A Division-Free Constructive Framework for Number Generation under Divisibility Constraints

Author: Amer Alaa Eldin Attia Gomaa

Assistant: Habib Mohamed Attia Gomaa

Date: May 20, 2025

Contact: ameralaah99@gmail.com

Abstract

This paper introduces a set of novel arithmetic techniques for generating integer sequences that exclude all values divisible by any element of a given set of divisors, without relying on division (/) or modulo (%) operations. These methods are grounded in constructive number theory and leverage properties of least common multiples (LCMs), exponential offsets, and coprime multiplicative constructions. We define each method formally, provide mathematical proofs, and demonstrate effectiveness through algorithmic examples. The techniques are applicable to secure computation, embedded arithmetic, and low-level algorithm design.

1. Introduction

The ability to generate integer sequences free of certain divisors is a common requirement in fields such as cryptographic key scheduling, embedded systems, and mathematical sequence filtering. Classical approaches rely on modulo operations or sieving techniques. However, in constrained computing environments, such operations may be costly or insecure. We propose a set of arithmetic constructions that avoid the use of division and modulo entirely, while ensuring that the generated outputs are not divisible by any member of a given set D = {d1, d2, ..., dn}.

2. Preliminaries

- LCM(D): Least common multiple of all elements in the set D.
- gcd(a, b): Greatest common divisor of integers a and b.
- Coprime: Two numbers are coprime if gcd(a, b) = 1.
- Power (P): Positive integer exponent.

We also introduce the following arithmetic variables for general formulations:
- X: A generated output number.
- Me: An even integer multiplier.
- Mo: An odd integer multiplier.
- Ce: A constant added to even-multiplier expressions (e.g., 1 to generate odd, 2 to generate even).
- Co: A constant added to odd-multiplier expressions (e.g., 2 to generate odd, 1 to generate even).

3.1 Method 1: LCM-Multiplier with Constant Offset

Definition:
Let L = LCM(D). Define:
$f(x) = Me \times L + Ce$ or $f(x) = Mo \times L + Co$,
where $Ce \bmod di \not\equiv 0$ and $Co \bmod di \not\equiv 0$ for all di in D.

Theorem 1: f(x) is not divisible by any d in D.
Proof: Since $L \equiv 0 \bmod d$, any $Me \cdot L \equiv 0$. Then:
$f(x) = L \cdot k + c \Rightarrow f(x) \bmod d = c \bmod d \not\equiv 0$. □

Example:
Let D = {3, 5}, so L = 15.
- $f(1) = 1 \times 15 + 2 = 17$
- $f(2) = 2 \times 15 + 1 = 31$
- $f(3) = 3 \times 15 + 2 = 47$
All results are not divisible by 3 or 5.

3.2 Method 2: Power-Based Offset with LCM

Definition:

$f(x) = Mo \times L + 2^P$,

Assume $2^P \bmod d \not\equiv 0$ for all d in D.

Theorem 2: The result of $f(x)$ avoids all divisors in D.

Proof: Since $Mo \times L \equiv 0 \bmod d$, then:

$f(x) \bmod d = 2^P \bmod d \not\equiv 0$. □

Example:

Let $D = \{3, 5\}$, $L = 15$, $Mo = 1$.

- $f(1) = 1\times15 + 2^1 = 17$
- $f(2) = 1\times15 + 2^2 = 19$
- $f(3) = 1\times15 + 2^3 = 23$

3.3 Method 3: Coprime Multiplicative Product Summation

Definition:

$f(x) = (a_1 \times p_1)^P + (a_2 \times p_2)^P$,

Subject to:

- $p_1, p_2 \notin D$
- $\gcd(a_1 \cdot p_1, a_2 \cdot p_2) = 1$
- One term must be even.

Theorem 3: $f(x)$ is not divisible by any d in D.

Proof: Since terms are coprime and exclude all primes in D, their sum avoids divisibility. □

Example:

Invalid: $f(x) = (3\times7)^1 + (5\times7)^1 = 56$ (invalid, 7 is repeated)

Valid: $f(x) = (2\times3)^1 + (5\times7)^1 = 6 + 35 = 41$

4. Applications and Use Cases

- Cryptographic systems requiring timing-safe logic
- Embedded systems lacking division instructions
- Functional and mathematical sequence generation with logic-only constraints

5. Density Analysis of Generated Sequences

We analyze the density δ of generated sequences within a range $[1, N]$, defined as:
$\delta = (\text{Number of Valid Elements}) / N$

For Method 1 using LCM with offsets, we expect:
$\delta \approx (\text{LCM}(D) - \#\text{Excluded Residues}) / \text{LCM}(D)$

Example ($D = \{3,5\}$):
- LCM = 15
- Valid residues per 15: 8
- Density $\approx 8/15 \approx 0.533$

As LCM increases, spacing between valid numbers grows, decreasing density.

6. Comparison with Classical Methods

Method	Uses Division/Modulo	Complexity	Density Control	Side-Channel Safe
Sieve of Eratosthenes	Yes	$O(n \log \log n)$	High	No
Proposed Method 1	No	$O(n)$	Medium	Yes
Proposed Method 2	No	$O(n)$	Medium	Yes
Proposed Method 3	No	$O(n)$	Customizable	Yes

The proposed methods are optimized for environments where side-channel resistance and low-level computation constraints are critical.

7. Function Generator: Automated Sequence Production

We propose a generic sequence generator to scale the approach:

Input:
- Divisor set D
- Range [start, end]
- Method type (1, 2, or 3)
- Optional parameters: offset constants, exponent P

Output:
- List of integers not divisible by any $d \in D$

Algorithm Sketch:
1. Compute LCM(D).
2. Use method formula to generate candidates.
3. Validate if needed for Method 3.

This algorithm can be implemented in Python or C for cryptographic or logic-based sequence production.

8. Conclusion

We presented three mathematically sound and computationally lightweight methods to generate sequences of integers not divisible by a given set of numbers, without using division or modulo. The methods are suitable for embedded logic, cryptographic preprocessing, and integer obfuscation tasks.

Repository

https://github.com/ameralaa/semiPrimes-cryptography