IT-Based Prime Number Computation and Cryptography Applications

Abd A. Hussein[1], Kilan M. Hussein[2], Syahida Che Dzul-kifli[3], Hassan Hadi Saleh[4], Mustafa A. Jawad[5], Omar Abdul Kareem Mahmood[5] and Mustafa N. Ghazal[5]

[1]*Department of Computer Science,College of Engineering, University of Diyala, 32001 Baqubah, Iraq*
[2]*Department of Computer Science, University of Diyala, 32001 Baqubah, Iraq*
[3]*Department of Mathematical Sciences, Faculty of Science and Technology, University Kebangsaan Malaysia, 43650 Bandar Baru Bangi, Malaysia*
[4]*College of Science, University of Diyala, Diyala, Iraq*
[5]*Department of Communication, College of Engineering, University of Diyala,32001 Baqubah, Iraq*
abdalihussein@uodiyala.edu.iq, kilan.m.h@uodiyala.edu.iq, syahida@ukm.edu.my, Hassan.hadi@uodiyala.edu.iq, mustafaawad98cv7@gmail.com, omar_abdulkareem@uodiyala.edu.iq, mustafa.nadhim@uodiyala.edu.iq

Keywords: Prime Numbers, Prime Numbers Formula, Programming, Odd Numbers.

Abstract: The ability to transmit and exchange vast amounts of information, as a result of the tremendous and widespread development in information and communications technology in our current era, has revealed a wonderful harmony and coherence between prime numbers, which are difficult to predict, decode, and distinguish, and the algorithms for encrypting that information and the methods for storing, encoding, and distributing it across communication networks. Today, prime numbers represent one of the cornerstones of mathematics, number theory, and modern applied sciences such as Information and Communications Technology (ICT), computer science, and Artificial Intelligence (AI). They are used in cryptographic systems such as RSA and Duffy-Hellman, and in calculating hash codes due to the difficulty of partitioning and factoring them into non-unit pairs, and their random occurrence in number sequences, making developing an equation to predict them a nearly impossible task. Although Fermat, Mersenne, and Euler used various formulas for calculating prime numbers that were applicable to limited applications, in this paper, researchers present a new mathematical approach to calculating and predicting prime numbers, using modern programming. This approach will open a new horizon for most applications across all fields.

1 INTRODUCTION

The tremendous development in information and communication technology, the proliferation of artificial intelligence, and the Internet of Things require a reconsideration of the approach to mathematical concepts and their application in the security of exchanged information and communication networks [1], [2]. Since the third century BC, Greek scientists were interested in the nature and properties of prime numbers, and a number of results emerged regarding these numbers, which Euclid explained in his book Elements [3], [4]. Odd numbers are divided into two main categories: Prime numbers, which can be divided only by themselves and one, while the second category of odd numbers is Non-Prime numbers, which have factors other than themselves and one, Pythagoras (in the sixth century BC) divided our positive integers into four groups: Odd, Even, Prime, and Composite numbers [5]. Also, two prime numbers are common when their greatest common factor is one. Furthermore, an even number results from the sum of at least two prime numbers [6], [7]. It's worth noting that prime numbers have no divisors other than themselves. They have baffled scientists in this field because they lack a clear pattern and appear randomly among infinite numbers. This makes it difficult to find an equation to predict them [8].

There are many intuitions to predict prime numbers such as the twin primes intuition, the Collatz intuition, Mersenne primes, Euler's guess, and others, which remain mere guesses, some of which have been proven incorrect and some of which have been rejected by advanced programs [9]. There are two

main types of prime number testing: 1) The deterministic test, which is the certainty that the number is prime or not, such as the Lehmer test, the elliptic curve, the experimental division test, and others. 2) The probabilistic test, which is finding a number that is very likely to be prime, such as the Miller, Fermat, Solovay tests, and other tests [10], [11]. The harmonization and adaptation of these conjectures and tests with modern and advanced programs and algorithms has not been implemented to keep pace with the huge developments in all scientific and practical fields. Rather, it has been limited to theoretical mathematics only [12], [13]. Furthermore, integrating modern and advanced programming into theoretical mathematics supports the ability to solve complex applied problems, contributes to creative thinking, and bridges the gap between theoretical frameworks and practical applications in our current digital age, and one of the most important and widespread applications of prime numbers is their use in cryptographic systems, such as the RSA encryption system, which is used to securely transmit encrypted information, making it difficult for unauthorized parties to hack or decipher it [14], [15].

