International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

Comparative Study of CRT and Extended Euclidean Algorithm in Solving Linear Diophantine Equations

[1]Tanushka , [2]Palak Sharma
[1-2]M.Sc. Student, Associate Professor, Chaudhary Charan Singh University
[3]Dr. Rajiv Kumar
[3]Associate Professor, Department of Mathematics, Chaudhary Charan Singh University

Abstract: Linear Diophantine equations are central to number theory and have wide-ranging applications in cryptography and computational mathematics. Among the classical methods for solving such equations, the Extended Euclidean Algorithm (EEA) and the Chinese Remainder Theorem (CRT) stand out as foundational approaches. The EEA is primarily used to compute greatest common divisors and modular inverses, making it effective for single-equation problems, while CRT provides a framework for solving systems of congruences with pairwise coprime moduli. This research project aims to conduct a comparative study of these two methods, focusing on their solvability conditions, computational efficiency, and practical applicability. The study will include theoretical analysis and algorithmic implementation to evaluate performance across varied problem sets. The expected outcome is a clear understanding of the strengths and limitations of each method, identifying contexts where one approach is more suitable than the other. This comparison will contribute to a deeper insight into algorithmic number theory and provide guidance for selecting appropriate techniques in mathematical and cryptographic applications.

Keywords: linear Diophantine equation, Chinese remainder theorem, Extended Euclidean algorithm, Euclidean algorithm, Bézout's identity.

INTRODUCTION

Number theory, one of the oldest branches of mathematics, is deeply concerned with the properties of integers. Within this field, **Diophantine equations** occupy a pivotal role, as they restrict solutions to integers alone. These equations are named after *Diophantus of Alexandria* (third century AD), whose pioneering work laid the foundation for what is now called *Diophantine Analysis*. Despite the limited mathematical tools available to him, Diophantus investigated quadratic equations such as

$ax^2 + bx = c$, $ax^2 = bx + c$, $ax^2 + c = bx$

which, though equivalent in modern algebra, were treated separately due to his lack of a concept of zero and avoidance of negative coefficients. His focus on positive solutions illustrates both the constraints and ingenuity of early number theory.

The study of Diophantine equations has since evolved into a rich discipline, encompassing equations with no solutions, finitely many solutions, or infinitely many solutions. The twentieth century marked a turning point, with the formulation of general theories that systematized the study of these equations and extended their applications to diverse areas such as cryptography, coding theory, and optimization.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

Among the various types, the **linear Diophantine equation** represents the most fundamental case. It is expressed as

$ax + by = c \quad a, b, c \in Z.$

This equation admits integer solutions if and only if c is divisible by $\gcd(a, b)$. When a particular solution (x, y) exists, the complete set of solutions can be described parametrically as

$$(x + kv, y - ku)\ , k \in Z$$

where $u = \frac{a}{\gcd(a,b)}$ and $v = \frac{b}{\gcd(a,b)}$. This characterization reveals the infinite families of solutions that arise when solvability conditions are met. The importance of integer solutions in mathematics cannot be overstated. Unlike real or complex solutions, integers are discrete and exact, making them indispensable in problems involving divisibility, modularity, and countability. They form the backbone of number theory, underpin cryptographic systems such as RSA, and play a crucial role in combinatorics, optimization, and computer science. In practical contexts, integers correspond to countable quantities—people, objects, or units—ensuring that mathematical models align with reality where fractional values are meaningless. Thus, the study of integer solutions is not merely theoretical but essential for bridging mathematics with real-world applications. As for solving linear Diophantine equation there are many ways but here, we only discuss two prominent methods that is by extended Euclidean algorithm and by Chinese remainder theorem.

