ЗМІСТ

|  |  |
| --- | --- |
| ВСТУП |  |
| 1. Випадкові і псевдовипадкові числа……………………..…………..…….… | 4 |
| 1. Критерії випадковості…………………………………..…………………… | 6 |
| * 1. Методика тестування NIST STS………………………………………… | 7 |
| * 1. Опис статистичних тестів……………………………………………….. | 9 |
| 1. Генератори псевдовипадкових чисел……………….……………………… | 15 |
| 3.1 Лінійний конгруентний генератор……………………………………… | 15 |
| 3.2 Blum Blum Shub генератор…….…………………………………………. | 17 |
| 3.3 Генератори на основі лінійних регістрів зсуву із зворотним зв’язком . | 19 |
| 1. Аналіз псевдовипадкових чисел…………………………………………… | 21 |
| 4.1 Реалізація конгруентного генератора та аналіз згенерованих послідовностей………………………………………………………………………… | 21 |
| 4.2 Реалізація Blum Blum Shub генератор та аналіз згенерованих послідовностей……………………………………………………………………… | 24 |
| 4.3 Реалізація генератора на основі лінійних регістрів зсуву із зворотним зв’язком та аналіз згенерованих послідовностей………………………….. | 27 |
| ВИСНОВКИ |  |
| ЛІТЕРАТУРА |  |
| ДОДАТОК А |  |

ВСТУП

Згідно принципу Керкгоффза стійкість криптографічної системи має базуватися на секретності лише ключа шифрування. Крім того, одна із ознак стійкого шифру, розглянутих Клодом Шенноном, вимагає використання випадкового ключа. Таким чином, можна зробити висновок, що одним із факторів секретності ключа є його випадковість. Випадкові числа в криптографії використовуються не тільки для формування секретних ключів, але і для синтезу надійних криптографічних перетворень (блоки заміни, випадкові перестановки). Так, у шифрах заміни, випадкові числа можуть використовуватись, як замінюючи елементи знаків відкритого тексту.

До появи ефективних способів генерування випадкових чисел, для їх отримання використовували відомі випадкові процеси (підкидання гральних кісток або монети, витягання кулі з урни, тощо). Одним із найпростіших механічних приладів для отримання випадкових чисел є рулетка. Саме тому математичні методи, які використовують випадкові числа, називають «методами Монте-Карло» за найменуванням міста у князівстві Монако, де знаходяться відомі будинки ігр. В першій половині XX століття було розроблено спеціальні машини, які формували випадкові числа механічним шляхом. У 1955 році компанія RAND Corporation опублікувала таблиці з мільйоном випадкових цифр, отриманих такою машиною [1].

Для отримання випадкових чисел застосовують також апаратні засоби. Однак необхідність додаткового апаратного забезпечення та неможливість повторної генерації випадкової послідовності не завжди задовольняє вимогам прикладних додатків. З цієї причини були розроблені математичні алгоритми, які дозволили б програмно створити послідовність чисел, властивості якої будуть схожі на властивості випадкових чисел. Такі послідовності називаються [псевдовипадковими](http://uk.wikipedia.org/wiki/%D0%9F%D1%81%D0%B5%D0%B2%D0%B4%D0%BE%D0%B2%D0%B8%D0%BF%D0%B0%D0%B4%D0%BA%D0%BE%D0%B2%D1%96_%D1%87%D0%B8%D1%81%D0%BB%D0%B0).

Коректне застосування таких методів та обгрунтування їх властивостей вперше запропонував Джон фон Нейман у 1951 р. [2] Його метод формування псевдо-випадкових чисел полягав в наступному: *n*-розрядне число підносилось до квадрата і з нього вибиралися середні *n* цифр [3]. Описаний метод був недосконалий: послідовності вироджувалися після першої нульової комбінації (квадрат 0 завжди 0) або зациклювалися з коротким періодом. Пізніше було запропоновано багато надійніших та ефективніших алгоритмів отримання псевдовипадкових чисел.

1 Випадкові і псевдовипадкові числа

Випадкова послідовність (або числа) - це штучно отримана послідовність числа реалізацій випадкової величини із заданим законом розподілу. Випадкові числа використовуються при досліджені та оптимізації складних стохастичних систем методом статистичного моделювання з допомогою електронних обчислюваних машин.

Джерела справжніх випадкових чисел знайти важко. Фізичні [шуми](http://znaimo.com.ua/%D0%A8%D1%83%D0%BC), такі як детектори подій іонізуючої [радіації](http://znaimo.com.ua/%D0%A0%D0%B0%D0%B4%D1%96%D0%B0%D1%86%D1%96%D1%8F), [дробовий шум](http://znaimo.com.ua/%D0%94%D1%80%D0%BE%D0%B1%D0%BE%D0%B2%D0%B8%D0%B9_%D1%88%D1%83%D0%BC) у резисторі або [космічне випромінювання](http://znaimo.com.ua/%D0%9A%D0%BE%D1%81%D0%BC%D1%96%D1%87%D0%BD%D0%B5_%D0%B2%D0%B8%D0%BF%D1%80%D0%BE%D0%BC%D1%96%D0%BD%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) можуть бути такими джерелами. Для генерування послідовностей випадкових послідовностей використовують апаратні генератори псевдовипадкових чисел. Апаратний генератор випадкових чисел - пристрій, який генерує послідовності випадкових чисел на основі вимірювання параметрів протікання фізичного процесу. Робота таких пристроїв часто заснована на процесах рівня елементарних часток, таких як тепловий шум, фотоелектричний ефект, інші квантові явища. Ці процеси, в теорії, абсолютно непередбачувані. Апаратні генератори випадкових чисел, засновані на квантових процесах, зазвичай складаються з спеціального підсилювача і перетворювача. Спочатку підсилюються дуже слабкі сигнали, що одержуються в результаті протікання фізичних явищ, до прийнятних розмірів, які трансформуються перетворювачем до цифрового вигляду.

Апаратні генератори випадкових чисел відносно повільні і можуть виробляти зміщені послідовності (коли певна послідовність чисел повторюється частіше). Використання подібних генераторів залежить від потреб конкретної предметної області й від пристрою самого генератора. Однак застосування таких пристроїв в додатках мережевої безпеки є непрактичним.

Альтернативою є створення деякого набору з великої кількості випадкових чисел і опублікування його в деякому [словнику](http://znaimo.com.ua/%D0%A1%D0%BB%D0%BE%D0%B2%D0%BD%D0%B8%D0%BA). Хоча дані набори дійсно забезпечують статистичну випадковість, вони не можуть використовуватися у криптографічних додатках, так як криптоаналітик має змогу отримати копію словника.

Генератор псевдовипадкових чисел включений до складу деяких сучасних [процесорів](http://znaimo.com.ua/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80) (наприклад архітектури [VIA](http://znaimo.com.ua/X86) C3) та більшості операційних систем.

Криптографічні додатки використовують для генерації випадкових чисел особливі алгоритми. Ці алгоритми є детермінованими і тому генерують послідовність чисел, яка теоретично не може бути статистично випадковою. У той же час, якщо вибрати хороший алгоритм, отримана чисельна послідовність буде проходити більшість тестів на випадковість. Такі числа називають псевдовипадковими числами.

Псевдовипадкові числа — числа, що отримуються в результаті роботи детермінованого алгоритму, але мають властивості, що неможливо відрізнити від властивостей випадкових чисел. Негативною особливістю псевдовипадкових чисел (як імітаційної моделі випадкових чисел) є періодичність отриманої послідовності. Реалізація статистичних моделей за допомогою обчислювальних машин передбачає можливість отримання у достатній мірі випадкових показників (що є нетривіальним завданням, оскільки електронно-обчислювальна машина працює на основі чітко прописаної логіки та детерміністичного підходу), а також імітацію законів розподілу, що часто використовуються на практиці.

Обов'язковим етапом генерування псевдовипадкових чисел є перевірка періодичності й випадковості отриманої послідовності та її відповідності заданому закону розподілу.

2 Критерії випадковості

Статистичні тести NIST, описані у стандарті NIST SP 800-22 - пакет статистичних тестів, розроблений Лабораторією інформаційних технологій (англ. Information Technology Laboratory), що є головною дослідницькою організацією Національного інституту стандартів і технологій (англ.. The National Institute of Standards and Technology NIST) [4].

Набір тестів NIST STS був запропонований у ході проведення конкурсу на новий національний стандарт США блокового шифрування. Цей набір використався для досліджень статистичних властивостей кандидатів на новий блоковий шифр. На сьогодні методика тестування, що запропонована NIST є найбільш поширеною у розробників криптографічних засобів захисту інформації.

Порядок тестування окремої двійкової послідовності *S* має наступний вигляд [11]:

* Висувається нульова гіпотеза  - припущення про те, що дана двійкова послідовність *S* випадкова.
* По послідовності *S* розраховується статистика тесту *с*(*S*)
* Із використанням спеціальної функції і статистики тесту розраховується значення імовірності 
* Значення імовірності *Р* порівнюється із рівнем значущості , [0.001, 0.01], Якщо , то гіпотеза  приймається. У противному випадку приймається альтернативна гіпотеза.

Пакет містить у собі 15 статистичних тестів. Але фактично, в залежності від вхідних параметрів обчислюється 188 значень імовірності Р, які можна розглядати як результат роботи окремих тестів.

Таким чином у результаті тестування двійкової послідовності формується вектор значень імовірності *P* ={*P1, P2,…, P188*}*.* Аналіз складових *Pi* даного вектору дозволяє вказати на конкретні дефекти випадковості послідовності, що тестується [12].

2.1 Методика тестування NIST STS

Генератор повинен формувати двійкову послідовність S ={s1, s2, …, sn}, si ∈ {0,1} довільної довжини n.

