A Novel Formula for Binomial Coefficients

Joseph M. Shunia

September 2023

Abstract

Computing binomial coefficients is a fundamental problem in combinatorics and number theory, with applications across mathematics. The typical approach relies on factorials, Gamma functions, or Pascal's triangle. In this paper, we introduce a novel formula for binomial coefficients: $\binom{n}{k} = \left\lfloor \frac{(1+2^n)^n}{2^{nk}} \right\rfloor \bmod 2^n \quad \text{for } n > 0 \text{ and } 0 \leq k \leq n$. We generalize the formula to multinomials, and discuss how it follows from a key property of polynomial interpolation rarely highlighted in textbooks. The simplicity of our arithmetic approach provides opportunities to develop more efficient algorithms and analyze computational complexity. Further, our formula establishes a new connection between binomial coefficients and modular arithmetic that may yield theoretical insights into their underlying mathematical foundations.

1 Introduction

Binomial coefficients, denoted as $\binom{n}{k}$, are ubiquitous in mathematics. They quantify the number of distinct ways to select k elements from a set of n elements, irrespective of the order of selection. These coefficients permeate various branches of mathematics, from the coefficients in the expansion of a binomial equation to the calculation of probabilities in statistical theory. Traditional methods to compute binomial coefficients include the use of factorials, Gamma functions, or Pascal's triangle [1]. Despite the prevalence of these methods, there is always room for fresh perspectives that can lead to novel insights and applications. In this paper, we hope to introduce such a perspective.

We present a new formula to compute binomial coefficients, given by:

$$\binom{n}{k} = \left\lfloor \frac{(1+2^n)^n}{2^{nk}} \right\rfloor \bmod 2^n, \quad \text{for } n > 0 \text{ and } 0 \leq k \leq n \tag{1}$$

This formula stems from a fascinating property of polynomials that has been rarely explored in mathematical literature. By connecting the computation of binomial coefficients with this polynomial property, we establish a formula that depends only on basic arithmetic operations.

2 Preliminaries and Definitions

2.1 Definition of Polynomial Function

We define $P(x)$ as the polynomial function:

$$P(x) = 1 + x \tag{2}$$

To denote $P(x)$ raised to the power of n, we write:

$$\begin{aligned} P(x)^n &= (1+x)^n \\ &= a_n x^n + a_{n-1} x^{n-1} + \ldots + a_k x^k + \ldots + a_1 x + a_0 \end{aligned}$$

We define $[x^k]P(x)^n$ to represent the coefficient of the k-th degree term in the polynomial expansion of $P(x)^n$, that is the value of a_k in $a_k x^k$.

2.2 Polynomial Expansion and Binomial Coefficients

If n is a non-negative integer, then by the Binomial Theorem [2] we have:

$$P(x)^n = (1+x)^n = \sum_{k=0}^{n} \binom{n}{k} x^k \tag{3}$$

As the coefficients of the terms in $P(x)$ are precisely the binomial coefficients for the n-th row of Pascal's Triangle [3], it is apparent:

$$[x^k]P(x)^n = \binom{n}{k} \tag{4}$$

2.3 Polynomial Interpolation

Polynomial interpolation is the problem of constructing a polynomial $P(x)$ that passes through a given set of data points $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$. That is, finding a polynomial $P(x)$ such that $P(x_i) = y_i$ for $i = 0, 1, \ldots, n$ [4, 5].

There are various methods to interpolate polynomials, including Lagrange interpolation, Newton polynomial interpolation, and methods using divided differences [4, 5]. A fundamental result is that there exists a unique polynomial of degree n or less that interpolates $n+1$ data points [4]. The key fact we require is that a polynomial is uniquely determined if we know its value at enough points. Specifically, if we know the value of $P(x)$ at $n+1$ distinct inputs, then we can recover its terms, and therefore the coefficients [6].

We will utilize a less known variant of this idea: that a polynomial can be unambiguously determined by evaluating just two carefully selected inputs [7, 8]. This forms the basis for our subsequent derivations of formula for binomial coefficients and its generalization to multinomials.

3 A Key Property of Polynomials

Theorem 1. If $P(x)$ is a polynomial with non-negative integer coefficients, it can be completely determined by the values $P(1)$ and $P(P(1))$.