In this study, we will present a new mathematical approach to identifying and characterizing prime numbers, supported by illustrative examples. We will then work on implementing this approach programmatically using a modern programming language to employ it in various application areas.

2 ADOPTING PRIME NUMBERS FORMULA (Abd's 3rd Intuition)

Our idea is to separate or extract the non-prime numbers from the odd numbers. For example, there are 25 non-prime numbers and 25 prime numbers in the first 100 natural numbers. To formulate the prime numbers equation, we follow these steps:

Step 1: Notation: Let's symbolize the mathematical terms as follows:

$$O(n) \equiv \text{odd numbers} \equiv 2n+1,$$
$$\mathrm{P}(n) \equiv \text{prime numbers},$$
$$NP(n) \equiv Non-\text{prime numbers},$$

where $n = 1,2,3,4$.

Step 2: Formulation: A general Abd's 3rd Intuition form for odd numbers as follows:

$$O(\mathrm{n}_i)=\begin{cases} \mathrm{NP(n)} \text{ If } (2\mathrm{ni}+1)|\mathrm{mod}(2\mathrm{j}+1) = 0, \\ \quad \text{For } (\mathrm{j} = 1,2,3,.,\mathrm{i}-1) \\ \mathrm{P(n)} \text{ If } (2\mathrm{ni}+1)|\mathrm{mod}(2\mathrm{j}+1) \neq 0, \\ \quad \text{For } (\mathrm{j} = 1,2,3,.,\mathrm{i}-1) \end{cases} \quad (1)$$

Where, (i) is the order of (n).

Therefore, it becomes clear that the specific formula for Abd's 3rd conjecture for prime and non-prime numbers from odd numbers is as follows:

$$\mathrm{P}(\mathrm{n}_i) = 2\,\mathrm{n}_i + 1 \quad \text{If } (2\mathrm{ni}+1)|\mathrm{mod}(2\mathrm{nj}+1) \neq 0, \text{ For } (\mathrm{j} = 1,2,3,\ldots\ldots,\mathrm{i}-1). \quad (2)$$

And

$$\mathrm{NP}(\mathrm{n}_i) = 2\,\mathrm{n}_i + 1 \quad \text{If}(2\mathrm{ni}+1)|\mathrm{mod}(2\mathrm{nj}+1) = 0, \text{ For } (\mathrm{j} = 1,2,3,\ldots\ldots,\mathrm{i}-1). \quad (3)$$

Step 3: Explanation: Abd's 3rd conjecture can be illustrated through the following examples:

Ex. 1): for n= 3 (i.e. i=3, j = {1, 2}, and O (3) = 7).

At j=1 → 7|mod (3) = 1, and j=2 →7|mod (5) = 2

∴ 7|mod{3, 5} ≠ 0, and (7) is prime.

Ex. 2): for n= 5 (i.e. i=5, j = {1, 2, 3, 4}, O (5) = 11).

At j=1 → 11|mod (3) = 2, j=2 → 11|mod (5) = 1, j=3 → 11|mod (7) = 4, and j=4 → 11|mod (9) = 2.

∴ 11|mod{3, 5, 7, 9} ≠ 0, and (11) is prime.

Ex. 3): for n= 8 (i.e. i=8, j = {1, 2, 3, 4, 5, 6, 7}, and O (8) = 17).

At j=1 → 17|mod (3) = 2, j=2 → 17|mod (5) = 2, j=3 → 17|mod (7) = 3, j=4 → 17|mod (9) = 8, j=5 → 17|mod (11) = 6, j=6 → 17|mod (13) = 4, j=7 → 17|mod (15) = 2.

∴ 17|mod{3, 5, 7, 9, 11, 13, 15} ≠ 0, and (17) is prime

Ex. 4): for n= 4 (i.e. i=4, j = {1, 2, 3}, and O (4) = 9)

At j=1 → 9|mod (3) = 0, we can be stopped because 9|mod (3) = 0 and (9) is non- prime, although at j=2 → 9|mod (5) = 4, and j=3 → 9|mod (7) = 2.

∴ 9|mod{3, 5, 7} = 0, and (9) is non- prime.

Ex. 5): for n= 7 (i.e. i=7, j = {1, 2, 3, 4, 5, 6}, and O(7)= 15).

At j=1 → 15|mod (3) = 0, and at j=2 → 15|mod (5) = 0. Therefore, we can be stopped because 15|mod(3) = 0 or 15|mod (5) = 0 and (15) is non-prime, although at j=3 → 15|mod (7) = 1, j=4 → 15|mod (9) = 6, j=5 → 15|mod (11) = 4, and at j=6 → 15|mod (13) = 2.

