

УДК 621.3.037.37

DOI <https://doi.org/10.36994/2788-5518-2025-01-09-11>

## ПЕРЕТВОРЕННЯ БІНОМІАЛЬНИХ ЧИСЕЛ У ДВІЙКОВІ

**Борисенко О. А.**, д.т.н., професор, Сумський державний університет, Суми, Україна. <https://orcid.org/0000-0001-7466-9135>, 5352008@ukr.net

**Хацько А. О.**, аспірант, Сумський державний університет, Суми, Україна. <https://orcid.org/0009-0004-6642-2991>, andrey5135vk@gmail.com

**Анотація.** У статті пропонується підхід до перетворення біноміальних чисел у двійкові. Запропоновано ефективний метод, що базується на поетапному декодуванні з одночасним застосуванням принципу повного перебору можливих комбінацій. Суть методу полягає в поступовому наближенні до результату шляхом синхронного аналізу значень біноміальних і двійкових чисел. Зокрема, перебір біноміальних чисел здійснюється в напрямі зменшення їх значень, тоді як відповідні двійкові еквіваленти формуються в напрямі зростання, доки біноміальне представлення не досягне свого мінімального порогового значення. Як наслідок, кінцевим результатом такого поетапного перетворення є точно підібране двійкове число, що відповідає заданому біноміальному значенню. Такий алгоритм перетворення вирізняється своєю логічною прозорістю, обчислювальною простотою та зручністю для практичної реалізації в цифрових системах.

Особливу увагу в роботі приділено надійності методу. Завдяки внутрішнім властивостям біноміальних чисел, алгоритм демонструє високу стійкість до зовнішніх завад і шумів, що можуть виникати в процесі передачі або зберігання інформації. Однією з вагомих переваг запропонованого підходу є його здатність до самокорекції: навіть за наявності незначних помилок у вхідних даних метод забезпечує точне відновлення двійкового еквівалента, що істотно підвищує загальну надійність системи.

Варто зазначити, що запропонований алгоритм не демонструє надзвичайно високої швидкодії порівняно з деякими іншими методами, однак це компенсується його простотою, технологічною доступністю й низькими вимогами до обчислювальних ресурсів. Така комбінація характеристик робить його надзвичайно зручним для вбудованих систем і пристроїв з обмеженим апаратним забезпеченням. Алгоритм легко інтегрується в наявні цифрові платформи, де перетворення біноміальних чисел у двійкові є важливим складником інформаційної обробки.

Запропонований метод є надійним, ефективним і технологічно доцільним рішенням для широкого спектра застосувань у сфері цифрових комунікацій, обробки сигналів, криптографії та інших галузях, де критично важливими є точність, стійкість до помилок і безперебійна робота в умовах дії зовнішніх завад. Завдяки поєднанню простоти реалізації, адаптивності та здатності до корекції, цей підхід має високий потенціал для практичного впровадження.

**Ключові слова:** двійкові числа, біноміальні числа, перебір, завадостійкість, біноміальні коефіцієнти.

## CONVERSION OF BINOMIAL NUMBERS INTO BINARY NUMBERS

**Oleksiy Borysenko**, D.Sc. (Tech.), Professor, Sumy State University, Sumy, Ukraine. <https://orcid.org/0000-0001-7466-9135>, 5352008@ukr.net

**Andrii Khatsko**, PhD Student, Sumy State University, Sumy, Ukraine. <https://orcid.org/0009-0004-6642-2991>, andrey5135vk@gmail.com

**Abstract.** The article proposes an approach to converting binomial numbers into binary form. An efficient method is introduced, based on step-by-step decoding combined with the simultaneous application of the principle of exhaustive enumeration of possible combinations. The core idea of the method involves gradually approximating the result through synchronized analysis of binomial and binary values. Specifically, the binomial numbers are iterated in decreasing order, while the corresponding binary equivalents are generated in increasing order, until the binomial representation reaches its minimum threshold. As a result, the final outcome of this step-by-step transformation is a precisely matched binary number corresponding to the given binomial value. This conversion algorithm is characterized by logical clarity, computational simplicity, and ease of practical implementation in digital systems.