Proof. Let $q = P(1)$, which gives the sum of the coefficients. Consider $P(P(1)) = P(q)$ expressed in base q; the digits correspond exactly to the coefficients of P. The only possible ambiguity arises if $P(q) = q \cdot N$ for some N, but since the coefficients sum to q, we deduce that $P = q \cdot x^{N-1}$ in this case [7]. □

Remark. This property appears to be under explored in the literature; we were unable to find any papers explicitly describing it. However, it has been the subject of some online discussion and at least one blog post [8]. The proof given above was first proposed by user ARupinski in a MathOverflow post [7] and has been slightly modified for the purposes of this paper.

The choice of $x = 1$ as an input for $P(x)$ is not particularly special; if we let x be any non-zero integer, the proof will hold. This revelation enables us to observe that the theorem and its proof can be generalized as follows:

Theorem 2. If $P(x)$ is a polynomial with non-negative integer coefficients, and $n \geq 1$ is an integer, then $P(x)$ can be completely determined by the values $P(n)$ and $P(P(n))$.

Proof. Let $q = P(n)$, which gives the sum of the coefficients of P evaluated at n. Consider $P(P(n)) = P(q)$ expressed in base q; the digits correspond exactly to the coefficients of P. The only possible ambiguity arises if $P(q) = q \cdot N$ for some N, but since the coefficients evaluated at $Q(n)$ sum to q, we deduce that $P = q \cdot x^{N-1}$ in this case. □

4 Derivation of the Formula

4.1 Application of the Polynomial Property

The value of $P(1)^n$ is simply the summation of the coefficients of $P(x)^n$. It is easy to see that:

$$P(1)^n = \sum_{k=0}^{n} \binom{n}{k} 1^k = (1+1)^n = 2^n \tag{5}$$

$$P(P(1)^n)^n = \sum_{k=0}^{n} \binom{n}{k} (P(1)^n)^k = (1 + P(1)^n)^n \tag{6}$$

$$= \sum_{k=0}^{n} \binom{n}{k} (2^n)^k = (1+2^n)^n \tag{7}$$

We can now use the results from Theorem 1 to recover the coefficients of $P(x)^n$.

We will achieve this by first encoding $P(P(1)^n)^n$ in base $P(1)^n$, and then recovering the coefficients of $P(x)^n$ from the encoded value.

4.2 Reconstruction of Polynomial Coefficients

Lemma 1. The formula to recover the k-th coefficient of $P(x)^n$ is given by:

$$[x^k]P(x)^n = \left\lfloor \frac{(1+P(1)^n)^n \bmod P(1)^{nk+n}}{P(1)^{nk}} \right\rfloor \tag{8}$$

Proof. Given an integer n and a base $b > 1$. Let $L = \lfloor \log_b n \rfloor$. Then the encoded representation of n in base b is the unique representation of n, such that [9]:

$$n = \sum_{k=0}^{L} b^k \left\lfloor \frac{n \bmod b^{k+1}}{b^k} \right\rfloor \tag{9}$$

By inserting $P(1)$ into the equation, we can see:

$$[x^k]P(x)^n = \left\lfloor \frac{P(P(1)^n)^n \bmod (P(1)^n)^{k+1}}{(P(1)^n)^k} \right\rfloor \tag{10}$$

$$= \left\lfloor \frac{(1+P(1)^n)^n \bmod P(1)^{nk+n}}{P(1)^{nk}} \right\rfloor \tag{11}$$

Hence, the lemma is proven. □

4.3 Main Theorem and Its Proof

Theorem 3.

$$\binom{n}{k} = \left\lfloor \frac{(1+2^n)^n}{2^{nk}} \right\rfloor \bmod 2^n, \quad \text{for } n > 0 \text{ and } 0 \le k \le n \tag{12}$$

Proof. **Step 1: Proof that the equation is valid:**
We begin with the identity we established in Lemma 1:

$$[x^k]P(x)^n = \left\lfloor \frac{(1+P(1)^n)^n \bmod P(1)^{nk+n}}{P(1)^{nk}} \right\rfloor \tag{13}$$

By substituting $[x^k]P(x)^n = \binom{n}{k}$ and $P(1) = 2$, we have:

$$\binom{n}{k} = \left\lfloor \frac{(1+2^n)^n \bmod 2^{nk+n}}{2^{nk}} \right\rfloor \tag{14}$$