One of the first systematic knowledge of the discipline we now call number theory came from ancient China, where queries leading to linear indeterminate equations and systems of linear congruences occurred. Indeterminate linear equations of two unknowns occurred mainly in commercial tasks, e.g. by selling several kinds of goods at integer prices. Apart from mathematics (e.g. in astronomy) appeared more complicated problems leading to systems of linear indeterminate equations with multiple unknowns, which we classify within the congruence do main nowadays. A very significant knowledge from this period, in particular, is the so-called Chinese remainder theorem, which determines the necessary and sufficient conditions for the solvability of a system of linear congruences. In the mathematical treatise, which comes from the ancient Chinese mathematician Sun-c' (4th century AD), we can find this problem: "An unknown number of things is given. If they are counted by three, two remain, if they are counted by five, three remain, if they are counted by seven, there remain two. Determine the number of these things." Solving this problem is easy. The least common multiple of the numbers 3, 5, and 7 is 105, so one solution of the problem will occur in the set {1, 2...,105}. From the second condition, it follows, that the solution is a number in the form 5n+3from the given set. Therefore, it suffices to check the numbers 3,8, 13...,103, if the remainder after division by three (resp. seven) meets the problem conditions. The smallest suitable solution is x = 23. There are still other solutions, which are "repeated by 105", and are in the form y ≡ 23 mod 105. The problem of the Chinese mathematician Sun-c' reached both India and Europe, and especially in the 18th and 19th centuries engaged the attention of mathematicians L. Euler and C. F. Gauss. The Chinese used a lunar calendar, in which small and large months changed by 29 and 30 days, so the year had 354 days.

However, such a calendar brought problems, because due to the different length of the solar year, which the Chinese set at 3651 4 days, it happened that the beginnings of the years were not in fixed dates. The Chinese inserted after 19 lunar years another 7 lunar months (around 600 BC). At the end of the first millennium AD, Chinese mathematicians and astronomers devoted great effort to calculate the so-called Great period, i.e. the question in how many years the three periods will meet the tropical year with 3651 4 days, the lunar month with 29499 940 days and a sixty-day cycle. The problem led to a system of congruences with large numbers. The Chinese mathematician Qin Jiushao came up with a solution to the system of congruences $x \equiv 193440 \pmod{1014000}$, $x \equiv 16377 \pmod{499067}$, where $x = 6172608n$, (n is the number of years elapsed since the Great Period). There are several ways to formulate the Chinese remainder theorem. It provides a systematic framework for solving systems of congruences. When moduli are pairwise coprime, CRT guarantees a unique solution modulo the product of the moduli. This makes CRT especially powerful in handling multiple simultaneous equations, and it is widely used in cryptography (e.g., RSA decryption optimization), distributed computing, and error correction codes. Whereas,

The Extended Euclidean Algorithm traces its origins back to Euclid's *Elements* (c. 300 BC), where the original Euclidean algorithm was introduced as a systematic method for finding the greatest common divisor (GCD) of two integers. Over time, mathematicians recognized that this procedure could be extended to not only compute the GCD but also express it as a linear combination of the inputs, a result now known as Bézout's identity. This extension became crucial for solving linear Diophantine equations, a topic explored by Diophantus in the 3rd century AD, and later formalized by Étienne Bézout in the 18th century. Islamic scholars such as Al-Khwarizmi contributed to the algorithmic tradition during the medieval period, while Euler and Gauss in the 18th and 19th centuries applied the extended algorithm to modular arithmetic and congruences, solidifying its role in number theory. In the modern era, the Extended Euclidean Algorithm has become indispensable in computational mathematics and cryptography, particularly for calculating modular inverses in systems like RSA, making it a bridge between ancient algorithmic thought and contemporary applications. The Extended Euclidean Algorithm (EEA**)** provides a systematic and efficient method for solving LDEs in two variables. Building upon the classical Euclidean Algorithm, which computes the greatest common divisor (gcd) of two integers, the extended version not only finds the gcd but also expresses it as a linear combination of the given integers:

$$ax + by = \gcd(a, b)$$

This property, known as **Bézout's identity**, is the cornerstone of solving LDEs. If $\gcd(a, b)$ divides c, then the equation ax+by=c has integer solutions. The EEA can be used to construct one particular solution, which can then be generalized into a family of solutions parameterized by an integer variable.

