# Efficiency of the Sieve of Eratosthenes and Comparative Analysis of Prime-Finding Algorithms

## Abstract

*Provide a brief summary of the paper, including the main objectives, methods, key findings, and conclusions.*

## Introduction

*Introduce the topic of prime number algorithms, the importance of efficiency in computational number theory, and an overview of what the paper will cover.*

Example equ: The quadratic formula is given by $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

## History of the Sieve of Eratosthenes

*Discuss the origins of the Sieve of Eratosthenes, its historical significance, and its foundational role in the study of prime numbers.*

One of the most popular prime number algorithms is the Sieve of Eratosthenes. This algorithm was founded by a famous Greek scientist named Eratosthenes of Cyrene. Eratosthenes had many talents, one of which was in mathematics. Mathematics during his life was nothing compared to modern-day mathematics; thus, it was easier to become talented at mathematics.

Eratosthenes discovered a systematic way to find primes. This system involved starting with a prime and then marking all multiples of that prime as composite. The numbers that do not get crossed off end up being the primes.

The Sieve of Eratosthenes has played a significant role in finding "small" primes back during the time when computers did not exist. Although better algorithms have been found to compute primes using computers, this algorithm is one that every math or computer science student should be taught.

## Efficiency of the Sieve of Eratosthenes

*Analyze the computational complexity of the Sieve of Eratosthenes, including time and space requirements. Discuss its practical performance and any optimizations that have been developed.*

**Pattern**

1. **Initialization**: Create a list of consecutive integers from 2 through n.
2. **Marking Non-Primes**: Starting with the first prime number $p = 2$ , mark all multiples of $p (i.e., 2p, 3p, 4p, ...)$ as non-prime.
3. **Iteration**: Move to the next unmarked number in the list, which is the next prime, and repeat the marking process.
4. **Termination**: Continue until $p^2 > n$ . All remaining unmarked numbers are primes.

**Example of the Sieve of Eratosthenes** Start by listing all numbers from 2 to 16. Initially, assume all numbers are prime.

**P** : Indicates the number is currently considered **prime**. **C** : Indicates the number has been marked as **not prime** (composite).

| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P | P | P | P | P | P | P | P | P | P | P | P | P | P | P |

Step 1: Mark Multiples of 2 The first prime number is 2. Mark all multiples of 2 (except 2 itself) as **not prime**.

| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P | P | C | P | C | P | C | P | C | P | C | P | C | P | C |

**Marked Multiples of 2:** 4, 6, 8, 10, 12, 14, 16

Step 2: Mark Multiples of 3 The next unmarked number is 3, which is prime. Mark all multiples of 3 (except 3 itself) as **not prime**.

| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P | P | C | P | C | P | C | C | C | P | C | P | C | C | C |

**Marked Multiples of 3:** 6, 9, 12, 15

Termination The next unmarked number is 5. Since $5^2 = 25 > 16$, we can stop the process. All remaining unmarked numbers are primes.

Final List of Primes:

| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P | P | C | P | C | P | C | C | C | P | C | P | C | C | C |

**Primes up to 16:** 2, 3, 5, 7, 11, 13

**Time Complexity** The time complexity of an algorithm measures how the running time increases with the size of the input. For the Sieve of Eratosthenes:

- **Initialization**: Creating the list takes $O(n)$ time.
- **Marking Non-Primes**: For each prime $p$ , marking its multiples up to $n$ takes $O(n/p)$ time.

The total time spent marking multiples is:

$$\sum_{p \leq \sqrt{n}} \frac{n}{p}$$

The sum of the reciprocals of the primes up to $\sqrt{n}$ is approximately $n \log \log n$. Therefore, the overall time complexity is:

$$O(n \log \log n)$$

This is nearly linear, making the sieve exceptionally efficient for large values of n.

**Benefit's:** The sieve of Eratosthenes is good for generating very large prime numbers due to its time complexity of $O(n \log \log n)$. The sieve is also very straightforward to understand and implement by hand or through algorithms on a computer. The sieve also provides a deterministic way to identify all primes up to $n$, ensuring that all primes can easily be identified and won't be missed. Unlike probabilistic algorithms, the sieve doesn't rely on chance, offering guaranteed results.

**Challenges** While the Sieve of Eratosthenes is highly efficient and simple for generating all prime numbers up to a moderate limit, it faces significant challenges when dealing with very large numbers or specific primality testing requirements. In such cases, alternative algorithms may offer better performance and scalability based on the specific needs of the application. It is designed to generate all primes within a specified range rather than efficiently verifying the primality of a single very large number. Additionally, the sieve operates with a fixed upper limit, meaning that the range of numbers to be sieved must be known in advance. If primes beyond this initial range are needed, the sieve must be rerun with a higher $n$, which can be inefficient and time-consuming.

**Real world usage** The Sieve of Eratosthenes has been instrumental in generating comprehensive lists of prime numbers efficiently, which are essential for mathematical research and educational purposes. In the field of cryptography, it aids in the generation of large prime numbers necessary for creating secure keys in systems like RSA encryption. The sieve also facilitates prime factorization in computational mathematics by providing a list of primes up to a certain limit, which can be used to decompose integers into their prime components. Additionally, it is a staple in programming contests and competitive programming, where quick prime generation is often required under strict time constraints. Researchers utilize the sieve to study the distribution of primes and to test conjectures related to prime numbers, such as the Goldbach and Twin Prime conjectures. Moreover, the sieve serves as a foundational tool in developing more advanced algorithms and optimizing data structures that rely on prime numbers for enhanced performance. Its simplicity and efficiency make the Sieve of Eratosthenes an invaluable asset across various domains in mathematics and computer science.