Для фіксованого значення n формують множину з m двійкових послідовностей:

|  |  |
| --- | --- |
| *S*1 *=*{*s*1, *s*2, …, *sn*};  *S*2 *=*{*s*1, *s*2, …, *sn*};  ………………….  *Sm =*{*s*1, *s*2, …, *sn*}. | (2.1) |

Таким чином, для тестування необхідно сформувати вибірку об’ємом *N* = *m* × *n*.

Кожну послідовність перевіряють з використанням пакету NIST STS. У результаті формується статистичний портрет генератора виду

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № теста *j*  № пос-  ти *i* | 1 | 2 | … | *q* |
| *S*1 | *P*1,1 | *P*1,2 |  | *P*1,*q* |
| *S*2 | *P*2,1 | *P*2,2 |  | *P*2,*q* |
| : | : |  |  |  |
| *Sm* | *Pm*,1 | *Pm*,2 |  | *Pm*,*q* |



*Статистичний портрет генератора* є матриця розмірністю *m*×*q*, де *m* – кількість двійкових послідовностей, що перевіряються, а *q* – кількість статистичних тестів, що використовуються для тестування кожної послідовності. Елементи матриці  представляють собою значення ймовірності, яка отримана у результаті тестування *i*-ої послідовності *j*-им тестом.

За отриманим статистичним портретом визначають долю послідовностей, що пройшли кожний статистичний тест. Для цього задають рівень значимості α ∈ [0,001, 0,01] та здійснюють підрахунок значень ймовірностей , що перевищують встановлений рівень α для кожного з *q* тестів, тобто визначають коефіцієнт [4]

. (2.2)

У результаті формується вектор коефіцієнтів **R** = {*r*1, *r*2, …, *rq*}, елементи якого характеризують, у відсотках, проходження послідовності *Si* всіх статистичних тестів.

**Правило 1.** *Вважається, що генератор G пройшов тестування по j-му тесту, якщо значення коефіцієнту* *rj знаходиться в межах и довірчого інтервалу* [*rmax*, *rmin*]. Границі довірчого інтервалу визначаються відповідно виразу[13]:

 (2.3)

де .

Здійснюється статистичний аналіз статистичного портрету. Отримані значення імовірностей *Pij* повинні підкорятися рівноймовірному закону розподілу на інтервалі [0, 1] [13]. Для кожного вектору-стовпцю статистичного портрету будується гістограма частостей *Fk* попаданій значень *Pij* у кожний з *k* = 1, 2, …, 10 під інтервалів, на які розбитий інтервал [0, 1]. Рівноймовірність розподілу значень ймовірностей *Pij* перевіряється з використанням критерію χ2. Для цього розраховується статистика виду:

 (2.4)

яка підкоряється розподілу χ2 с дев’ятьмами ступенями волі.

**Правило 2***. Вважається, що генератор G пройшов тестування за j-им тестом, якщо виконується умова .*

Прикінцеве рішення приймають у відповідності з правилом: *вважається, що генератор G пройшов статистичне тестування пакетом NIST STS, якщо значення коефіцієнтів rj* *для всіх * з*находяться в межах довірчого інтервалу* [*rmax*, *rmin*] та *виконується умова* ** для всіх **.

2.2 Опис статистичних тестів

* + 1. Частотний побітовий тест

Суть даного тесту полягає у визначенні співвідношення між нулями і одиницями у всій двійковій послідовності. Мета – з'ясувати, чи справді число нулів і одиниць в послідовності приблизно однакові, як це можна було б припустити у випадку істинно випадкової бінарної послідовності. Тест оцінює, наскільки близька частка одиниць до 0,5. Таким чином, число нулів і одиниць має бути приблизно однаковим. Якщо обчислена в ході тесту значення імовірності p < 0,01, то дана двійкова послідовність не є істинно випадковою.

* + 1. Частотний блочний тест

Суть тесту – визначення частки одиниць всередині блоку довжиною *m* біт. Мета – з'ясувати чи дійсно частота повторення одиниць в блоці довжиною *m* біт приблизно дорівнює *m*/2, як можна було б припустити в разі абсолютно випадкової послідовності. Обчислене в ході тесту значення імовірності *p* повинно бути не менше 0,01. В іншому випадку (*p* < 0,01), двійкова послідовність не носить істинно випадковий характер. Якщо прийняти *m* = 1, даний тест переходить в тест № 1.

* + 1. Тест на послідовність однакових бітів

Суть полягає в підрахунку повного числа рядів у вихідній послідовності, де під словом «ряд» мається на увазі безперервна підпослідовність однакових бітів. Ряд довжиною *k* біт складається з *k* абсолютно ідентичних бітів, починається і закінчується з біта, що містить протилежне значення. Мета даного тесту – зробити висновок про те, чи дійсно кількість рядів, що складаються з одиниць і нулів з ​​різними довжинами, відповідає їх кількості у довільній послідовності. Зокрема, визначається швидко або повільно чергуються одиниці і нулі у вихідній послідовності. Якщо обчислене в ході тесту значення ймовірності *p* < 0,01, то дана двійкова послідовність не є істинно випадковою.

* + 1. Тест на найдовшу послідовність одиниць в блоці

У даному тесті визначається найдовший ряд одиниць всередині блоку довжиною m біт. Мета – з'ясувати чи дійсно довжина такого ряду відповідає очікуванням довжини найдовшого ряду одиниць в разі абсолютно випадкової послідовності. Якщо вирахувати в ході тесту значення імовірності *p* < 0,01 допускається, що вихідна послідовність не є випадковою. В іншому випадку, робиться висновок про її випадковості. Слід зауважити, що з припущення про приблизно однаковою частотою появи одиниць і нулів (тест № 1) випливає, що точно такі ж результати даного тесту будуть отримані при розгляді самого довгого ряду нулів. Тому вимірювання можна проводити тільки з одиницями.

* + 1. Тест рангів бінарних матриць

Тут проводиться розрахунок рангів непересічних підматриць, побудованих з вихідної двійковій послідовності. Метою цього тесту є перевірка на лінійну залежність підрядів фіксованої довжини документів першої послідовності. У разі, якщо обчислене в ході тесту значення ймовірності *p* < 0,01, робиться висновок про невипадковий характер вхідної послідовності біт. В іншому випадку, вважаємо її абсолютно випадковою. Даний тест також присутній в пакеті DIEHARD [14].

* + 1. Спектральний тест

Суть тесту полягає в оцінці висоти піків дискретного перетворення Фур'є вихідної послідовності. Мета – виявлення періодичних властивостей вхідної послідовності, наприклад, близько розташованих один до одного повторюваних ділянок. Тим самим це явно демонструє відхилення від випадкового характеру досліджуваної послідовності. Ідея полягає в тому, щоб число піків,що перевищують порогове значення у 95% по амплітуді, було значно більше 5%. Якщо обчислена в ході тесту значення ймовірності *p* < 0,01, то дана двійкова послідовність не є випадковою.

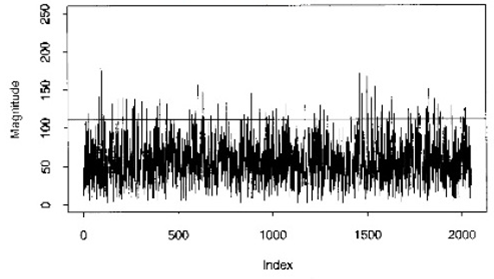


Рисунок 1 - Відсутність періодичності

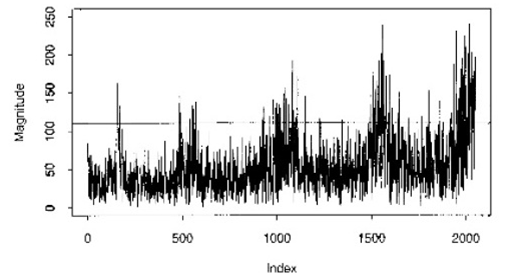


Рисунок 2 - Присутня періодичність

* + 1. Тест на збіг шаблонів, що не перекриваються

У даному тесті підраховується кількість наперед визначених шаблонів, знайдених у вихідній послідовності. Мета - виявити генератори випадкових або псевдовипадкових чисел, що формують занадто часто задані неперіодичні шаблони. Як і у тесті № 8 на збіг шаблонів, що перекриваються для пошуку конкретних шаблонів довжиною *m* біт використовується вікно також довжиною *m* біт. Якщо шаблон не виявлений, вікно зміщується на один біт. Якщо ж шаблон знайдений, вікно переміщається на біт, наступний за знайденим шаблоном, і пошук триває далі. Обчислене в ході тесту значення ймовірності *p* повинно бути не менше 0,01. В іншому випадку (*p* <0,01), двійкова послідовність не є випадковою.

* + 1. Тест на збіг шаблонів, що перекриваються

Суть даного тесту полягає в підрахунку кількості заздалегідь визначених шаблонів, знайдених у вихідній послідовності. Як і в тесті № 7 на збіг шаблонів, що не перекриваються для пошуку конкретних шаблонів довжиною *m* біт використовується вікно також довжиною *m* біт. Сам пошук проводиться аналогічним чином. Якщо шаблон не виявлений, вікно зміщується на один біт. Різниця між цим тестом і тестом № 7 полягає лише в тому, що якщо шаблон знайдений, вікно переміщається тільки на біт вперед, після чого пошук продовжується далі. Обчислене в ході тесту значення імовірності *p* повинно бути не менше 0,01. В іншому випадку (*p* <0,01), двійкова послідовність не є випадковою.

* + 1. Універсальний статистичний тест Маурера

Тут визначається число біт між однаковими шаблонами у вихідній послідовності (міра, що має безпосереднє відношення до довжини стислої послідовності). Мета тесту - з'ясувати чи може дана послідовність бути значно стиснута без втрат інформації. У випадку, якщо це можливо зробити, то вона не є істинно випадковою. У ході тесту обчислюється значення ймовірності *p*. Якщо *p* <0,01, то вважається, що вихідна послідовність не є випадковою. В іншому випадку, робиться висновок щодо її випадковості.