Particular attention in the study is given to the reliability of the method. Due to the inherent properties of binomial numbers, the algorithm exhibits high resilience to external interferences and noise that may occur during the transmission or storage of information. One of the significant advantages of the proposed approach is its self-correcting capability: even in the presence of minor errors in the input data, the method

ensures accurate reconstruction of the binary equivalent, which significantly enhances the overall reliability of the system.

*It should be noted that the proposed algorithm does not demonstrate exceptionally high speed compared to some existing methods. However, this is more than compensated by its simplicity, technological accessibility, and low computational resource requirements. This combination of characteristics makes it highly suitable for embedded systems and devices with limited hardware capabilities. The algorithm can be easily integrated into existing digital platforms where binomial-to-binary conversion is an essential part of information processing.*

*The proposed method offers a reliable, efficient, and technologically feasible solution for a wide range of applications in digital communications, signal processing, cryptography, and other fields where accuracy, error resistance, and stable operation under external interference are critically important. Owing to its simplicity, adaptability, and error-correction capabilities, this approach holds significant potential for practical implementation.*

**Key words:** binary numbers, binomial numbers, enumeration, noise resistance, binomial coefficients.

## Вступ

Більшість сучасних цифрових систем обробки й передачі інформації будуються на основі двійкових чисел, які належать до природної двійкової системи числення. Вона залишається найпростішою завдяки своїй нульовій складності, однак двійкові числа без додаткового кодування не забезпечують можливості виявлення помилок. У зв'язку з цим розроблені складніші позиційні системи числення, зокрема біноміальні, які дають змогу підвищити завадостійкість цифрових систем.

Біноміальні числа, які генеруються в межах біноміальної системи числення, вирізняються високою завадостійкістю завдяки своїй структурі. Однак для практичної реалізації цієї системи часто виникає потреба в перетворенні біноміальних чисел у двійкові, що є важливим для багатьох спеціалізованих цифрових пристроїв, які вимагають підвищеної точності й стійкості до помилок.

Перетворення біноміальних чисел у двійкові, запропоноване в попередніх дослідженнях [1; 2], має значну складність у реалізації, особливо в апаратних рішеннях. Це може знижувати завадостійкість під час обробки даних. Із цієї причини на практиці частіше застосовуються простіші методи, такі як перебір комбінацій, що хоч і поступаються у швидкодії, але забезпечують стабільні результати й легкість інтеграції в системи передачі інформації.

Біноміальні системи числення, зокрема, мають особливе значення для побудови складних комбінаторних структур, які використовуються для кодування, створення рівноважних кодів та інших композицій. Однак для їх ефективного застосування необхідні ефективні перетворювачі кодів із біноміальних чисел у двійкові, що робить розвиток таких алгоритмів актуальним завданням у сучасній цифровій обробці інформації.

Існують дві основні задачі перетворення, пов'язані з біноміальними числами, для яких переборні алгоритми є ефективними рішеннями, забезпечуючи високу простоту й надійність. З одного боку, це задача перетворення двійкових чисел у біноміальні, а з іншого – зворотне перетворення біноміальних чисел у двійкові. Оскільки біноміальні числа генеруються за допомогою двійкових біноміальних систем числення, вони мають здатність перебирати, генерувати й нумерувати комбінаторні коди, забезпечуючи перешкодостійкість, що робить їх корисними для передачі, зберігання й обробки інформації в цифрових системах.

## Виконання досліджень

Основним завданням роботи є розроблення переборного алгоритму для ефективного перетворення біноміальних чисел у двійкові. Це особливо актуально для практичних систем, де після перетворення двійкових чисел у біноміальні часто виникає потреба виконати зворотне перетворення для коректної роботи цифрових пристроїв. Алгоритм повинен забезпечувати мінімальну складність і надійність, що є критично важливим для стабільної роботи цифрових систем.