Although both methods are foundational in number theory, they serve different scopes: EEA is efficient for single equations and modular inverses, while CRT excels in systems of congruences. Despite their importance, there is limited comparative research that systematically evaluates their

efficiency, solvability conditions, and applicability across different problem types. Most literature treats them independently, without highlighting contexts where one method may outperform the other. This research project aims to conduct a **comparative study of CRT-based and EEA-based approaches** in solving linear Diophantine equations. By providing a structured comparison, this study will contribute to a deeper understanding of algorithmic number theory. It will also offer practical guidance for selecting appropriate methods in cryptographic applications and computational mathematics, where efficiency and reliability are critical.

Theoretical framework

2.1 Linear Diophantine Equations (LDEs)

At its heart, a linear Diophantine equation is simply a linear equation where we constrain the solution to be integers. For a single equation with two variables, it takes the form $ax + by = c$, where a, b, c are given integers, and we seek integer values for x and y. What I found particularly interesting early in my studies is that not all such equations have integer solutions. The fundamental condition for solvability states that an LDE $ax + by = c$ possesses integer solutions if and only if the greatest common divisor of a and b, denoted as $\gcd(a, b)$, divides c. If this condition isn't met, there's no point in searching for integer solution.

When solution do exist, they form an infinite set, which can be elegantly described by a general solution. If (x_0, y_0) is any particular integer solution to$ax + by = c$, then all other integer solutions are given by:

$$x = x_0 + k\left(\frac{b}{\gcd(a, b)}\right)$$
$$y = y_0 - k\left(\frac{a}{\gcd(a, b)}\right)$$

For any integer k. this structure, a particular solution offset by multiple of basis vector derived from the homogeneous equation, is a recurring theme in linear algebra and is a crucial for understanding the completeness of solutions.

Extending this, we often encounter systems of linear Diophantine equations, which involve multiple linear equation and several variables, all constrained to have integer solutions. These systems can be quickly become complex, demanding robust methods to determine their consistency and to characterize their entire set of integer solutions. Much like their single-equation counterparts, the general solution for a system of LDEs, when one exists, typically comprises a particular solution vector combined with linear combination of basis vectors for the associated homogeneous system. Navigating these complexities is precisely where our enhanced approach seeks to offer a more streamlined pathway.

2.2 Chinese remainder theorem (CRT)

Theorem 1.1. Let there be a system of solvable linear congruences

$$a_1 x \equiv b_1 (mod\ m_1)$$
$$a_2 x \equiv b_2 (mod\ m_2)$$

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

.
.
$a_k x \equiv b_k (mod\ m_k)$,
where$a_i, b_i, m_i (i = 1,2,..,k)$ are given integers. If m_i are pairwise coprime, then the system is solvable; or more precisely, elements of the congruence class modulo $m = m_1 m_2 \ldots . m_k$ satisfy all given congruences. This statement is called the Chinese remainder theorem.

Proof. By mathematical induction. First, consider a system with two congruences:

$$a_1 x \equiv b_1 (mod\ m_1)$$

$a_2 x \equiv b_2 (mod\ m_2)$.

The first congruence is solvable from assumption, hence there exists $x \equiv\ c_1((mod\ m_1))$ which satisfies the first congruence. Substitute this solution $x = c_1 + tm_1, t \in Z$, into the second congruence:

$$a_1(c_1 + m_1 t) \equiv b_2 (mod\ m_2)$$
$$(a_2 m_1)t \equiv (b_1 - a_2 c_1)(mod\ m_2)$$

Since m_1 and m_2 are coprime, then $\gcd(a_2 m_1, m_2) = gcd(a_2, m_2)$. From the theorem assumptions the second congruence is solvable too, therefore gcd. However, this already results in $\gcd(a_2 m_1, m_2)\ b_2$which is the condition for solvability of the congruence $(a_2 m_1)t \equiv (b_2 - a_2 c_1)(mod m_2)$.So we have t ≡ c2 (mod m2), which satisfies the second congruence. Then we can rewrite x as:

$x = c_1 + m_1(c_2 + sm_2) = (c_1 + c_2 m_1) + s(m_1 m_2)$ where $c_1, c_2, s \in Z$

Thus, the solution of the system of the two linear congruences is the whole congruence class $x \equiv e_1 (mod m_1 m_2)$, where $e_1 = c_1 + c_2 m_1$.