* + 1. Тест на лінійну складність

В основі тесту лежить принцип роботи лінійного регістра зсуву зі зворотним зв'язком (англ. Linear Feedback Shift Register, LFSR) [6]. Мета - з'ясувати чи є вхідна послідовність досить складною для того, щоб вважатися абсолютно випадковою. Абсолютно випадкові послідовності характеризуються довгими лінійними регістрами зсуву зі зворотним зв'язком. Якщо ж такий регістр занадто короткий, то передбачається, що послідовність не є в повній мірі випадковою. У ході тесту обчислюється значення ймовірності *p*. Якщо *p* <0,01, то вважається, що вихідна послідовність не є випадковою.

* + 1. Тест на періодичність

Даний тест полягає в підрахунку частоти всіх можливих перекривань шаблонів довжини *m* біт на протязі вихідної послідовності бітів. Метою є визначення чи дійсно кількість появ 2*т* перекриваються шаблонів довжиною *m* біт, приблизно таке ж як у випадку абсолютно випадковою вхідної послідовності біт. Остання, як відомо, володіє одноманітністю, тобто кожен шаблон довжиною m біт з'являється в послідовності з однаковою ймовірністю. Якщо обчислена в ході тесту значення ймовірності *p* <0,01, то дана двійкова послідовність не є абсолютно випадковою. В іншому випадку, вона носить випадковий характер. Варто відзначити, що при *m* = 1 тест на періодичність переходить в частотний побітовий тест (№ 1).

* + 1. Тест приблизною ентропії

Як і в тесті на періодичність в даному тесті акцент робиться на підрахунку частоти всіх можливих перекривання шаблонів довжини *m* біт на протязі вихідної послідовності бітів. Мета тесту - порівняти частоти перекривання двох послідовних блоків вихідної послідовності з довжинами *m* і *m* +1 з частотами перекривання аналогічних блоків в абсолютно випадкової послідовності. Обчислюване в ході тесту значення ймовірності *p* повинно бути не менше 0,01. В іншому випадку (*p* <0,01), двійкова послідовність не є випадковою.

* + 1. Тест кумулятивних сум

Тест полягає в максимальному відхиленні (від нуля) при довільному обході, обумовленим кумулятивною сумою заданих (-1, +1) цифр у послідовності. Мета даного тесту - визначити чи є кумулятивна сума часткових послідовностей, що виникають у вхідний послідовності, занадто великою або занадто маленькою у порівнянні з очікуваним поведінкою такої суми для абсолютно випадковою вхідної послідовності. Таким чином, кумулятивна сума може розглядатися як довільний обхід. Для випадкової послідовності відхилення від довільного обходу повинні бути поблизу нуля. Для деяких типів послідовностей, які не є в повній мірі випадковими подібні відхилення від нуля при довільному обході будуть досить суттєвими. Якщо обчислена в ході тесту значення ймовірності *p* <0,01, то вхідна двійкова послідовність не є випадковою. В іншому випадку, вона носить випадковий характер.

* + 1. Тест на довільні відхилення

Суть даного тесту полягає в підрахунку числа циклів, що мають строго *k* відвідувань при довільному обході кумулятивної суми. Довільний обхід кумулятивної суми починається з часткових сум після послідовності (0,1), перекладеної в відповідну послідовність (-1, +1). Цикл довільного обходу складається з серії кроків одиничної довжини, чинених у випадковому порядку. Крім того такий обхід починається і закінчується на одному і тому ж елементі. Мета даного тесту – визначити чи відрізняється число відвідувань певного стану всередині циклу від аналогічного числа у випадку абсолютно випадковою вхідної послідовності. Фактично даний тест є набір, що складається з восьми тестів, проведених для кожного з восьми станів циклу: -4, -3, -2, -1 і +1, +2, +3, +4. У кожному такому тесті приймається рішення про ступінь випадковості вихідної послідовності у відповідності з наступним правилом: якщо обчислене в ході тесту значення ймовірності *p* <0,01, то вхідна двійкова послідовність не є випадковою. В іншому випадку, вона носить випадковий характер.

* + 1. Інший тест на довільні відхилення

У цьому тесті підраховується загальна кількість відвідувань певного стану при довільному обході кумулятивної суми. Метою є визначення відхилень від очікуваного числа відвідувань різних станів при довільному обході. В дійсності цей тест складається з 18 тестів, правдивих для кожного стану: -9, -8, ..., -1 і +1, +2, ..., +9. На кожному етапі робиться висновок про випадковість вхідної послідовності. Якщо обчислена в ході тесту значення ймовірності *p* <0,01, то вхідна двійкова послідовність не є випадковою. В іншому випадку, вона носить випадковий характер.

3 ГЕНЕРАТОРИ ПСЕВДОВИПАДКОВИХ ЧИСЕЛ

3.1 Лінійний конгруентний генератор

Конгруентний генератор псевдовипадкових чисел застосовується лише для моделювання випадкових процесів і не є криптостійким. Такий генератор реалізовано у багатьох компіляторях, наприклад Borland C/C++, Borland Delphi, Virtual Pascal, Microsoft Visual/Quick C/C++, Open Watcom, Digital Mars, Metrowerks, IBM VisualAge C/C++, Apple CarbonLib, Java's java.util.Random та інші.

Метод конгруентного генератору полягає в обчисленні членів лінійної рекурентної послідовності по модулю деякого натурального числа *m*, що задається наступною формулою:

|  |  |
| --- | --- |
| , | (3.1) |

де *а* і *с –* деякі цілочислові коефіцієнти. Структурна схема конгруентного генератора зображена на рис.3.

випадкова

послідовність

а

c

m

*X0*

зворотній зв’язок

Рисунок 3 – Схема генерації псевдовипадкової послідовності

Послідовність, що отримується залежить від вибору стартового числа *Х0* і при різних його значеннях отримуються різні послідовності випадкових чисел. В той час, більшість властивостей цієї послідовності визначаються вибором коефіцієнтів в формулі і не залежать від вибору стартового числа [15].

Послідовність чисел, що генерується конгруентним генератором, періодична з періодом, що не перевищує *m.*  При цьому довжина періоду дорівнює *m*  лише тоді, коли:

1. НСД(*c,m*) = 1 (тобто коли *c,m*  взаємно прості);
2. *а*-1 кратне *р*  для усіх простих дільників *р*  числа *m;*
3. *а*-1 кратне 4, якщо *m* кратне 4.

Статистичні властивості послідовності випадкових чисел, що отримується, повністю визначається вибором коефіцієнтів *а* і *с.* Для цих констант виписані умови, які гарантують задовільну якість отриманих випадкових чисел.

Особливістю лінійного конгруентного методу є те, що якщо співмножник і модуль відповідним чином підібрані, то результуюча послідовність чисел не буде статистично відрізнятись від випадкової послідовності елементів множини {*0, 1, 2, ..., m-1*}. Однак, всі елементи цієї послідовності однозначно визначаються чотирма параметрами: *X0, a, c, m*.

Якщо криптоаналітик знає про використання лінійного конгруентного методу з відомими параметрами, то відомою стає вся послідовність чисел. У той же час, навіть якщо криптоаналітик знає тільки лише про використання лінійного конгруентного методу, то інформація про невелику частину послідовності достатня для виявлення параметрів методу і всіх наступних елементів послідовності. Зокрема, якщо криптоаналітику відомі значення *X0, X1*,  *X2, X3*, то вони задовольняють системі рівнянь:

|  |  |
| --- | --- |
|  | (3.2) |

з якої можна отримати значення параметрів *а, с* і *т.*

Тому конгруентний генератор не є криптографічно стійким.

3.2 Blum Blum Shub (BBS) генератор

Blum Blum Shub (далі BBS) – це генератор псевдовипадкових чисел, запропонований в 1986 році Ленором Блюмом, Мануелем Блюмом і Майклом Шубом (Blum et al, 1986).

BBS виглядає так:

|  |  |
| --- | --- |
|  | (3.3) |

де *M=pq* є добудком двох великих простих *p* і *q*. На кожному кроці алгоритму вихідні дані виходять із *Xk* шляхом взяття або біта парності, або одного чи більше найменш значущих біт. Структурна схема зображена на рис.4 [15].

випадковий

потік бітів

зворотній зв’язок

r

*m*

*X0*= *r2* mod *m*

*Xk+1*=*Xk2*mod *m*

Вилучення молодшого біта

Рисунок 4 – Схема генерації псевдовипадкової послідовності бітів BBS генератора

Два простих числа, *p* і *q*, повинні бути обидва порівнювані з 3 по модулю 4 (це гарантує, що кожний квадратичний залишок має один квадратний корінь, який також є квадратичним залишком) і найбільший спільний дільник *НСД(φ(p-1), φ(q-1))* повинен бути малий (це збільшує довжину циклу) [5].

Цікавою особливістю цього алгоритму є те, що для отримання *Xk* необов'язково обчислювати всі *k*-1 попередніх чисел, якщо відомо початковий стан генератора *X0* і числа *p* і *q*. *n*-не значення може бути обчислене «напряму» використовуючи формулу:

|  |  |
| --- | --- |
|  | (3.4) |

BBS генератор придатний для криптографічних перетворень, але не для моделювання, бо недостатньо швидкий. Однак, він має незвично високу стійкість, яка забезпечується якістю генератора виходячи з обчислювальної складності задачі факторизації чисел. Коли прості числа обрані обережно, і *O*(loglog*M*) біт кожного *Xk* є вихідними даними, тоді межа обрана як *M* швидко росте, і обчислення вихідних біт буде настільки ж важко, як і факторизація *M*.