**Поняття двійкових чисел.** Двійкові числа – це числа, які представлені у двійковій системі числення, де використовуються тільки дві цифри: 0 і 1. Двійкова система є основою для роботи сучасних комп'ютерів і цифрових пристроїв. Кожна цифра у двійковому числі називається біт, і вона має вагу, яка визначається позицією цього біта в числі.

**Поняття біноміальних чисел.** Існує багато різновидностей біноміальних чисел [1]. У роботі розглядаються найбільш прості двійкові біноміальні числа.

Двійковими біноміальними числами називаються послідовності розрядів цифр 0 і 1 з вагами у вигляді біноміальних коефіцієнтів, які дають змогу перетворювати їх у номери природних чисел і зворотно.

**Нерівномірні біноміальні числа.** Існують нерівномірні й отримані на їх основі рівномірні біноміальні числа [1]. Нерівномірним біноміальним числом називається число з двійковим алфавітом довжини  $n$ , у якому кількість одиниць не перевищує значення  $k$ , а кількість нулів  $n - k$  [1]. Вони мають різну довжину й тому можуть ефективно використовуватися для зберігання та передачі інформації. У цьому випадку біноміальні числа меншої довжини кодують більш імовірні повідомлення, а довші –

менш імовірні. Другим їх завданням є те, що вони створюють рівномірні біноміальні числа, які можуть використовуватися для побудови завадостійких цифрових пристроїв [1].

Діапазон нерівномірних і, відповідно, рівномірних біноміальних чисел визначається біноміальним коефіцієнтом:

$$P = C_{n+1}^k = C_{n+1}^{n-k} = \frac{(n+1)!}{k!(n-k+1)!} \quad (1)$$

У таблиці 1 наведені величини діапазонів  $P$  нерівномірних біноміальних чисел для  $n = 1, 2, \dots, 20$  і  $k = 2$ .

Таблиця 1  
Діапазони біноміальних чисел  $P$

| n | P  | n  | P  | n  | P   | n  | P   |
|---|----|----|----|----|-----|----|-----|
| 1 | 1  | 6  | 21 | 11 | 66  | 16 | 136 |
| 2 | 3  | 7  | 28 | 12 | 78  | 17 | 153 |
| 3 | 6  | 8  | 36 | 13 | 91  | 18 | 171 |
| 4 | 10 | 9  | 45 | 14 | 105 | 19 | 190 |
| 5 | 15 | 10 | 55 | 15 | 120 | 20 | 210 |

Алгоритм побудови нерівномірних біноміальних чисел у зростаючому порядку з будь-якого біноміального числа складається з таких кроків:

1. В останній розряд справа нерівномірного біноміального числа, у якому знаходиться 0, записується 1 і підраховується кількість одиниць у ньому.

2. Якщо кількість одиниць у біноміальному числі менша за  $k$ , то справа від отриманої одиниці записується нуль і далі відбувається повернення до кроку 1.

3. Якщо кількість одиниць у біноміальному числі дорівнює  $k$ , а нулів перед ними немає, то – кінець.

4. Якщо кількість одиниць у біноміальному числі буде дорівнювати  $k$  і перед останніми з них буде стояти хоча б один нуль, то цей 0 повинен бути замінений на 1, а в молодші стосовно нього справа розряди записуються нулі до того часу, поки їх кількість у числі не стане дорівнювати  $n - k$ .

5. Повернення до кроку 1.

У таблиці 2 для  $n = 4$  наведені 10 можливих нерівномірних біноміальних чисел зліва з кількістю одиниць  $k = 3$ , а справа – з  $k = 2$ .