Now suppose the statement holds true for k = ν. Consider the system of ν +1 solvable linear congruences with pairwise coprime moduli $m_1 m_2 \ldots \ldots . m_v + 1$. The system of first ν congruences is solvable from the induction assumption, so we have $x \equiv e_v (mod m_1 m_2 \ldots \ldots . m_v)$ satisfying the first ν congruences. We have to find out if any element of this congruence class is also the solution of the last congruence. We solve the system of congruences:

$$x \equiv e_v (mod m_1 m_2 \ldots \ldots . m_v) a_v + 1_x \equiv b_v + 1 (mod m_v + 1)$$

Since $\gcd(m_v + 1, m_1, \ldots . , m_v) = 1$, then there exists the solution of this system of two congruences (by analogy to the first step of the proof). The Chinese remainder theorem says nothing about a case of the congruence system with non-coprime moduli. In this case, the system can be unsolvable, although individual congruences are solvable. But the system also can be solvable.

2.3. Euclidean Algorithm (EA) and Extended Euclidean Algorithm (EEA)

The study of algorithms for divisibility traces back to Euclid's Elements (circa 300 BCE), where the Euclidean Algorithm was first described as a method for finding the greatest common measure of two magnitudes. This work laid the foundation for computational number theory. Centuries later, Étienne Bézout (18th century) formalized the identity $ax + by = \gcd(a, b)$, which extended

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

Euclid's method to produce integer coefficients. This extension became known as the Extended Euclidean Algorithm (EEA). Later mathematicians such as Euler and Gauss refined these ideas, embedding them into the broader framework of modular arithmetic and congruences, which underpin modern cryptography and computational mathematics**.**

The Euclidean Algorithm is a systematic procedure for computing the greatest common divisor (GCD) of two integers.

$$gcd(a, b) = gcd(b, ab)$$

To solve first we divide a by b, obtaining quotient q and remainder r.

$$a = bq + r, \qquad 0 \leq r < b$$

Now replace a with b and b with r, then repeat this process until $r = 0$. The last non-zero remainder is $\gcd(a, b)$.

The Extended Euclidean Algorithm computes integers x, y such that:

$$ax + by = \gcd(a, b)$$

This is known as **Bézout's identity**. This produces Bézout coefficients useful in solving **linear Diophantine equations**. Also enables computation of **modular inverses**, essential for cryptography and the **Chinese Remainder Theorem (CRT)**. It also extends the Euclidean Algorithm from mere GCD computation to constructive problem-solving.

For solving this we perform the Euclidean Algorithm to compute successive remainders then back-substitute to express the GCD as a linear combination of a and b. further we track coefficients iteratively to obtain integers x and y.

Beyond pure number theory, the EEA has significant applications in computational mathematics and cryptography. Modular inverses, which are essential in RSA encryption and decryption, are computed directly using the EEA. Its efficiency makes it suitable for large-scale computations, where speed and reliability are critical. In coding theory, the EEA is used to solve congruences and error detection problems, further demonstrating its versatility.

2.3 Numerical examples

1. Solving a Linear Diophantine Equation with EEA

Problem

Solve the equation:

$$99x + 78y = 3$$

Step 1: Compute $gcd(99, 78)$ using Euclidean Algorithm

$$99 = 78 * 1 + 2$$
$$78 = 21 * 3 + 15$$
$$21 = 15 * 1 + 6$$
$$15 = 6 * 2 + 3$$
$$6 = 3 * 2 + 0 \rightarrow \gcd(99,78) = 3.$$

Step 2: Back-substitution (Extended Euclidean Algorithm)

From the steps above:

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

$$3 = 15 - 6 * 2$$
$$= 15 - (21 - 15 * 1) * 2 = 15 * 3 - 21 * 2$$
$$= (78 - 21 * 3) * 3 - 21 * 2 = 78 * 3 - 21 * 11$$
$$= 78 * 3 - (99 - 78 * 1) * 11 = 78 * 14 - 99 * 11$$