Якщо факторизація цілих чисел важка (як передбачається), тоді BBS з великим *M* матиме вихід, вільний від будь-яких невипадкових шаблонів, які можуть бути виявлені при достатньому обсязі обчислень. Однак, можлива поява швидкого алгоритму для факторизації, і внаслідок цього BBS не є гарантовано надійним.

3.2 Генератори на основі лінійних регістрів зсуву із зворотним зв’язком

Регістр зсуву з лінійним зворотним зв'язком ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *linear feedback shift register, LFSR*) — поширений спосіб отримання [псевдовипадкових послідовностей](http://uk.wikipedia.org/wiki/%D0%9F%D1%81%D0%B5%D0%B2%D0%B4%D0%BE%D0%B2%D0%B8%D0%BF%D0%B0%D0%B4%D0%BA%D0%BE%D0%B2%D0%B0_%D0%BF%D0%BE%D1%81%D0%BB%D1%96%D0%B4%D0%BE%D0%B2%D0%BD%D1%96%D1%81%D1%82%D1%8C), наприклад [ключ-потоків](http://uk.wikipedia.org/wiki/%D0%9F%D0%BE%D1%82%D1%96%D0%BA_%D0%BA%D0%BB%D1%8E%D1%87%D0%B0) [6, 7]. Це зумовлене такими причинами:

* LFSR добре втілюються на апаратному рівні;
* вони можуть утворювати послідовності із великими періодами;
* вони можуть утворювати послідовності з хорошими статистичними

властивостями;

* завдяки своїй будові, вони легко піддаються аналізу за допомогою

алгебраїчних технік.

Регістр зсуву з лінійної зворотним зв'язком - регістр зсуву бітових слів, у якого вхідний біт є лінійною функцією стану інших бітів регістра до зсуву. Може бути організований як програмними, так і апаратними.

У регістрі зсуву з лінійної зворотним зв'язком виділяють дві частини: власне регістра зсуву і схеми (або підпрограми), що обчислює значення вхідного біта. Регістр складається з функціональних осередків (або бітів машинного слова або кількох слів), в кожній з якої зберігається поточний стан одного біта. Кількість осередків *L*, називають довжиною регістра. Біти (комірки) зазвичай нумеруються числами 0, 1, 2,…, *L*- 1, кожна з яких здатна зберігати 1 біт, причому в осередок 0 відбувається ін’єкція обчисленого біта, а з комірки *L*- 1 витягується черговий згенерований біт. Обчислення вхідного біта зазвичай виробляється до зсуву регістра, і тільки після зсуву значення обчисленого біта поміщається в комірку 0 (рис.5).[6]

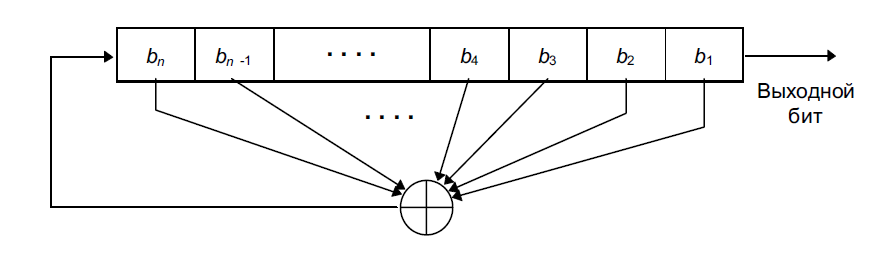


Рисунок 5 – Структурна схема генерації псевдовипадкової послідовності LFSR-генератора

Періодом регістра зсуву називають мінімальну довжину одержуваної послідовності до початку її повторення. Так як регістр з *L* бітових осередків має тільки 2*L*-1 різних ненульових станів, то, принципово, період регістра не може перевищувати це число. Якщо період регістра дорівнює цьому числу, то такий регістр називають регістром максимального періоду [8].

Для LFSR функція зворотного зв'язку є лінійною булевою функцією від станів всіх або деяких бітів регістра. Наприклад, сума по модулю два або її логічна інверсія є лінійною булевою функцією (операція XOR) і найбільш часто застосовується в таких регістрах.

Управління регістром в апаратних реалізаціях проводиться подачею з імпульсу (інакше званого тактового або синхроімпульса) на всі комірки, в програмних - виконанням програмного циклу, що включає обчислення функції зворотного зв'язку і зсуву бітів у слові.

Протягом кожного такту часу виконуються наступні операції:

* вміст комірки формує черговий біт вихідний послідовності бітів;
* новий вміст комірки 0 визначається бітом зворотного зв'язку, що є значенням функції зворотного зв'язку (XOR);
* вміст *і*-тої комірки переміщується в комірку *і+*1 для будь-якого *і*, де

0<*i*<*L*-1;

* вміст 0-ї комірки приймає значення обчисленого біта.

Комірки, що мають обчислятись операцією XOR, обираються згідно обраного поліному *f*(*x*).

Для того, щоб конкретний LFSR мав максимальний період, багаточлен, створений з послідовності та константи 1, має бути примітивним по модулю 2. Степінь багаточлена є довжиною регістру зсуву. Примітивний багаточлен степені *n* – це непривідний багаточлен, який є дільником *x2n-1*+1, але не є дільником *xd* +1 для усіх *d* , що є дільниками 2*n*-1 [6, 7, 8, 9].

В загальному випадку не існує простого способу генерації примітивних багаточленів по модулю 2. Краще обрати багаточлен випадковим чином, і перевіряти, чи є даний багаточлен примітивним [6, 10].

4 Аналіз генераторів псевдовипадкових чисел

4.1 Реалізація конгруентного генератора та аналіз згенерованих послідовностей

Лінійний конгруентний генератор, що працює згідно формули (3.1) реалізовано мовою С++, код программи наведено нижче:

#include <cstdio>

#include <cstdlib>

#include <iostream>

typedef unsigned long long int uint64\_t;

using namespace std;

uint64\_t lcg(uint64\_t seed);

uint64\_t bbs(uint64\_t seed);

void printBin(uint64\_t);

int main(int argc, char\*\* argv) {

if (argc < 2) {

cout << "Usage: " << argv[0] << ' ' << "<seed>" << endl;

return 0;

}

for (long i = 0; i < 240000000/32; ++i) {

printBin(lcg(atoi(argv[1])));

}

return 0;

}

uint64\_t lcg(uint64\_t seed) {

uint64\_t a = 1103515245;

uint64\_t c = 12345;

uint64\_t m = 0x80000000;

static uint64\_t x = seed;

x = (a \* x + c) % m;

return x;

}

void printBin(uint64\_t x) {

int bit = 0;

for (int shift = sizeof(int) \* 8 - 1 ; shift >= 0; --shift) {

bit = (x >> shift) & 1;

cout << bit;

}

cout << endl;

}

Дана програма генерує 240 послідовностей, які мають 1000000 біти кожна.

Вхідними даними є:

* початкове значення х0 задається користувачем;
* значення m = 2147483648;
* значення a = 1103515245;
* значення c = 12345.

Згенерована послідовність псевдовипадкових чисел піддана статистичному тестуванню з використанням методики NIST STS, що рекомендована Національним інститутом по стандартизації і технологіям США - NIST SP 800-22.

Для здійснення тестування було обрано наступні параметри:

* довжина послідовності, що тестується n = 10^6 біт;
* кількість послідовностей, що тестується m = 240;

Таким чином, обсяг вибірки, що тестується склав *N* = 240 × 10^6 біт;

Кількість тестів (*q*) для різних довжин *q* = 188, таким чином, статистичний портрет генератора містить 18800 значень імовірності *Р.*

Нижня границя складає значення *rmin* = 0.9666, а для тесту Random excursion (variant) test *rmin* = 0.9655. З цих позицій аналізуються результати тестування генераторів.

В Додатку А наведено результати обчислення по кожному з тестів.

У таблиці 1 наведено зведені результати тестування лінійного конгруентного генератора. За даною методикою та при тих самих параметрах тестуються генератори, що описані у підрозділах 4.2 та 4.3. Розподілення ймовірностей зображено на рис. 3.

Таблиця 1

|  |  |
| --- | --- |
| Припустиме значення долі проходження тесту для вибірки розміром 240 двійкових послідовностей дорівнює | 0.9666 |
| Припустиме значення долі проходження тесту для вибірки розміром 141 двійкових послідовностей для тесту Random-Excursion дорівнює | 0.9655 |
| Кількість тестів, в яких тестування пройшло 99% послідовностей | 165 |
| Кількість тестів, в яких тестування пройшло 96% послідовностей | 187 |
| Кількість тестів, в яких значення імовірності Р <=0,01 | 146 |
| Кількість тестів, в яких значення імовірності Р <=0,001 | 135 |
| Кількість тестів, в яких значення імовірності Р <=0,05 | 155 |