Таблиця 2  
Нерівномірні біноміальні числа з  $n = 4$  і  $k = 3, 2$

| № | Бін. числа | № | Бін. числа | № | Бін. числа | № | Бін. числа |
|---|------------|---|------------|---|------------|---|------------|
| 0 | 00         | 5 | 1010       | 0 | 000        | 5 | 011        |
| 1 | 010        | 6 | 1011       | 1 | 0010       | 6 | 1000       |
| 2 | 0110       | 7 | 1100       | 2 | 0011       | 7 | 1001       |
| 3 | 0111       | 8 | 1101       | 3 | 0100       | 8 | 101        |
| 4 | 100        | 9 | 111        | 4 | 0101       | 9 | 11         |

**Рівномірні біноміальні числа.** Окрім нерівномірних біноміальних чисел, при побудові біноміальних цифрових пристроїв використовуються рівномірні біноміальні числа. Вони, на відміну від нерівномірних чисел, мають постійну довжину (див. таблицю 3), яка містить  $n$  розрядів. Їх особливістю є те, що кількість нулів у них перед останньою одиницею справа не може бути більшою  $n - k$ , що призводить до їх завадостійкості [1]. Тобто побудовані на основі рівномірних біноміальних чисел цифрові пристрої чи програми будуть завадостійкі.

**Рішення задачі.** Кількісний еквівалент числа  $n$  – розрядної двійкової  $k$ -біноміальної системи числення:

$$A_i = (a_{j-1}, a_{j-2}, \dots, a_0)$$

$$i = 0, 1, \dots, P - 1$$

визначається числовою функцією:

$$A_i = a_{j-1}C_{n-1}^{k-q_j} + \dots + a_jC_{n-j+1}^{k-q_{j+1}} + \dots + a_0C_{n-j}^{k-q_1} \quad (2)$$

Вона має дві системи обмежень:  
або

$$\begin{cases} q_0 = k \\ k \leq j \leq n-1 \end{cases}$$

або

$$\begin{cases} a_0 = 1 \\ n-k = j - q_0 \\ 0 \leq q_0 \leq k-1 \\ a_0 = 0, \end{cases}$$

де  $q_0$  – кількість одиниць;

$j$  – кількість розрядів;

$l = 0, 1, \dots, j-1$  – порядковий номер розряду;

Сума одиничних значень цифр:

$$q_{l+1} = \sum_{\gamma=l+1}^{j-1} a_{\gamma},$$

максимальне число при цьому:

$$1C_{n-1}^{k-q_j} + 1C_{n-2}^{k-q_{j-1}} + \dots + 1C_{n-j}^{k-q_1} = C_n^k - 1 = P - 1$$

Відповідно, кількість біноміальних чисел:

$$P = C_n^k$$

Як випливає з формули (1), ці числа нерівномірні, тобто можуть відрізнятися своєю довжиною. Щоб отримати відповідний біноміальному числу двійковий номер, треба підставити це число у формулу (2).

У таблиці 1 ці числа з їх кількісними еквівалентами і двійковими номерами показані для  $k = 4$ ,  $n = 6$ . Їх кількість, відповідно, дорівнює:

$$P = C_n^k = C_6^4 = 15.$$

Таблиця 3

Нерівномірні біноміальні числа та їх кількісні еквіваленти

| №  | Біноміальне число | Кількісний еквівалент                                                           | №    |
|----|-------------------|---------------------------------------------------------------------------------|------|
| 0  | 00                | $0 \cdot C_5^4 + 0 \cdot C_4^4$                                                 | 0000 |
| 1  | 010               | $0 \cdot C_5^4 + 1 \cdot C_4^4 + 0 \cdot C_3^3$                                 | 0001 |
| 2  | 0110              | $0 \cdot C_5^4 + 1 \cdot C_4^4 + 1 \cdot C_3^3 + 0 \cdot C_2^2$                 | 0010 |
| 3  | 01110             | $0 \cdot C_5^4 + 1 \cdot C_4^4 + 1 \cdot C_3^3 + 1 \cdot C_2^2 + 0 \cdot C_1^1$ | 0011 |
| 4  | 01111             | $0 \cdot C_5^4 + 1 \cdot C_4^4 + 1 \cdot C_3^3 + 1 \cdot C_2^2 + 1 \cdot C_1^1$ | 0100 |
| 5  | 100               | $1 \cdot C_5^4 + 0 \cdot C_4^4 + 0 \cdot C_3^3$                                 | 0101 |
| 6  | 1010              | $1 \cdot C_5^4 + 0 \cdot C_4^4 + 1 \cdot C_3^3 + 0 \cdot C_2^2$                 | 0110 |
| 7  | 10110             | $1 \cdot C_5^4 + 0 \cdot C_4^4 + 1 \cdot C_3^3 + 1 \cdot C_2^2 + 0 \cdot C_1^1$ | 0111 |
| 8  | 10111             | $1 \cdot C_5^4 + 0 \cdot C_4^4 + 1 \cdot C_3^3 + 1 \cdot C_2^2 + 1 \cdot C_1^1$ | 1000 |
| 9  | 1100              | $1 \cdot C_5^4 + 1 \cdot C_4^4 + 0 \cdot C_3^3 + 0 \cdot C_2^2$                 | 1001 |
| 10 | 11010             | $1 \cdot C_5^4 + 1 \cdot C_4^4 + 0 \cdot C_3^3 + 1 \cdot C_2^2 + 0 \cdot C_1^1$ | 1010 |
| 11 | 11011             | $1 \cdot C_5^4 + 1 \cdot C_4^4 + 0 \cdot C_3^3 + 1 \cdot C_2^2 + 1 \cdot C_1^1$ | 1011 |
| 12 | 11100             | $1 \cdot C_5^4 + 1 \cdot C_4^4 + 1 \cdot C_3^3 + 0 \cdot C_2^2 + 0 \cdot C_1^1$ | 1100 |
| 13 | 11101             | $1 \cdot C_5^4 + 1 \cdot C_4^4 + 1 \cdot C_3^3 + 0 \cdot C_2^2 + 1 \cdot C_1^1$ | 1101 |
| 14 | 1111              | $1 \cdot C_5^4 + 1 \cdot C_4^4 + 1 \cdot C_3^3 + 1 \cdot C_2^2$                 | 1111 |

Відповідно до (2), алгоритм перетворення нерівномірних біноміальних чисел у двійкові числа складається з такого:

- 3 їх підстановки у формулу 1.
- Обчислення біноміальних коефіцієнтів у розрядах, де стоять 1.
- Підсумовування отриманих значень біноміальних коефіцієнтів.

Результат буде номером нерівномірного біноміального числа.

Доповнюючи в таблиці 3 розряди справа нулями, можемо отримати рівномірні біноміальні числа, як це показано в таблиці 4. Їх кількісний еквівалент залишиться незмінним.