Next, we define $X = (1+2^n)^n$, $Y = 2^{nk+n}$, $Z = 2^{nk}$. By replacement:

$$\binom{n}{k} = \left\lfloor \frac{X \bmod Y}{Z} \right\rfloor \tag{15}$$

To simplify the expression, we can utilize the following well-known identity for the mod operation [10]:

$$a \bmod b = a - b \left\lfloor \frac{a}{b} \right\rfloor \tag{16}$$

By application of the identity, the expression simplifies as follows:

$$\binom{n}{k} = \left\lfloor \frac{X \bmod Y}{Z} \right\rfloor \tag{17}$$

$$= \left\lfloor \frac{X - Y \left\lfloor \frac{X}{Y} \right\rfloor}{Z} \right\rfloor \tag{18}$$

$$= \left\lfloor \frac{X}{Z} - \frac{Y}{Z} \left\lfloor \frac{X}{Y} \right\rfloor \right\rfloor \tag{19}$$

Next, we replace X, Y, and Z with their respective values and simplify further:

$$\binom{n}{k} = \left\lfloor \frac{X}{Z} - \frac{Y}{Z} \left\lfloor \frac{X}{Y} \right\rfloor \right\rfloor \tag{20}$$

$$= \left\lfloor \frac{(1+2^n)^n}{2^{nk}} - \frac{2^{nk+n}}{2^{nk}} \left\lfloor \frac{(1+2^n)^n}{2^{nk+n}} \right\rfloor \right\rfloor \tag{21}$$

$$= \left\lfloor \frac{(1+2^n)^n}{2^{nk}} - 2^n \left\lfloor \frac{(1+2^n)^n}{2^{nk+n}} \right\rfloor \right\rfloor \tag{22}$$

$$= \left\lfloor \frac{(1+2^n)^n}{2^{nk}} \right\rfloor - 2^n \left\lfloor \frac{(1+2^n)^n}{2^{nk+n}} \right\rfloor \tag{23}$$

Finally, we again make use of the mod operation identity (16) to simplify:

$$\binom{n}{k} = \left\lfloor \frac{(1+2^n)^n}{2^{nk}} \right\rfloor \bmod 2^n \tag{24}$$

We have arrived at our formula.

Step 2: Proof that the equation holds only for $n > 0$ and $0 \leq k \leq n$:
To prove why the equation $\binom{n}{k} = \left\lfloor \frac{(1+2^n)^n}{2^{nk}} \right\rfloor \bmod 2^n$ holds only for $n > 0$ and $0 \leq k \leq n$, we consider the following cases:

1. $n \leq 0$: In this case, the modulus becomes 2^{-n} and the original equation does not make sense.
2. $k < 0$: In this case, we have $2^{-nk} = \frac{1}{2^{nk}}$ and the original expression becomes $2^{nk}(1+2^n)^n \bmod 2^n$, which is always equal to 0 since $2^n \mid 2^{nk}$.
3. $k > n$: In this case, $2^{nk} > (1+2^n)^n > 2^{nn}$, hence $\left\lfloor \frac{(1+2^n)^n}{2^{nk}} \right\rfloor = 0$. We note that $\binom{n}{k} = \binom{k}{n}$ here, which cannot equal 0.

Conclusion:
We have shown that our formula is valid and that it holds only for $n > 0$ and $0 \leq k \leq n$. Hence, the theorem is proven. □

4.4 Close Inspection of Our Formula

Upon initial inspection, it may not be obvious why our formula works:

$$\binom{n}{k} = \left\lfloor \frac{(1+2^n)^n}{2^{nk}} \right\rfloor \bmod 2^n, \quad \text{for } n > 0 \text{ and } 0 \leq k \leq n$$

However, by expanding the numerator $(1+2^n)^n$, we can see:

$$(1+2^n)^n = \sum_{j=0}^{n} \binom{n}{j} (2^n)^j \tag{25}$$

Dividing by the denominator 2^{nk} and taking the sum modulo 2^n gives:

$$\frac{(1+2^n)^n}{2^{nk}} \bmod 2^n = \sum_{j=0}^{n} \binom{n}{j} (2^n)^{j-k} \bmod 2^n \tag{26}$$

The terms with $j < k$ will sum to a value that is less than 1 and therefore, will vanish upon application of the floor operation. The terms with $j > k$ will be divisible by 2^n and therefore, vanish upon application of the mod operation. The only remaining term is $j = k$, which is equal to $\binom{n}{k}$ since $(2^n)^{j-k} = (2^n)^0 = 1$.