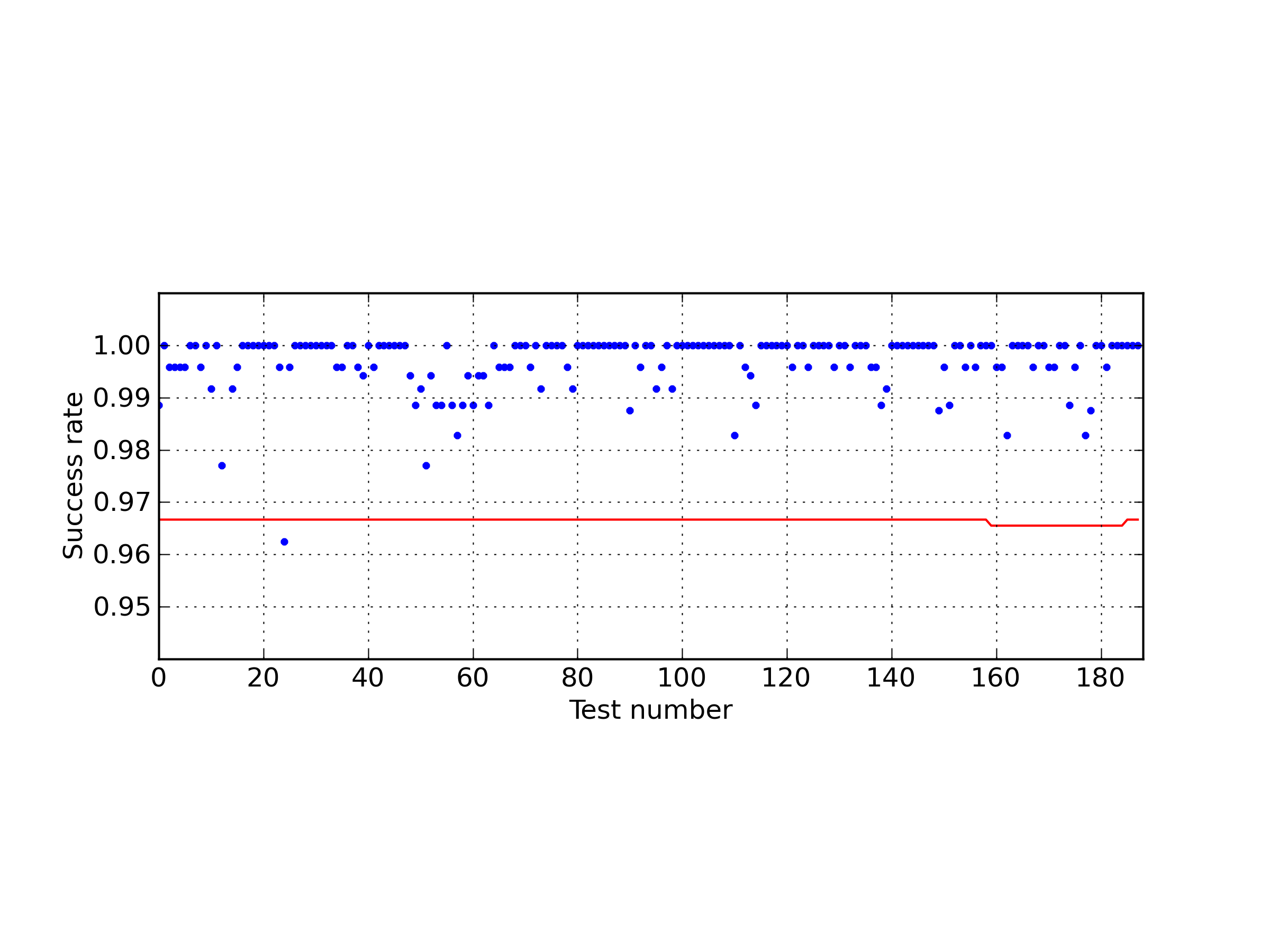


Рисунок 3 – Графік результату тестування

Відповідно до отриманих результатів можна зробити висновок, що лінійний конгруентний генератор не пройшов комплексний контроль і не відповідає вимогам згідно з методикою NIST STS.

4.2 Реалізація Blum Blum Shub BBS генератора та аналіз згенерованих послідовностей

Генератор BBS, що генерує послідовності згідно формули (3.3) реалізовано мовою С++, використовуючи великі числа, за допомогою функцій спеціалізованої бібліотеки GMP. Код программи наведено нижче:

#include <cstdlib>

#include <iostream>

#include <gmp.h>

#include <gmpxx.h>

using namespace std;

void bbsInit(unsigned int seed);

unsigned int bbs();

void printBin(unsigned int x);

void printBin(unsigned int x) {

int bit = 0;

for (int shift = sizeof(unsigned int) \* 8 - 1; shift >= 0; --shift) {

bit = (x >> shift) & 1;

cout << bit;

}

cout << endl;

}

struct {

mpz\_t state;

mpz\_t mod;

mpz\_t output\_mask;

} bbs\_conf;

void bbsInit(unsigned int seed) {

mpz\_init(bbs\_conf.state);

mpz\_init(bbs\_conf.mod);

mpz\_init(bbs\_conf.output\_mask);

mpz\_set\_ui(bbs\_conf.state, seed);

mpz\_set\_str(bbs\_conf.mod, "87f8c4dccb9485c7c330b5757d2d751", 16);

mpz\_set\_ui(bbs\_conf.output\_mask, 0xFFFFFFFF);

for (int i = 0; i < 32; ++i) {

bbs();

}

}

unsigned int bbs() {

mpz\_mul(bbs\_conf.state, bbs\_conf.state, bbs\_conf.state);

mpz\_mod(bbs\_conf.state, bbs\_conf.state, bbs\_conf.mod);

return mpz\_get\_ui(bbs\_conf.state);

}

int main(int argc, char\*\* argv) {

bbsInit(77);

for (int i = 0; i < 240000000/32; ++i) {

printBin(bbs());

}

return 0;

}

Дана програма генерує 240 послідовностей, які мають 10^6 біти кожна. При заданих значеннях:

* початкове значення х0 - 77;
* значення m = 87f8c4dccb9485c7c330b5757d2d75116

У таблиці 2 наведено зведені результати тестування генератора BBS.

Розподілення ймовірностей зображено на рис. 4.

Таблиця 2

|  |  |
| --- | --- |
| Припустиме значення долі проходження тесту для вибірки розміром 240 двійкових послідовностей дорівнює | 0.9666 |
| Припустиме значення долі проходження тесту для вибірки розміром 174 двійкових послідовностей для тесту Random-Excursion дорівнює | 0.9589 |
| Кількість тестів, в яких тестування пройшло 99% послідовностей | 106 |
| Кількість тестів, в яких тестування пройшло 96% послідовностей | 188 |
| Кількість тестів, в яких значення імовірності Р <=0,01 | 5 |
| Кількість тестів, в яких значення імовірності Р <=0,001 | 0 |
| Кількість тестів, в яких значення імовірності Р <=0,05 | 12 |

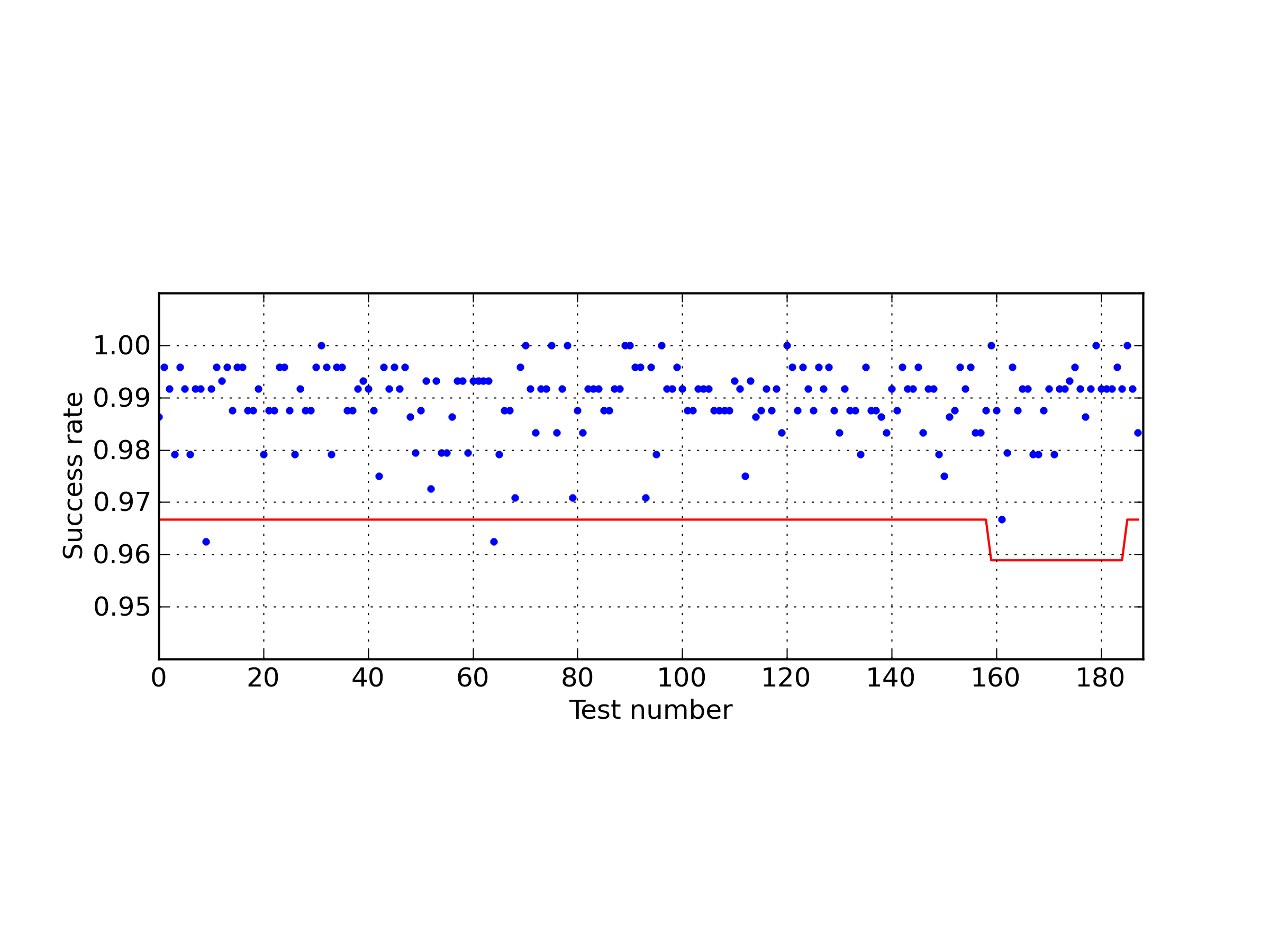


Рисунок 4 – Графік результату тестування

Відповідно до отриманих результатів можна зробити висновок, що генератор BBS пройшов комплексний контроль і відповідає вимогам згідно з методикою NIST STS.