Таблиця 4  
Рівномірні біноміальні числа та їх кількісні еквіваленти

| №  | Біноміальне число | Рівн. біном. число | Кількісний еквівалент                                                           | №    |
|----|-------------------|--------------------|---------------------------------------------------------------------------------|------|
| 0  | 00                | 00000              | $0 \cdot C_5^4 + 0 \cdot C_4^4$                                                 | 0000 |
| 1  | 010               | 01000              | $0 \cdot C_5^4 + 1 \cdot C_4^4 + 0 \cdot C_3^3$                                 | 0001 |
| 2  | 0110              | 01100              | $0 \cdot C_5^4 + 1 \cdot C_4^4 + 1 \cdot C_3^3 + 0 \cdot C_2^2$                 | 0010 |
| 3  | 01110             | 01110              | $0 \cdot C_5^4 + 1 \cdot C_4^4 + 1 \cdot C_3^3 + 1 \cdot C_2^2 + 0 \cdot C_1^1$ | 0011 |
| 4  | 01111             | 01111              | $0 \cdot C_5^4 + 1 \cdot C_4^4 + 1 \cdot C_3^3 + 1 \cdot C_2^2 + 1 \cdot C_1^1$ | 0100 |
| 5  | 100               | 10000              | $1 \cdot C_5^4 + 0 \cdot C_4^4 + 0 \cdot C_3^3$                                 | 0101 |
| 6  | 1010              | 10100              | $1 \cdot C_5^4 + 0 \cdot C_4^4 + 1 \cdot C_3^3 + 0 \cdot C_2^2$                 | 0110 |
| 7  | 10110             | 10110              | $1 \cdot C_5^4 + 0 \cdot C_4^4 + 1 \cdot C_3^3 + 1 \cdot C_2^2 + 0 \cdot C_1^1$ | 0111 |
| 8  | 10111             | 10111              | $1 \cdot C_5^4 + 0 \cdot C_4^4 + 1 \cdot C_3^3 + 1 \cdot C_2^2 + 1 \cdot C_1^1$ | 1000 |
| 9  | 1100              | 11000              | $1 \cdot C_5^4 + 1 \cdot C_4^4 + 0 \cdot C_3^3 + 0 \cdot C_2^2$                 | 1001 |
| 10 | 11010             | 11010              | $1 \cdot C_5^4 + 1 \cdot C_4^4 + 0 \cdot C_3^3 + 1 \cdot C_2^2 + 0 \cdot C_1^1$ | 1010 |
| 11 | 11011             | 11011              | $1 \cdot C_5^4 + 1 \cdot C_4^4 + 0 \cdot C_3^3 + 1 \cdot C_2^2 + 1 \cdot C_1^1$ | 1011 |
| 12 | 11100             | 11100              | $1 \cdot C_5^4 + 1 \cdot C_4^4 + 1 \cdot C_3^3 + 0 \cdot C_2^2 + 0 \cdot C_1^1$ | 1100 |
| 13 | 11101             | 11101              | $1 \cdot C_5^4 + 1 \cdot C_4^4 + 1 \cdot C_3^3 + 0 \cdot C_2^2 + 1 \cdot C_1^1$ | 1101 |
| 14 | 1111              | 11110              | $1 \cdot C_5^4 + 1 \cdot C_4^4 + 1 \cdot C_3^3 + 1 \cdot C_2^2$                 | 1111 |

Особливістю нерівномірних біноміальних чисел є те, що вони можуть бути використані для швидкої передачі інформації. Однак вони не можуть виявляти помилки, які виникають при їх передачі. Цю задачу успішно виконують рівномірні біноміальні числа. Ознакою помилки в них буде поява в числі кількості одиниць більшої за  $k$  або поява перед останньою 1 зліва більше  $n - k - 1$ . Таких помилкових біноміальних чисел для  $n = 5$  буде 17. Таким числом, наприклад, буде число 10010 або число 11111.

**Методи перетворення біноміальних чисел у двійкові.** На практиці для використання двійкових чисел потрібні перетворювачі біноміальних чисел у двійкові, оскільки цифрові системи, як правило, працюють із двійковими значеннями. Загальний вигляд процесу перетворення біноміального числа у двійкове число зображено на рис. 1.

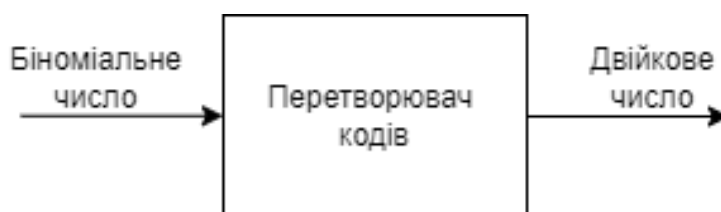


Рис. 1. Перетворення біноміальних чисел у двійкові

У ньому біноміальне число подається на перетворювач кодів, який залежно від обраного алгоритму виконує перетворення з різною швидкістю.

**Найбільш швидкодіючим** буде табличний метод перетворення, за якого в пам'яті перетворювача зберігається готовий результат, а вхідне біноміальне число виступає як адреса. За цією адресою можна отримати відповідне двійкове число. Однак недоліком цього алгоритму є потреба у великій ємності пам'яті, що обмежує кількість біноміальних чисел, які можуть зберігатися.