Thus:

$$3 = 99(-11) + 78(14)$$

Step 3: General Solution

Divide both coefficients by $\frac{3}{3} = 1$ (since RHS is 3). So, one solution is:

$$x = -11, y = 14$$

General solution:

$$x = -11 + \frac{78t}{3} = -11 + 26t, y = 14 - \frac{99t}{3} = 14 - 33t, \qquad t \in Z$$

2. Solving a System with CRT

Problem

Solve the system of congruences:

$$x \equiv 2(mod\ 3)$$
$$x \equiv 3(mod\ 5)$$
$$x \equiv 2(mod\ 7)$$

Step 1: Compute product of moduli

$$N = 3 * 5 * 7 = 105$$

Step 2: Compute partial moduli

$$N_1 = \frac{105}{3} = 35$$
$$N_2 = \frac{105}{5} = 21$$
$$N_3 = \frac{105}{7} = 15$$

Step 3: Find modular inverses (using EEA)

- Find inverse of 35(mod3): 35≡2(mod3). Inverse of 2 mod 3 is 2.
- Find inverse of 21(mod5): 21≡1(mod5). Inverse of 1 mod 5 is 1.
- Find inverse of 15(mod7): 15≡1(mod7). Inverse of 1 mod 7 is 1.

So:

$$M_1 = 2, M_2 = 1, M_3 = 1$$

Step 4: Construct solution

$$x = a_1 N_1 M_1 + a_2 N_2 M_2 + a_3 N_3 M_3$$
$$= 2 * 35 * 2 + 3 * 21 * 1 + 2 * 15 * 1$$

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

$$= 140 + 63 + 30 = 233$$

Step 5: Reduce modulo N

$$x \equiv 233(mod\ 105)$$
$$x \equiv 23(mod\ 105)$$

Final Answer

The solution is:

$$x = 23 + 105k, \quad k \in Z$$

Comparative Analysis

Comparison table for comparing extended Euclidean algorithm and Chinese remainder theorem for solving linear Diophantine equation as follow:

ATTRIBUTE	EXTENDED EUCLIDEAN ALGORITHM	CHINESE REMAINDER THEOREM
Core idea	Compute $\gcd(a,b)$ and Bézout coefficients s, t with $as + bt = \gcd(a,b)$	Reconstruct an integer from residues modulo pairwise coprime moduli; solution unique modulo product of moduli.
Typical input	Single linear equation $ax + by = c$.	System of congruences $x \equiv r_i(mod m_i)$ with pairwise coprime m_i.
Output	One particular integer solution and full parametric family $x = x_0 + \frac{bk}{g}, y = y_0 - \frac{ak}{g}$.	Unique solution modulo $M = \prod m_i$ equivalence class $x + kM$.
Time complexity (arithmetic steps)	We compute Bézout coefficients using the extended Euclidean algorithm; the algorithm requires $O(logmin(\|a\|,\|b\|))$ division steps and attains near quasi-linear bit complexity with fast integer arithmetic.	We reconstruct integers from residues using CRT; constructive algorithms' runtime depends on the number and bit-length of moduli, and balanced merging is used to limit growth of intermediate moduli and multi precision cost.
Bit complexity for large integers	Quasi-linear in input bit-length with fast integer	Bit-cost grows with product modulus M; iterative

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

	arithmetic; extended step adds negligible overhead.	combination accumulates cost and can be costly when M is large.
Strengths	Extremely efficient for two-variable Diophantine equations; returns Bézout coefficients and modular inverses.	Ideal for reconstructing integers from many residues; central to modular algorithms and distributed residue systems.
Weaknesses	Not directly suited to many-modulus systems without repeated reductions.	Requires pairwise coprime moduli or compatibility checks; cost and bit-size grow with number and size of moduli.

Use EEA when you have a single two-variable linear Diophantine equation $ax + by = c$ or need a modular inverse for one modulus. It is simple to implement, fast in practice, and scales well to very large integers. **Use CRT when** your problem is naturally expressed as multiple congruences with pairwise coprime moduli or when you must reconstruct an integer from residue data across independent moduli. Be mindful of the growth of the combined modulus M and choose reconstruction strategies (pairwise merging, balanced tree merging, or fast multi precision routines) that minimize repeated large-number operations.