5 Generalization to Coefficients in the Multinomial Expansion of Univariate Unit Polynomials

Building upon the methodologies employed in our binomial coefficient derivation, we have discovered a generalized formula for calculating coefficients within the multinomial expansion of arbitrary degree univariate unit polynomials. That is, the coefficients in the expansion for polynomials of the form:

$$\left(\frac{x^D - 1}{x - 1}\right)^n = (1 + x + \ldots + x^{D-1})^n \tag{27}$$

The conventional approach to determine these coefficients utilizes conditional summations of multinomial coefficients $\binom{n}{k_0, k_1, \ldots, k_{D-1}}$, which represent the number of ways specific choices can be made to yield the term x^k [11]:

$$\binom{n}{k_0, k_1, \ldots, k_{D-1}} = \frac{n!}{k_0! k_1! \ldots k_{D-1}!} \tag{28}$$

In the context of our univariate unit polynomial, for each power of x in the expansion, the coefficient will come from all the combinations of powers that sum up to that specific power. Specifically, the coefficient of x^k in the expansion of our polynomial is [1]:

$$[x^k](1 + x + \ldots + x^{D-1})^n = \sum \binom{n}{k_0, k_1, \ldots, k_{D-1}} \tag{29}$$

Where the summation criteria are:

$$k_0 + k_1 + \ldots + k_{D-1} = n \tag{30}$$

$$0 \cdot k_0 + 1 \cdot k_1 + \ldots + (D-1) \cdot k_{D-1} = k \tag{31}$$

5.1 Generalized Formula and Its Proof

Theorem 4.

$$[x^k] \left(\frac{x^D - 1}{x - 1} \right)^n = \left\lfloor \left(\frac{D^{Dn} - 1}{D^{n+k} - D^k} \right)^n \right\rfloor \bmod D^n,$$
$$\text{for } n > 0 \text{ and } 0 \leq k \leq n \cdot (D-1)$$

Proof. **Step 1: Proof that the equation is valid:**
We define the polynomial function:

$$P_D(x)^n = \left(\frac{x^D - 1}{x - 1} \right)^n = (1 + x + \ldots + x^{D-1})^n \tag{32}$$

In this case, it is clear that:

$$P_D(1)^n = D^n \tag{33}$$

Therefore, we have:

$$P_D(P_D(1)^n)^n = (1 + D^n + \ldots + D^{n(D-1)})^n \tag{34}$$

Observe that the inner sum is equivalent to the summation of the powers of D^n from 0 to $(D-1)$. We note the following identity [12]:

$$\sum_{k=0}^{n-1} n^k = \frac{n^n - 1}{n - 1} \tag{35}$$

By substitution, we have:

$$P_D(P_D(1)^n)^n = \left(\sum_{k=0}^{D-1} D^{nk} \right)^n = \left(\frac{D^{Dn} - 1}{D^n - 1} \right)^n \tag{36}$$

In of our proof of Theorem 4, we established that:

$$[x^k]P(x)^n = \left\lfloor \frac{P(P(1)^n)^n}{P(1)^{nk}} \right\rfloor \bmod P(1)^n \tag{37}$$

By equivalence, we have:

$$[x^k]P_D(x)^n = \left\lfloor \frac{P_D(P_D(1)^n)^n}{P_D(1)^{nk}} \right\rfloor \bmod P_D(1)^n \tag{38}$$

Finally, we replace the values of $P_D(1)^n$ and $P_D(P_D(1)^n)^n$ and simplify to arrive at our original equation:

$$[x^k]\left(\frac{x^D-1}{x-1}\right)^n = \left\lfloor \left(\frac{D^{Dn}-1}{D^n-1}\right)^n \cdot D^{-nk} \right\rfloor \bmod D^n \tag{39}$$

$$= \left\lfloor \left(\frac{D^{Dn}-1}{D^n-1} \cdot D^{-k}\right)^n \right\rfloor \bmod D^n \tag{40}$$

$$= \left\lfloor \left(\frac{D^{Dn}-1}{D^k(D^n-1)}\right)^n \right\rfloor \bmod D^n \tag{41}$$

$$= \left\lfloor \left(\frac{D^{Dn}-1}{D^{n+k}-D^k}\right)^n \right\rfloor \bmod D^n \tag{42}$$

