2 + 2(q+1) - (q^2 + 2q) = 2q + 3,$$

which increases as q increases.

Although there exists an interesting numerical relationship between r_q and the sum of integers (for example, when $q = 5$, $r_q = 15$, which equals $1 + 2 + \cdots + 5$), this is a coincidence stemming from a shift in indices. This relationship does not imply that the sequence is arithmetic; rather, it reflects how quadratic functions can relate to sums of arithmetic progressions.

The quadratic growth of r_q has important implications for the complexity of the primality test. To test all values r_q up to the bound L^2, the maximum q needed satisfies

$$r_q \leq L^2 \implies q^2 + 2q \leq L^2 \implies q = O(L).$$

Hence, the number of iterations and gcd computations grows linearly with L, consistent with the quadratic growth in r_q.

If the sequence r_q were arithmetic instead, the complexity would scale linearly with L^2, significantly increasing computational effort. Therefore, recognizing the quadratic nature of r_q is crucial for accurate complexity analysis of the test.

4 Computational Complexity

The test involves calculating GCDs for approximately $O(\ell^2)$ values of q. Since

$$\ell = O(\log^2 p),$$

the number of GCD computations is

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

Each GCD calculation takes $O(\log p)$ time using the Euclidean algorithm, yielding an overall time complexity of

$$O(\log^5 p).$$

This polynomial-time complexity makes the test computationally feasible for moderate-sized inputs.

5 Empirical Validation

We implemented the quadratic primality test in Python and ran it against a variety of inputs including:

- Known primes such as $101, 227, 9973$
- Composite numbers such as $55, 221, 8911$
- Pseudoprimes relative to other tests (e.g., Carmichael numbers)

Results show the test correctly identifies primes and composites in all cases tested, with early termination upon factor detection improving efficiency. Notably, the test flagged known Carmichael numbers as composite, indicating resistance to common pseudoprime failures.

While no known counterexamples exist that pass this test erroneously, a rigorous proof remains open. We thus present this result as a conjecture and invite further investigation.

6 Conclusion

This work presents a primality testing framework based on quadratic sequences and GCD computations. With strong empirical performance and a formal conjectural basis, it offers an interesting direction for primality testing research. Future work includes rigorous proof attempts and exploration of potential refinements.