∴ 15|mod{3, 5, 7, 9, 11, 13} = 0, and (15) is non-prime

Note that: It is difficult to apply Abd's 3rd Intuition for large prime numbers manually without using programming.

3 PROGRAMMING ABD'S 3rd INTUITION

To utilize our intuition, popularize it, and apply it in various fields, it will be programmed this in the Python language [13]-[14] as shown below:

3.1 Mathematical Formula Representation

To evaluate the computational efficiency of the proposed approach, the algorithmic time and space complexities were analyzed for both single-number verification and prime generation within a finite range.

```
  Abd's 3rd Intuition Formula:
  For odd number O(ni) = 2ni + 1:
  IF (2ni + 1) MOD (2nj + 1) ≠ 0 for
all j = 1, 2, 3, ..., i-1
  THEN O(ni) is PRIME
  ELSE IF (2ni + 1) MOD (2nj + 1) = 0
for any j ∈ {1, 2, 3, ..., i-1}
  THEN O(ni) is NON-PRIME
  Where:
  - ni is the position index
  - j iterates through all smaller
position indices
  - MOD represents the modulo operation
```

3.2 Time Complexity Analysis

To evaluate the computational efficiency of the proposed intuition-based algorithm, its time and space complexities were analyzed for both single-number verification and full prime-number generation.

```
   Time Complexity: O(n²) for checking
number at position n
   Space Complexity: O(k) where k is
the number of divisors found
   For finding all primes up to limit
L:
    - Time Complexity: O(m³) where m is
the number of odd numbers ≤ L
    - Space Complexity: O(p) where p is
the number of primes found
```

3.3 Main Abd's 3rd Intuition Algorithm

We can illustrate our conjecture as shown in Figure 1 below:

Hence, this figure shows the programming of our intuition to facilitate the mathematical application of the above equations, especially if the odd numbers are very large, which makes it easy and highly flexible to employ it in other fields, such as information and encryption systems.

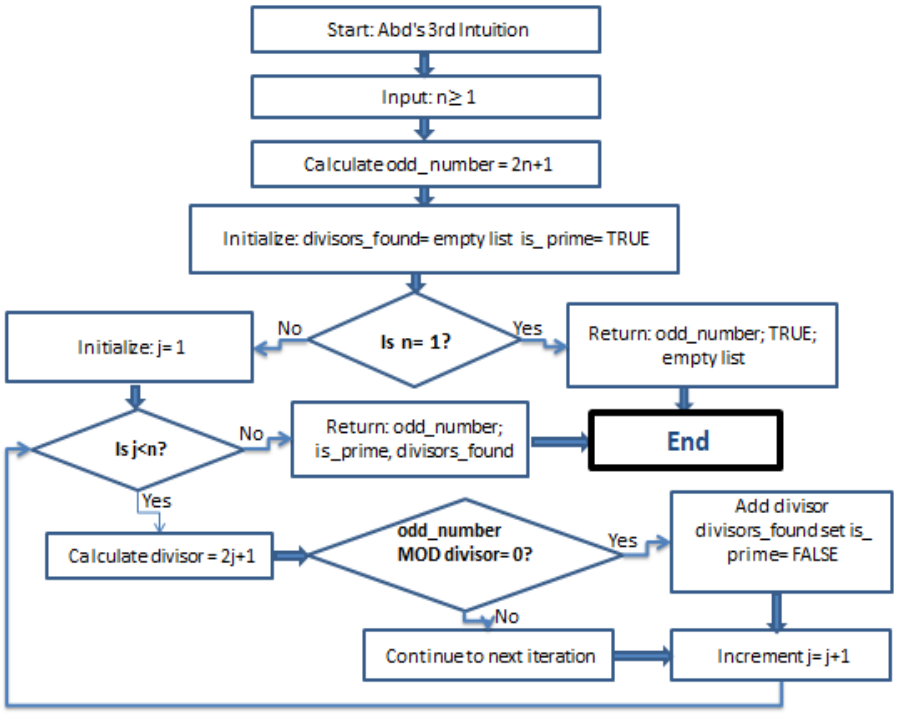

Figure 1: Main Abd's 3rd intuition algorithm.