4.3 Реалізація генератора на основі лінійних регістрів зсуву із зворотним зв’язком та аналіз згенерованих послідовностей

Генератор на основі лінійних регістрів зсуву із зворотним зв’язком (LFSR) реалізовано мовою C/С++, використовуючи великі числа, за допомогою функцій спеціалізованої бібліотеки GMP. Код програми наведено нижче:

#include <cstdlib>

#include <iostream>

#include <gmp.h>

#include <gmpxx.h>

using namespace std;

typedef unsigned int uint;

void printBin(unsigned int x);

void printBin(unsigned int x) {

int bit = 0;

for (int shift = sizeof(unsigned int) \* 8 - 1; shift >= 0; --shift) {

bit = (x >> shift) & 1;

cout << bit;

}

cout << endl;

}

class LFSR {

private:

uint tap1;

uint tap2;

uint tap3;

uint tap4;

mpz\_class reg;

public:

LFSR(uint e1, uint e2, uint e3, uint e4);

void init\_seed(const char\* value);

int tick();

unsigned int rand();

};

LFSR::LFSR(uint e1, uint e2, uint e3, uint e4) {

tap1 = e1;

tap2 = e2;

tap3 = e3;

tap4 = e4;

};

void LFSR::init\_seed(const char\* value) {

reg = mpz\_class(value, 16);

// Make sure the init value will not make the register longer.

mpz\_class mask(1);

mask = (mask << tap1) - 1;

reg = reg & mask;

for (int i = 0; i < 1024; ++i) {

tick();

}

}

int LFSR::tick() {

mpz\_t temp;

mpz\_init(temp);

int rand\_bit = reg.get\_ui() & 1;

int xorbit = 1;

mpz\_set(temp, reg.get\_mpz\_t());

xorbit ^= mpz\_tstbit(temp, tap1);

xorbit ^= mpz\_tstbit(temp, tap2);

xorbit ^= mpz\_tstbit(temp, tap3);

xorbit ^= mpz\_tstbit(temp, tap4);

reg = reg >> 1;

mpz\_set(temp, reg.get\_mpz\_t());

if (xorbit == 1) {

mpz\_setbit(temp, tap1);

}

reg = mpz\_class(temp);

mpz\_clear(temp);

return rand\_bit;

}

unsigned int LFSR::rand() {

unsigned int result = 0;

for (int i = 0; i < sizeof(unsigned int) \* 8; ++i) {

result |= tick() << i;

}

return result;

}

int main(int argc, char\*\* argv) {

LFSR test(133, 9, 8, 2);

test.init\_seed("0123456789abcdef");

for (int i = 0; i < 240000000/32; ++i) {

printBin(test.rand());

}

return 0;

}

Дана програма генерує 240 послідовностей, які мають 10^6 біти кожна. При заданих значеннях:

* початкове значення 0123456789abcdef16 ;

Поліном послідовності,яка генерує генератор має вигляд (133, 9, 8, 2):

Вхідна послідовність має довжину – 133 біти. Період генерації послідовності дорівнює 2n-1 = 1,0889035741470030830827987437817e+40.

У таблиці 3 наведено зведені результати тестування LFSR-генератора. Розподілення ймовірностей зображено на рис. 3.

Таблиця 3

|  |  |
| --- | --- |
| Припустиме значення долі проходження тесту для вибірки розміром 100 двійкових послідовностей дорівнює | 0.9666 |
| Припустиме значення долі проходження тесту для вибірки розміром 146 двійкових послідовностей для тесту Random-Excursion дорівнює | 0.9645 |
| Кількість тестів, в яких тестування пройшло 99% послідовностей | 103 |
| Кількість тестів, в яких тестування пройшло 96% послідовностей | 188 |
| Кількість тестів, в яких значення імовірності Р <=0,01 | 4 |
| Кількість тестів, в яких значення імовірності Р <=0,001 | 0 |
| Кількість тестів, в яких значення імовірності Р <=0,05 | 10 |

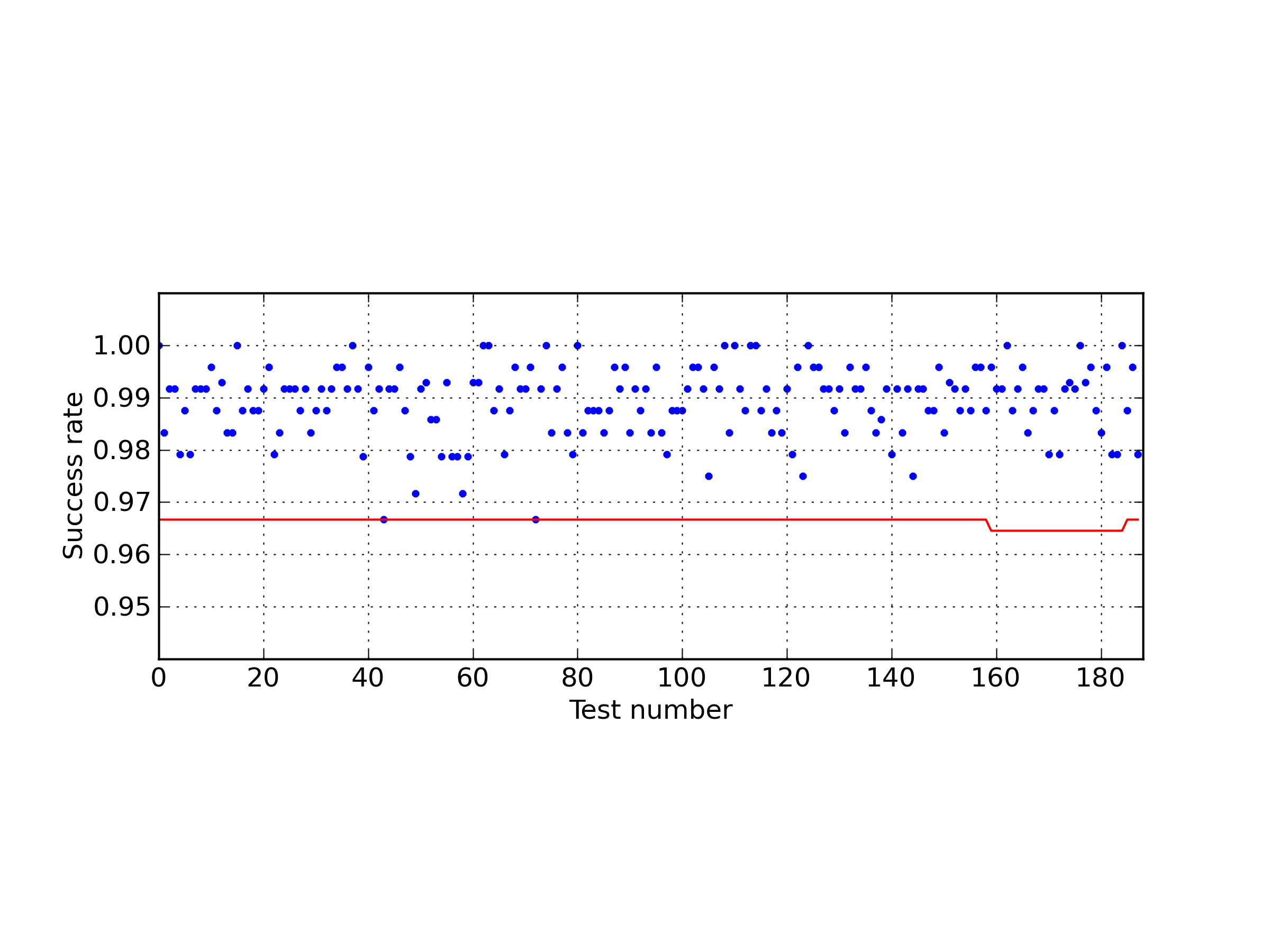


Рисунок 5 – Графік результату тестування

Відповідно до отриманих результатів можна зробити висновок, що LFSR-генератор пройшов комплексний контроль і відповідає вимогам згідно з методикою NIST STS.

Висновок

Методика NIST STS є гнучким та ефективним інструментом перевірки властивостей випадкових і псевдовипадкових послідовностей, який фактично став стандартним засобом тестування. Пакет тестів NIST STS містить 15 тестів, що дозволяють виконати комплексну оцінку статистичних властивостей послідовностей, що тестуються.

Для порівняльного аналізу статистичних властивостей ГПСЧ були взяті три найвідоміші генератори: лінійний конгруентний генератор, BBS-генератор та LFSR генератор. Найгірші результати тестування показав лінійний конгруентний генератор, тим самим підтвердив припущення щодо його ненадійної криптостійкості. BBS-генератор реалізовувався на великий числах, та у ході тестування отримав дуже добрі результати. Найкращі результати виявились у LFSR генератора, який теж реалізовувався на великих числах, тим самим підтвердив популярність використання, та те що він вважається найнадійнішим ГПСЧ. Генератори BBS та LFSR він успішно пройшла всі тести і повністью відповідає всім вимогам пакету статистичного тестування NIST STS.