Step 2: Proof that the equation holds only for $n > 0$ and $0 \leq k \leq n \cdot (D-1)$:
To prove why the equation holds only for $n > 0$ and $0 \leq k \leq n \cdot (D-1)$, we consider the following cases in relation to:

$$[x^k]\left(\frac{x^D-1}{x-1}\right)^n = \left\lfloor \left(\frac{D^{Dn}-1}{D^{n+k}-D^k}\right)^n \right\rfloor \bmod D^n \tag{43}$$

$$= \left\lfloor \left(\frac{D^{Dn}-1}{D^n-1}\right)^n \cdot D^{-nk} \right\rfloor \bmod D^n \tag{44}$$

1. $n \leq 0$: In this case, the modulus becomes D^{-n} and the original equation does not make sense.
2. $k < 0$: In this case, we have $D^{-(-nk)} = D^{nk}$ and thus the original expression is always equal to 0, since $D^n \mid D^{nk}$.
3. $k > n \cdot (D-1)$: In this case, $D^{nk} > \left(\frac{D^{Dn}-1}{D^n-1}\right)^n > D^{nn}$, so the value of the expression is always equal to 0.

Conclusion:
We have shown that our formula is valid and that it holds only for $n > 0$ and $0 \leq k \leq n \cdot (D-1)$. Hence, the theorem is proven. □

6 Computational Complexity Analysis of Binomial Coefficient Formula

We now analyze the computational complexity associated with naively computing $\binom{n}{k}$ using our formula. The analysis is based on a direct, unoptimized approach. While this offers a foundational understanding, potential optimizations might not be addressed.

6.1 Time Complexity Analysis

Analyzing the time complexity for computing $\binom{n}{k}$ using our formula, we use the representation (14):

$$\binom{n}{k} = \left\lfloor \frac{(1+2^n)^n \bmod 2^{nk+n}}{2^{nk}} \right\rfloor$$

The procedure can be delineated as follows:

1. **Modular Exponentiation**: The main component affecting the time complexity is the modular exponentiation step. Using exponentiation by squaring, the time complexity for computing $a^b \mod c$ is $O(\log b \log c)$ [13]. Given c is exponential in n:

$$T_1(n) = O((nk+n)\log n)$$

2. **Bit Shift Operation**: The floored division by 2^{nk} translates to a bit shift, which has a linear time complexity in terms of bit count:

$$T_2(n) = O(nk)$$

3. **Combining Computations**: Taking both primary operations into account, the overall time complexity is:

$$T(n) = O((nk+n)\log n) + O(nk) = O((nk+n)\log n)$$

However, as n grows, the nk term becomes dominant:

$$T(n) = O(nk \log n)$$

6.2 Space Complexity Analysis

Analyzing the space complexity for computing $\binom{n}{k}$ requires us to consider the primary memory-consuming steps:

1. **Intermediate Results**: Exponentiation by squaring requires storage for intermediate results. For modulo 2^{nk+n} computations, the maximum number of bits needed is $nk+n$. Thus, space complexity is:

$$S_1(n) = O(nk+n)$$

2. **Bit Shift Operation**: The bit shift operation does not require proportional additional storage, leading to:

$$S_2(n) = O(1)$$

3. **Miscellaneous Variables**: Auxiliary variables, including n, k, and loop counters, occupy:

$$S_3(n) = O(1)$$

Combining these, the overall space complexity becomes:

$$S(n) = O(nk + n)$$

When n grows significantly, this simplifies to:

$$S(n) = O(nk)$$

6.3 Comparison to State of the Art

The most efficient algorithm for computing $\binom{n}{k}$ makes use of the factorial formula:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{45}$$

Notice that when multiplying $k!$ and $(n-k)!$, the product contains all integers from 1 to n in the sequence. Thus, many terms in the numerator $n!$ and the denominator $k!(n-k)!$ will cancel out, leaving only k terms in the numerator. This results in the simplified formula:

$$\binom{n}{k} = \prod_{j=1}^{k} \frac{n-j+1}{j} \tag{46}$$

In pseudocode:

```
Function Binomial(n, k)
    Set c to 1
    For j from 1 to k
        Set c to c * (n - j + 1)
        Set c to c / j
    End Loop
    Return c
End Function
```

The program above requires k multiplications and divisions to compute $\binom{n}{k}$. Harvey and van Der Heoven have given an algorithm for integer multiplication which has a time complexity of $O(k \log k)$ [14]. This gives the program an overall time complexity of:

$$T(n) = O(k(k \log k)) = O(k^2 \log k)$$

The program computes $\binom{n}{k}$ directly, so the space complexity is trivially:

$$S(n) = O(k)$$

The time complexity of our formula compares favorably when k is large. Considering the symmetry $\binom{n}{k} = \binom{n}{n-k}$, when $k = \left\lfloor \frac{n}{2} \right\rfloor$ and is therefore as large

as possible, the time complexity of both methods is $O(n^2 \log n)$. However, our formula requires $O(n^2)$ space to compute, whereas the factorial method requires only $O(n)$ space.