**Менш швидкодіючим**, але з достатньою швидкодією й без великих вимог до пам'яті є алгоритм, який використовує числову функцію для біноміальних чисел. Проте його реалізація може бути досить складною, особливо в схемному вигляді:

$$C(n, k) = \frac{n!}{k!(n-k)!}.$$

Тобто алгоритм отримує значення  $n$  і  $k$ , обчислює відповідне значення біноміального коефіцієнта, а потім перетворює його у двійкову форму. Це дає змогу уникнути використання великої пам'яті, оскільки результат обчислюється безпосередньо під час запиту й не потрібно зберігати таблицю всіх можливих результатів.

Однак реалізація цього методу є складнішою, особливо якщо алгоритм має бути апаратно реалізованим, наприклад, у цифровій схемі. Потрібно враховувати точність розрахунків факторіалів і правильно обробляти великі числа, що може ускладнити розробку й вплинути на загальну швидкодію.

**Найменш швидкодіючим**, але простим і надійним методом є переборний алгоритм, який використовує програмно або апаратно реалізований біноміальний лічильник. Цей лічильник підсумовує одиниці, а разом із ним працює двійковий лічильник, який зменшує своє значення. Система порівняння перевіряє, чи досягнув двійковий лічильник нульового стану.

Процес роботи переборного алгоритму:

**Ініціалізація** – біноміальний лічильник починається зі значення, яке потрібно перетворити, а двійковий лічильник ініціалізується на 0.

**Перебір** – біноміальний лічильник послідовно підсумовує одиниці, тоді як двійковий лічильник зменшує своє значення.

**Порівняння** – процес триває, поки двійковий лічильник не досягне нульового значення, після чого схема порівняння видає сигнал.

**Виведення результату** – коли двійковий лічильник переходить у нульовий стан, значення двійкового лічильника буде дорівнювати результату перетворення біноміального числа.

#### **Висновок**

Таким чином, надані в статті переборні алгоритми перетворення біноміальних чисел у двійкові є найбільш простими й надійними. Незважаючи на певний недолік у вигляді малої швидкодії, для багатьох практичних застосувань ця швидкодія є цілком достатньою, що робить ці алгоритми ефективними в таких випадках. Ефективність перетворення зростає завдяки врахуванню завадостійкості біноміальних чисел, які за своєю природою володіють властивістю стійкості до помилок, що дає їм змогу забезпечувати надійність навіть в умовах перешкод.

#### **Література**

1. Борисенко О. А. Дискретна математика : підручник. Суми, 2019. 255 с.
2. Бардачов Ю., Соколова Н., Ходаков В. Дискретна математика. Київ : Вища шк., 2002. 287 с.
3. Бондаренко М., Білоус Н., Руткас А. Комп'ютерна дискретна математика. Київ : Компанія СМІТ, 2004. 480 с.
4. Капітонова Ю., Кривий С., Лєтичевський О. Основи дискретної математики. Київ : Наук. думка, 2002. 580 с.

#### **References**

1. Borysenko, O. A. (2019) Dyskretna matematyka : pidruchnyk [Discrete Mathematics], 255 s. [in Ukrainian].
2. Bardachov, Yu., Sokolova, N., Khodakov, V. (2002) Dyskretna matematyka [Discrete Mathematic]. 287 s. [in Ukrainian].
3. Bondarenko, M., Bilous, N., Rutkas, A. (2004) Komp'yuterna dyskretna matematyka [Computer Discrete Mathematics]. 480 s. [in Ukrainian].
4. Kapitonova, Yu., Kryvyi, S., Letychevskiy, O. (2002) Osnovy dyskretnoi matematyky [Fundamentals of Discrete Mathematics]. 580 s. [in Ukrainian].

*Дата першого надходження рукопису до видання: 15.05.2025*  
*Дата прийнятого до друку рукопису після рецензування: 10.06.2025*  
*Дата публікації: 25.07.2025*