Перелік посилань

1. A Million Random Digits with 100,000 Normal Deviates, Santa Monica, Calif.: RAND Corporation, MR-1418-RC, 2001. As of October 4, 2012: <http://www.rand.org/pubs/monograph_reports/MR1418>.
2. John von Neumann "Various techniques used in connection with random digits" («Методи, пов’язані з випадковими числами») // Applied Mathematics Series, no. 12, 36–38 (1951).
3. J. von Neumann, "Various Techniques Used in Connection with Random Digits," // Monte Carlo Method (A. S. Householder, G. E. Forsythe, and H. H. Germond, eds.), National Bureau of Standards Applied Mathematics Series, 12, Washington, D.C.: U.S. Government Printing Office, 1951, pp. 36–38.
4. Andrew Rukhin , Juan Soto , James Nechvatal et al. "A statistical test suite for random and pseudorandom number generators for cryptographic applications” // NIST Special Publication 800-22 (revised May 15 (2002)
5. І.Д. Горбенко, Т.О. Гріненко Захист інформації в інформаційно-телекомунікаційних системах: Навч. посібник. Ч. 1. Криптографічний захист інформації – Харків: ХНУРЕ, 2004. – 368 с.
6. Брюс Шнайер «Прикладная криптография» // - Изд. Триумф, 2002, с,275-279.
7. N. Zierler, “Primitive Trinomials Whose Degree Is a Mersenne Exponent”, Information and Control, v. 15, 1969, pp. 67-69.
8. N. Zierler and J. Brillhart. “On Primitive Trinomials (mod 2)” // Information and Control, v. 13, n.6, 1968, pp. 541-544.
9. S.W. Golomb, Shift Register Sequences, San Francisco: Holden-Day, 1967.
10. T. Hansen and G.L. Mullen, “Primitive Polynomials over Finite Fields”// Mathematics of Computation, v.59, n. 200, Oct 1992, pp. 639-643.
11. A.Rukhin, J.Soto. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST Special Publication 800-22, 09.2000.
12. F. N. David and D. E. Barton, Combinatorial Chance. New York: Hafner Publishing Co., 1962, p. 230.
13. Pal Revesz, Random Walk in Random and Non-Random Environments. Singapore: World Scientic, 1990.
14. George Marsaglia, DIEHARD: a battery of tests of randomness. <http://stat.fsu.edu/~geo/diehard.html>.
15. Горбенко І.Д. Прикладна криптографія. Теорія. Практика. //МОНМС України ХНУРЕ, ПАТ «ІІТ». – Харків: Форт, 2012. – 870с.

ДОДАТОК А

Результати тестування вихідної псевдовипадкової послідовності BBS генератора.