In practice, our algorithm may be more efficient to compute due to the reduction in overheads associated with incrementing counters and looping. Since our algorithm uses exponentiation by squaring, it requires only $O(\log n)$ multiplications compared to $O(k)$ multiplications for the factorial method. While our method has a higher memory footprint, modern systems often have large amounts of RAM available. Further analysis of the tradeoffs associated with both approaches is warranted.

7 Summary

In this paper, we have made two key contributions. First, we derived and proved a novel formula for computing binomial coefficients:

$$\binom{n}{k} = \left\lfloor \frac{(1+2^n)^n}{2^{nk}} \right\rfloor \bmod 2^n \quad \text{for } n > 0 \text{ and } 0 \le k \le n$$

This formula depends only on basic arithmetic operations and establishes an interesting connection between binomial coefficients and modular arithmetic. Second, we generalized this formula to find closed form expressions for the coefficients of terms in the multinomial expansion of arbitrary degree univariate unit polynomials. Specifically, we showed:

$$[x^k]\left(\frac{x^D-1}{x-1}\right)^n = \left\lfloor \left(\frac{D^{Dn}-1}{D^{n+k}-D^k}\right)^n \right\rfloor \bmod D^n,$$
$$\text{for } n > 0 \text{ and } 0 \le k \le n \cdot (D-1)$$

Our novel formulae stem from connecting binomial coefficients to a key, but rarely explored, property of polynomials: that they can be unambiguously determined by two judiciously chosen inputs.

Discoveries such as ours underscore the idea that profound insights can often emerge from concepts that, once discovered, seem simple and obvious, yet have remained overlooked. We hope that this reminds others of the vast potential for innovation that still lies latent even within well-studied fields of mathematics. By revisiting old problems with fresh perspectives, there is always the possibility of unearthing powerful new insights.

8 Future Directions

Future research could further explore the computational efficiency of the formulas and their behavior for large values of n, k, and D with a view to potential

optimizations. A deeper investigation into the polynomial property and its generalization could potentially reveal applications in computing values of other sequences. An examination of the relationship between modular arithmetic and binomial coefficients also presents a promising avenue for exploration.

References

[1] Richard A. Brualdi. *Introductory Combinatorics*. Pearson, 2010.

[2] James Stewart. *Calculus: Early Transcendentals*. Cengage Learning, 2007.

[3] Kenneth H Rosen. *Discrete Mathematics and Its Applications*. McGraw-Hill Science/Engineering/Math, 2011.

[4] Richard L Burden, J Douglas Faires, and Annette M Burden. *Numerical Analysis*. Brooks/Cole, Cengage Learning, Boston, MA, 9th edition, 2011.

[5] David R Kincaid and E Ward Cheney. *Numerical Analysis: Mathematics of Scientific Computing*. American Mathematical Soc., 2012.

[6] Elliott Ward Cheney and William Allan Light. *A Course in Approximation Theory*. American Mathematical Soc., 2013.

[7] Rupinski A. Mathoverflow: Application of polynomials with non-negative coefficients. `https://mathoverflow.net/questions/91827/`, 2012. Accessed: 2023-08-05.

[8] Cook J. Polynomial determined by two inputs. `https://www.johndcook.com/blog/2012/03/27/polynomial-trick/`, 2012. Accessed: 2023-08-05.

[9] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT press, 2009.

[10] Ivan Niven, Herbert S Zuckerman, and Hugh L Montgomery. *An Introduction to the Theory of Numbers*. John Wiley & Sons, 2008.

[11] Ronald L. Graham, Donald Ervin Knuth, and Oren Patashnik. *Concrete mathematics: a foundation for computer science*. Addison-Wesley Professional, 1994.

[12] David W. Wilson. Entry a023037 in the on-line encyclopedia of integer sequences. `https://oeis.org/A023037`, 2023. Accessed: 2023-08-07.

[13] Neal Koblitz. *A Course in Number Theory and Cryptography*. Springer-Verlag, 2nd edition, 1994.

[14] Joris van Der Hoeven David Harvey. Integer multiplication in time o(n log n). *Annals of Mathematics*, 2021.