4 MATHEMATICAL PROOF OF ABD'S 3RD INTUITION

The intuition can be formalized as a mathematical criterion for primality. Before proceeding to its algorithmic implementation, we provide a theorem and proof to show that the criterion is both necessary and sufficient for determining whether an odd number is prime.

Theorem (Abd's 3rd Intuition):

Let $O: \mathbb{N}_{\geq 1} \to 2\mathbb{Z} + 1$ enumerate the odd integers by:

$$O(i) = 2i + 1, \quad i = 1,2,3, \cdots.$$

For every $i \geq 1$, and odd number $O(i)$ is prime if it is not divisible by any smaller odd number.

$$O(i) \text{ is prime} \Leftrightarrow O(i) \bmod O(j) \neq 0 \text{ for all } j = 1,2, \cdots, i-1.$$

Proof:

($\Rightarrow$) Suppose $O(i)$ is prime. By definition, a prime has no positive divisors other than 1 and itself. Every $O(j)$ with $1 \leq j < i$ satisfies $1 < O(j) < O(i)$. Hence $O(j) \nmid O(i)$ for each such j, so $O(i) \bmod O(j) \neq 0$.

($\Leftarrow$) Conversely, suppose $O(i) \bmod O(j) \neq 0$ for all $j < i$. If $O(i)$ were composite, then by the Fundamental of Arithmetic it would have some divisor d with $1 < d < O(i)$. Because $O(i)$ is odd, any nontrivial divisor d must be odd, thus $d = O(j)$ for some $j < i$. This contradicts the assumption that no smaller odd $O(j)$ divides $O(i)$. Therefore $O(i)$ must be prime.

The theorem provides a complete characterization of prime numbers among the odd integers. However, checking divisibility against all smaller odd numbers may be computationally inefficient. To address this, we introduce a refinement that limits the range of divisibility tests without losing correctness. This leads to the following lemma.

4.1 Lemma (Square Root Bound)

In the test it suffices to restrict j to those with

$$O(j) \leq \sqrt{O(i)}.$$

Proof:

If a composite $n = O(i)$ has no divisor less than or equal to $\sqrt{n}$, then any factorization $n = ab$ would force $a > \sqrt{n}$ and $b > \sqrt{n}$, giving $ab > n$, a contradiction. Hence a composite number has a divisor less than or equal to $\sqrt{n}$. Since n is odd, so it equals some $O(j)$.

The lemma refines the theorem by showing that it is not necessary to test divisibility by every smaller odd number, but only by those up to the square root of the candidate. Beyond this optimization, the condition can also be reformulated in terms of the greatest common divisor, which provides a compact algebraic expression of Abd's 3rd Intuition. This leads to the following corollary.

4.2 Corollary (GCD Characterization of Abd's 3rd Intuition)

Let $O(i) = 2i + 1$ denote the i-th odd number, and let

$$P_{i-1} = \prod_{j=1}^{i-1} O(j).$$

Be the product of all smaller odd numbers. Then:

$$O(i) \text{ is prime} \Leftrightarrow \gcd(O(i), P_{i-1}) = 1.$$

Proof:

($\Rightarrow$) If $O(i)$ is prime, then it is not divisible by any smaller odd number $O(j)$ with $j < i$. Therefore, $O(i)$ shares no common factor with P_{i-1} and $\gcd(O(i), P_{i-1}) = 1$.

($\Leftarrow$) Conversely, if $\gcd(O(i), P_{i-1}) = 1$, then no smaller odd number $O(j)$ divides $O(i)$. By theorem, this implies that $O(i)$ is prime.

Together, the theorem, lemma, and corollary provide a rigorous foundation for Abd's 3rd Intuition. The theorem establishes the fundamental equivalence between primality and indivisibility by smaller odd numbers, the lemma refines this by reducing the divisibility checks to those up to the square root, and the corollary condenses the criterion into a single algebraic gcd condition. These results not only validate the intuition mathematically but also prepare the ground for its efficient implementation in programming environments.

5 CONCLUSIONS

This paper introduced "Abd's 3rd Intuition," a computationally oriented method for identifying prime numbers through structured analysis of odd integers. The proposed framework formalizes primality as a divisibility condition among ordered odd sequences and provides both mathematical and

algorithmic representations suitable for implementation in programming environments.

Theoretical analysis establishes the correctness of the approach via a primality criterion based on non-divisibility by smaller odd numbers, further refined by a square-root optimization and supported by a GCD-based formulation. These results ensure internal mathematical consistency while maintaining algorithmic interpretability.