# Other Prime-Finding Algorithms

### Trial Division

**Description** *Explain how the Trial Division algorithm works and its methodology for finding prime numbers.*

The Trial Division algorithm is typically the easiest to understand prime number algorithm. This algorithm aims to determine if a number can be factored compared to how the Sieve of Eratosthenes works by eliminating multiples of primes. There is a strategic strategy to determine the possible factors of a number. The possible factors of a number turn out being all numbers less than the square root of said number

The algorithm goes as follows:

1. Take in a positive integer n
2. Determine the range of numbers you need to check as possible factors
3. Iterate through the range of possible factors
4. If the factor divides evenly into the number, then it is composite
5. If all possible factors do not divide evenly into the number, then it is prime.

**Efficiency** *Analyze the computational complexity and practical efficiency compared to the Sieve of Eratosthenes.*

**Dont know if I should include this here or in the description:** Trial Division algorithm is efficient is determining if single numbers are prime; however, the Sieve of Eratosthenes is efficient is finding all the primes in a specified limit. A main trade off for the Trial Division algorithm is that …

### Sieve of Atkin

**Description** *Detail the Sieve of Atkin algorithm, highlighting its innovative approach to prime number generation.*

The Sieve of Atkin algorithm is a spin off of the Sieve of Eratosthenes but contains a few changes. The main change is that the Sieve of Atkin algorithm does work beforehand to remove non-prime numbers and then proceeds by marking off **squares** of primes.

The algorithm goes as follows:

1. Create a results list to store all primes, start with 2, 3, 5 initially in this list.
2. Create a sieve list containing all positive integers up to specified limit, start by marking them all non-prime.
3. For each number in this list, we are concerned with the remainder r when divided by 60
    1. "If r is 1, 13, 17, 29, 37, 41, 49, or 53, flip the entry for each possible solution to $4x^2 + y^2 = n$." ("Sieve of Atkin," n.d.)
    2. "If r is 7, 19, 31, or 43, flip the entry for each possible solution to $3x^2 + y^2 = n$." ("Sieve of Atkin," n.d.)
    3. "If r is 11, 23, 47, or 59, flip the entry for each possible solution to $3x^2 - y^2 = n$ when $x > y$." ("Sieve of Atkin," n.d.)
    4. If r is not one of those, ignore it.
4. Start with low number in sieve list.
5. Find the next number still marked prime.
6. Add this number to results list.
7. Now, square this number and mark all multiples of this square as composite.
8. Repeat steps 4-7 until the end of the sieve list is reached.

**Efficiency** *Evaluate its performance metrics and compare its efficiency with other sieves.*

### Miller-Rabin Primality Test

**Description** *Describe the Miller-Rabin Primality Test method for finding primes, including its simplicity and implementation.*

The Miller-Rabin Primality Test is a probabilistic primality test. This means that this algorithm determines whether a number is likely prime but does not determine for certain. Gary Miller discovered a deterministic version of this test; however, this relies on a big problem in math being true called the extended Riemann hypothesis. Michael Rabin modified Millers version to make it probabilistic, and thus not dependent on an unproven problem.

This algorithm relies on mathematical concepts such as "Strong probable primes" and "Choice of bases".

Strong probable primes

"For a given odd integer $n > 2$, we can write $n - 1$ as $2^s d$ where $s$ is a positive integer and $d$ is an odd positive integer. Now lets consider an integer $a$ (called a base) which is co-prime to $n$, Then $n$ is said to be a strong probable prime to base $a$ if one of these congruence relations holds:

- $a^d \equiv 1 \pmod{n}$
- $a^{2^r d} \equiv -1 \pmod{n}$ for some $0 \leq r < s$

If neither of these congruence relations hold, then $n$ is composite and $a$ is considered a **witness** to the compositeness of $n$" ("Miller-Rabin primality test," n.d.)

Choices of bases

Picking a base $a$ at random will yield a fast probabilistic test. Most bases $a$ will be a witness to $n$ being composite and thus will reduce odds of a false positive to a very small rate. The typical interval for choosing a base is $1 < a < n - 1$

**Efficiency** *Discuss the limitations in terms of efficiency, especially for large numbers, and compare it to sieve-based methods.*

# Comparative Analysis

*Compare and contrast the Sieve of Eratosthenes with the Sieve of Sundaram, Sieve of Atkin, and Trial Division in terms of efficiency, scalability, and practicality. Include tables or charts if necessary.*

**Results from current program run:**

| Algorithm | Largest Prime | Elapsed (s) | Elapsed (HH:MM:SS) |
|---|---|---|---|
| Trial Division | 1,188,567,577 | 12,000.00s | 03:20:00 |
| Sieve of Eratosthenes | 1,188,671,413 | 12,000.00s | 03:20:00 |
| Sieve of Atkin | 1,215,682,201 | 12,000.00s | 03:20:00 |
| Miller-Rabin | 1,188,766,571 | 12,000.00s | 03:20:00 |

# Conclusion

*Summarize the key findings of the paper, reiterate the efficiency of the Sieve of Eratosthenes, discuss the circumstances under which other algorithms may be more effective, and suggest potential areas for future research.*

# References

*List all the sources cited in the paper in the appropriate citation style.* Undetermined what style yet for citations https://medium.com/geekculture/sieve-of-eratosthenes-one-of-the-oldest-algorithms-still-prevalent-as-if-it-were-born-yesterday-c3e854df5dc6 - provides a few optimizations for the sieve of Eratosthenes

https://en.wikipedia.org/wiki/Sieve_of_Atkin https://en.wikipedia.org/wiki/Miller%E2%80%93Rabin_primality_test#