Hybrid strategies: when solving systems that reduce to repeated two-variable steps, combine EEA for local inverses with CRT-style merging to keep intermediate sizes smaller. This often yields better practical performance than naive merging.

Conclusion

This comparative study examined the Extended Euclidean Algorithm (EEA) and the Chinese Remainder Theorem (CRT) as algorithmic tools for solving linear Diophantine problems. Both methods are fundamental in computational number theory and have complementary strengths depending on problem structure and scale.

Summary of findings

- **EEA** is the natural, direct method for a single two-variable equation $ax + by = c$. It computes $\gcd(a, b)$ and Bézout coefficients in a logarithmic number of Euclidean divisions, yields a particular integer solution, and parameterizes the full solution set.
- **CRT** is the canonical tool for reconstructing integers from multiple modular residues. Algorithmic reconstruction requires combining residues and managing the growth of the product modulus M; practical performance depends on merge strategy and multi precision arithmetic costs.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

Complexity comparison

- **EEA time complexity**: $O(logmin(|a|,|b|))$ Euclidean-division steps; with fast integer arithmetic the bit complexity is near quasi-linear in input bit-length.
- **CRT constructive cost**: depends on the number k of congruences and the total bit-length of moduli; naive sequential merging can incur large intermediate multi precision costs while balanced merging reduces overhead. In practice CRT cost grows with the bit-size of the combined modulus M.

Practical recommendations

- **Use EEA** for single linear Diophantine equations and single modular inverse computations because it is simple, efficient, and scales well to large integers.
- **Use CRT** when the problem is naturally expressed as multiple congruences or when reconstructing an integer from independent residues; implement balanced or tree merging to limit intermediate modulus growth.
- **Hybrid approach**: combine EEA for local inverses with CRT-style merging when solving systems that decompose into repeated two-variable steps to minimize multi precision work.

Limitations of the study

- Theoretical complexity statements assume standard multi precision arithmetic models and do not account for constant factors of specific libraries or hardware.
- Empirical performance depends on implementation details such as multiplication algorithms, memory management, and merge ordering; these factors were not exhaustively benchmarked here.

Directions for future work

- Provide empirical microbenchmarks across representative bit-lengths and residue counts to quantify practical breakpoints where CRT overhead outweighs EEA-based strategies.
- Investigate optimized multi-modulus reconstruction algorithms and parallel merging strategies to improve CRT scalability for very large M.
- Extend the comparison to constrained Diophantine problems (nonnegative solutions, bounded solutions) and to multi-variable integer systems where complexity and hardness results differ.

Final statement

For two-variable linear Diophantine equations the **Extended Euclidean Algorithm** is the preferred method for both theoretical efficiency and practical simplicity. The **Chinese Remainder Theorem** remains indispensable for multi-residue reconstruction and distributed residue systems, but careful algorithmic design is required to control bit-level costs. Choosing between EEA and CRT should therefore be guided by problem structure, the number and size of moduli, and implementation constraints.

REFERENCE

1. Euclid. (1956). The thirteen books of Euclid's Elements (T. L. Heath, Trans.). Dover Publications. (Original work published ca. 300 BCE)
2. Gauss, C. F. (1801/1966). Disquisitiones Arithmeticae (A. A. Clarke, Trans.). Yale University Press.
3. Hardy, G. H., & Wright, E. M. (2008). An introduction to the theory of numbers (6th ed.). Oxford University Press.
4. Ireland, K., & Rosen, M. (1990). A classical introduction to modern number theory (2nd ed.). Springer.
5. Burton, D. M. (2010). Elementary number theory (7th ed.). McGraw-Hill Education.
6. Koblitz, N. (1994). A course in number theory and cryptography (2nd ed.). Springer.
7. Knuth, D. E. (1997). The Art of Computer Programming, Volume 2: Seminumerical Algorithms (3rd ed.). Addison-Wesley.
8. Bach, E., & Shallit, J. (1996). Algorithmic number theory, Volume I: Efficient algorithms. MIT Press.
9. Cohen, H. (1993). A course in computational algebraic number theory. Springer.
10. Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of applied cryptography. CRC Press.