Although the method is not intended to outperform advanced probabilistic primality tests in large-scale cryptographic applications, it offers a clear, deterministic, and educationally valuable alternative for understanding prime structure and algorithmic construction. Its main contribution lies in bridging number theory with programmable logic, highlighting potential applications in computational mathematics and introductory cryptographic modeling.

6 ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to Assistant Professor Ya'arub Mahmood Hamiedi, Professor of English Linguistics, for his valuable support, guidance, and assistance in improving the quality of this research.

REFERENCES

[1] Abd A. Hussein, "Survey Towards a Sustainable Information and Communication Technologies (ICT) in Iraq," Journal of Physics: Conference Series, vol. 1530, p. 012089, 2020, [Online]. Available: https://doi.org/10.1088/1742-6596/1530/1/012089.

[2] S. Petroccia, "From Mathematical Theory of Communication to Network Society: A Sociological Transformation," Società Mutamento Politica, vol. 14, no. 28, pp. 49-59, 2023, [Online]. Available: https://doi.org/10.36253/smp15012.

[3] A. T. Tessema, "Advanced Mathematical Formulas to Calculate Prime Numbers," Mathematics and Computer Science, vol. 6, no. 6, pp. 88-91, 2021, [Online]. Available: https://doi.org/10.11648/j.mcs.20210606.12.

[4] V. Koushik, "Prediction of Prime Numbers Using Prime Pattern Algorithm," Journal of Emerging Technologies and Innovative Research, vol. 12, no. 3, pp. 15-18, 2025, [Online]. Available: www.jetir.org.

[5] R. Kosova, F. Bushi, R. Kapçiu, F. Cullhaj, and A. M. Kosova, "A Review of Primality Tests and Algorithms: Engaging Students to Code for Mathematics," International Journal of Advanced Natural Sciences and Engineering Researches, vol. 8, no. 2, pp. 182-195, 2024, [Online]. Available: https://www.researchgate.net/publication/379147485.

[6] B. U. Zaman, "New Prime Number Theory," Annals of Mathematics and Physics, vol. 7, no. 2, pp. 158-161, 2024, [Online]. Available: https://dx.doi.org/10.17352/amp.000119.

[7] L. J. Goldstein, "A History of the Prime Number Theorem," The American Mathematical Monthly, vol. 80, no. 6, pp. 599-615, 1973.

[8] A. R. C. De Vas Gunasekara, A. A. C. A. Jayathilake, and A. A. I. Perera, "Survey on Prime Numbers," Elixir Applied Mathematics, vol. 88, pp. 36296-36301, 2015, [Online]. Available: www.elixirpublishers.com.

[9] L. J. Lander and T. R. Parkin, "Counterexample to Euler's Conjecture on Sums of Like Powers," Bulletin of the American Mathematical Society, vol. 72, no. 6, p. 1079, 1966.

[10] C. L. Duta, L. Gheorghe, and N. Tapus, "Framework for Evaluation and Comparison of Primality Testing Algorithms," in 20th International Conference on Control Systems and Computer Science, pp. 483-490, 2015.

[11] T. Desai, "Application of Prime Numbers in Computer Science and the Algorithms Used to Test the Primality of a Number," International Journal of Science and Research, vol. 4, no. 9, 2015.

[12] R. Kosova, R. Kapçiu, S. Hajrulla, and A. M. Kosova, "A Review of Mathematical Conjectures: Exploring Engaging Topics for University Mathematics Students," International Journal of Advanced Natural Sciences and Engineering Researches, vol. 7, no. 11, pp. 180-186, 2023, [Online]. Available: https://doi.org/10.59287/as-ijanser.581.

[13] R. Kuang and M. Barbeau, "Indistinguishability and Non-Deterministic Encryption of the Quantum Safe Multivariate Polynomial Public Key Cryptographic System," in IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1-5, 2021.

[14] A. A. Hussein, S. C. Dzul-Kifli, H. H. Saleh, K. M. Hussein, M. N. Ghazal, and D. A. Ali, "A New Approach with Software Implementation to Extend the Pythagorean Theorem in Multi-Dimensions," in Proceedings of the 13th International Conference on Applied Innovations in IT (ICAIIT 2025), vol. 13, no. 2, pp. 353-359, 2025.

[15] J. Burkhardt, I. Damgård, T. K. Frederiksen, S. Ghosh, and C. Orlandi, "Improved Distributed RSA Key Generation Using the Miller-Rabin Test," in Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, pp. 2501-2515, 2023.