Статистичний портрет – 240 двійкових послідовностей по 1000000 біт кожна

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Гістограма значень ймовірностей Р | | | | | | | | | | | Значення імовірності Р | Доля проходження тесту | Статистичний тест |
| №  Тесту | 0..0,1 | 0,1..0,2 | 0,2..0,3 | 0,3..0,4 | 0,4..0,5 | 0,5..0,6 | 0,6..0,7 | 0,7..0,8 | 0,8..0,9 | 0,9..1,0 |
|  | 29 | 29 | 25 | 21 | 21 | 27 | 27 | 20 | 25 | 16 | 0.6371 | 0.9875 | AperiodicTemplate |
|  | 27 | 28 | 21 | 28 | 14 | 30 | 22 | 21 | 29 | 20 | 0.3504 | 0.9958 | AperiodicTemplate |
|  | 23 | 26 | 23 | 27 | 26 | 35 | 19 | 21 | 23 | 17 | 0.4070 | 0.9625 | AperiodicTemplate |
|  | 22 | 22 | 25 | 22 | 33 | 23 | 22 | 22 | 21 | 28 | 0.8043 | 0.9916 | AperiodicTemplate |
|  | 26 | 24 | 25 | 20 | 27 | 25 | 21 | 16 | 32 | 24 | 0.6371 | 0.9875 | AperiodicTemplate |
|  | 29 | 26 | 25 | 25 | 26 | 20 | 16 | 30 | 27 | 16 | 0.4070 | 0.9916 | AperiodicTemplate |
|  | 20 | 26 | 23 | 19 | 21 | 22 | 23 | 28 | 34 | 24 | 0.6024 | 0.9791 | AperiodicTemplate |
|  | 25 | 23 | 27 | 24 | 29 | 20 | 22 | 18 | 26 | 26 | 0.9001 | 0.9916 | AperiodicTemplate |
|  | 20 | 23 | 24 | 28 | 21 | 26 | 21 | 22 | 19 | 36 | 0.3924 | 0.9916 | AperiodicTemplate |
|  | 28 | 29 | 19 | 22 | 22 | 24 | 21 | 28 | 22 | 25 | 0.8881 | 0.9791 | AperiodicTemplate |
|  | 31 | 19 | 24 | 22 | 28 | 35 | 20 | 23 | 21 | 17 | 0.2086 | 0.9791 | AperiodicTemplate |
|  | 23 | 21 | 19 | 26 | 26 | 32 | 25 | 22 | 19 | 27 | 0.7315 | 0.9875 | AperiodicTemplate |
|  | 15 | 25 | 27 | 29 | 26 | 23 | 20 | 23 | 33 | 19 | 0.3371 | 0.9958 | AperiodicTemplate |
|  | 25 | 23 | 18 | 18 | 29 | 31 | 19 | 28 | 26 | 23 | 0.5257 | 0.9791 | AperiodicTemplate |
|  | 18 | 27 | 27 | 21 | 33 | 17 | 26 | 24 | 26 | 21 | 0.4606 | 0.9958 | AperiodicTemplate |
|  | 31 | 27 | 20 | 24 | 20 | 21 | 24 | 27 | 21 | 25 | 0.8415 | 0.9958 | AperiodicTemplate |
|  | 19 | 28 | 24 | 16 | 33 | 26 | 22 | 26 | 23 | 23 | 0.5009 | 0.9958 | AperiodicTemplate |
|  | 23 | 24 | 21 | 39 | 20 | 18 | 23 | 23 | 28 | 21 | 0.1588 | 0.9916 | AperiodicTemplate |
|  | 30 | 21 | 21 | 19 | 25 | 25 | 27 | 28 | 21 | 23 | 0.8485 | 0.9875 | AperiodicTemplate |
|  | 16 | 33 | 21 | 33 | 23 | 22 | 26 | 17 | 28 | 21 | 0.1516 | 0.9875 | AperiodicTemplate |
|  | 23 | 16 | 28 | 20 | 16 | 28 | 29 | 29 | 24 | 27 | 0.3641 | 0.9833 | AperiodicTemplate |
|  | 27 | 23 | 21 | 21 | 21 | 27 | 21 | 25 | 27 | 27 | 0.9608 | 0.9916 | AperiodicTemplate |
|  | 26 | 18 | 24 | 24 | 25 | 25 | 29 | 28 | 14 | 27 | 0.5341 | 0.9791 | AperiodicTemplate |
|  | 15 | 26 | 21 | 26 | 26 | 20 | 32 | 28 | 20 | 26 | 0.4296 | 0.9958 | AperiodicTemplate |
|  | 19 | 19 | 25 | 18 | 26 | 29 | 28 | 27 | 23 | 26 | 0.7315 | 0.9875 | AperiodicTemplate |
|  | 31 | 29 | 17 | 26 | 18 | 19 | 23 | 33 | 20 | 24 | 0.2180 | 0.9875 | AperiodicTemplate |
|  | 27 | 22 | 23 | 21 | 32 | 25 | 28 | 23 | 18 | 21 | 0.7146 | 0.9875 | AperiodicTemplate |
|  | 28 | 22 | 25 | 31 | 22 | 18 | 20 | 22 | 25 | 27 | 0.7564 | 0.9875 | AperiodicTemplate |
|  | 18 | 29 | 18 | 27 | 26 | 18 | 25 | 26 | 29 | 24 | 0.6024 | 0.9875 | AperiodicTemplate |
|  | 21 | 15 | 18 | 32 | 16 | 19 | 30 | 35 | 22 | 32 | 0.0126 | 0.9916 | AperiodicTemplate |
|  | 20 | 27 | 21 | 23 | 24 | 29 | 21 | 25 | 23 | 27 | 0.9496 | 0.9916 | AperiodicTemplate |
|  | 18 | 20 | 32 | 24 | 24 | 17 | 30 | 29 | 26 | 20 | 0.3306 | 0.9916 | AperiodicTemplate |
|  | 22 | 28 | 17 | 22 | 24 | 25 | 25 | 26 | 21 | 30 | 0.8195 | 0.9875 | AperiodicTemplate |
|  | 31 | 18 | 28 | 21 | 15 | 21 | 26 | 22 | 33 | 25 | 0.2086 | 0.975 | AperiodicTemplate |
|  | 22 | 24 | 29 | 23 | 22 | 24 | 24 | 27 | 18 | 27 | 0.9319 | 0.9916 | AperiodicTemplate |
|  | 26 | 18 | 28 | 21 | 27 | 27 | 29 | 26 | 17 | 21 | 0.6284 | 0.9875 | AperiodicTemplate |
|  | 28 | 21 | 20 | 28 | 28 | 28 | 27 | 26 | 19 | 15 | 0.4685 | 0.9708 | AperiodicTemplate |
|  | 21 | 15 | 28 | 26 | 22 | 30 | 22 | 21 | 27 | 28 | 0.5510 | 0.9875 | AperiodicTemplate |
|  | 33 | 23 | 22 | 26 | 27 | 24 | 23 | 10 | 24 | 28 | 0.1626 | 0.9833 | AperiodicTemplate |
|  | 24 | 26 | 21 | 19 | 35 | 26 | 26 | 18 | 26 | 19 | 0.3781 | 0.9916 | AperiodicTemplate |
|  | 26 | 20 | 21 | 28 | 23 | 22 | 25 | 25 | 25 | 25 | 0.9868 | 0.975 | AperiodicTemplate |
|  | 23 | 16 | 19 | 25 | 25 | 24 | 33 | 23 | 26 | 26 | 0.5766 | 0.9958 | AperiodicTemplate |
|  | 25 | 24 | 21 | 24 | 26 | 29 | 27 | 18 | 21 | 25 | 0.9168 | 0.9916 | AperiodicTemplate |
|  | 16 | 27 | 25 | 30 | 20 | 19 | 28 | 22 | 27 | 26 | 0.5680 | 1.0 | AperiodicTemplate |
|  | 16 | 26 | 18 | 31 | 25 | 30 | 21 | 22 | 29 | 22 | 0.3781 | 0.9916 | AperiodicTemplate |
|  | 23 | 28 | 20 | 25 | 29 | 22 | 28 | 21 | 20 | 24 | 0.8881 | 0.9916 | AperiodicTemplate |
|  | 32 | 19 | 20 | 23 | 24 | 29 | 32 | 19 | 14 | 28 | 0.1223 | 0.9708 | AperiodicTemplate |
|  | 21 | 18 | 21 | 32 | 26 | 20 | 27 | 25 | 21 | 29 | 0.5766 | 0.9875 | AperiodicTemplate |
|  | 25 | 22 | 16 | 23 | 26 | 25 | 31 | 14 | 22 | 36 | 0.0780 | 0.9916 | AperiodicTemplate |
|  | 20 | 23 | 25 | 28 | 35 | 19 | 22 | 12 | 32 | 24 | 0.0602 | 0.9958 | AperiodicTemplate |
|  | 26 | 25 | 20 | 24 | 21 | 27 | 23 | 16 | 33 | 25 | 0.5595 | 0.9875 | AperiodicTemplate |
|  | 16 | 24 | 27 | 22 | 23 | 35 | 19 | 26 | 25 | 23 | 0.3852 | 0.9958 | AperiodicTemplate |
|  | 13 | 22 | 20 | 34 | 22 | 26 | 26 | 27 | 24 | 26 | 0.2700 | 1.0 | AperiodicTemplate |
|  | 16 | 30 | 20 | 21 | 27 | 29 | 24 | 20 | 28 | 25 | 0.5341 | 0.9958 | AperiodicTemplate |
|  | 28 | 28 | 29 | 21 | 23 | 14 | 19 | 28 | 25 | 25 | 0.4606 | 0.9791 | AperiodicTemplate |
|  | 15 | 31 | 27 | 24 | 28 | 27 | 23 | 21 | 18 | 26 | 0.4450 | 0.9958 | AperiodicTemplate |
|  | 16 | 22 | 24 | 26 | 27 | 24 | 29 | 27 | 14 | 31 | 0.2757 | 1.0 | AperiodicTemplate |
|  | 25 | 24 | 22 | 26 | 32 | 16 | 20 | 30 | 21 | 24 | 0.5091 | 0.9916 | AperiodicTemplate |
|  | 29 | 16 | 23 | 25 | 20 | 23 | 28 | 21 | 32 | 23 | 0.5091 | 0.9833 | AperiodicTemplate |
|  | 25 | 31 | 27 | 27 | 23 | 25 | 20 | 23 | 22 | 17 | 0.7564 | 0.9916 | AperiodicTemplate |
|  | 24 | 21 | 23 | 24 | 24 | 22 | 33 | 23 | 27 | 19 | 0.7965 | 0.9833 | AperiodicTemplate |
|  | 18 | 26 | 23 | 26 | 28 | 29 | 24 | 18 | 24 | 24 | 0.8269 | 0.9916 | AperiodicTemplate |
|  | 17 | 34 | 26 | 19 | 25 | 29 | 26 | 20 | 23 | 21 | 0.3711 | 1.0 | AperiodicTemplate |
|  | 14 | 13 | 27 | 23 | 20 | 27 | 24 | 30 | 34 | 28 | 0.0487 | 0.9958 | AperiodicTemplate |
|  | 17 | 28 | 19 | 26 | 30 | 26 | 36 | 16 | 28 | 14 | 0.0244 | 0.9916 | AperiodicTemplate |
|  | 23 | 19 | 26 | 24 | 24 | 23 | 16 | 29 | 24 | 32 | 0.5680 | 0.9708 | AperiodicTemplate |
|  | 18 | 30 | 23 | 23 | 33 | 20 | 17 | 36 | 21 | 19 | 0.0556 | 0.9958 | AperiodicTemplate |
|  | 22 | 18 | 28 | 31 | 21 | 31 | 20 | 21 | 25 | 23 | 0.5425 | 0.9791 | AperiodicTemplate |
|  | 29 | 27 | 24 | 33 | 22 | 16 | 18 | 26 | 22 | 23 | 0.3924 | 0.9958 | AperiodicTemplate |
|  | 20 | 27 | 28 | 20 | 18 | 26 | 23 | 27 | 23 | 28 | 0.8195 | 0.9875 | AperiodicTemplate |
|  | 26 | 17 | 33 | 19 | 27 | 20 | 25 | 23 | 22 | 28 | 0.4765 | 0.9916 | AperiodicTemplate |
|  | 26 | 20 | 29 | 28 | 21 | 25 | 27 | 24 | 18 | 22 | 0.8343 | 0.9916 | AperiodicTemplate |
|  | 19 | 29 | 27 | 19 | 29 | 24 | 27 | 24 | 22 | 20 | 0.7646 | 1.0 | AperiodicTemplate |
|  | 19 | 32 | 20 | 23 | 19 | 27 | 22 | 32 | 32 | 14 | 0.0780 | 0.9958 | AperiodicTemplate |
|  | 23 | 18 | 23 | 21 | 23 | 24 | 17 | 31 | 27 | 33 | 0.3641 | 0.9916 | AperiodicTemplate |
|  | 21 | 31 | 16 | 24 | 32 | 32 | 25 | 15 | 16 | 28 | 0.0461 | 0.9833 | AperiodicTemplate |
|  | 24 | 21 | 25 | 21 | 30 | 34 | 19 | 23 | 24 | 19 | 0.4765 | 0.9916 | AperiodicTemplate |
|  | 24 | 21 | 18 | 21 | 25 | 30 | 24 | 20 | 33 | 24 | 0.5510 | 0.9875 | AperiodicTemplate |
|  | 23 | 23 | 28 | 20 | 23 | 23 | 27 | 32 | 22 | 19 | 0.7646 | 0.9875 | AperiodicTemplate |
|  | 23 | 21 | 27 | 25 | 29 | 21 | 18 | 31 | 19 | 26 | 0.6371 | 0.9958 | AperiodicTemplate |
|  | 23 | 30 | 22 | 25 | 19 | 24 | 24 | 23 | 25 | 25 | 0.9675 | 0.9916 | AperiodicTemplate |
|  | 24 | 26 | 27 | 27 | 29 | 26 | 18 | 19 | 22 | 22 | 0.8343 | 0.9875 | AperiodicTemplate |
|  | 26 | 23 | 23 | 29 | 25 | 23 | 21 | 28 | 17 | 25 | 0.8755 | 0.9958 | AperiodicTemplate |
|  | 43 | 21 | 15 | 24 | 18 | 22 | 22 | 17 | 33 | 25 | 0.0019 | 0.9666 | AperiodicTemplate |
|  | 25 | 18 | 17 | 25 | 23 | 22 | 23 | 36 | 24 | 27 | 0.3306 | 0.9875 | AperiodicTemplate |
|  | 27 | 18 | 27 | 32 | 23 | 27 | 17 | 20 | 21 | 28 | 0.4296 | 1.0 | AperiodicTemplate |
|  | 28 | 23 | 25 | 19 | 24 | 22 | 17 | 26 | 36 | 20 | 0.2873 | 0.9875 | AperiodicTemplate |
|  | 24 | 24 | 21 | 25 | 22 | 17 | 26 | 27 | 32 | 22 | 0.7399 | 0.9833 | AperiodicTemplate |
|  | 33 | 21 | 18 | 28 | 20 | 35 | 19 | 21 | 23 | 22 | 0.1516 | 0.9833 | AperiodicTemplate |
|  | 18 | 31 | 27 | 25 | 19 | 31 | 21 | 31 | 17 | 20 | 0.2040 | 0.9958 | AperiodicTemplate |
|  | 21 | 27 | 25 | 18 | 21 | 17 | 30 | 32 | 21 | 28 | 0.3572 | 0.9916 | AperiodicTemplate |
|  | 28 | 31 | 16 | 20 | 23 | 19 | 26 | 21 | 34 | 22 | 0.2133 | 0.9958 | AperiodicTemplate |
|  | 20 | 18 | 21 | 28 | 35 | 23 | 23 | 25 | 22 | 25 | 0.4765 | 0.9875 | AperiodicTemplate |
|  | 23 | 25 | 21 | 21 | 22 | 33 | 24 | 28 | 21 | 22 | 0.7807 | 0.9833 | AperiodicTemplate |
|  | 24 | 27 | 19 | 19 | 27 | 19 | 21 | 27 | 27 | 30 | 0.6890 | 0.9916 | AperiodicTemplate |
|  | 27 | 23 | 28 | 26 | 16 | 20 | 23 | 26 | 24 | 27 | 0.8195 | 1.0 | AperiodicTemplate |
|  | 33 | 25 | 19 | 20 | 19 | 27 | 29 | 25 | 22 | 21 | 0.5174 | 0.9916 | AperiodicTemplate |
|  | 30 | 26 | 21 | 19 | 29 | 27 | 25 | 24 | 17 | 22 | 0.6631 | 0.9833 | AperiodicTemplate |
|  | 22 | 25 | 27 | 15 | 30 | 27 | 24 | 17 | 23 | 30 | 0.3997 | 0.9916 | AperiodicTemplate |
|  | 25 | 24 | 26 | 22 | 25 | 26 | 20 | 30 | 14 | 28 | 0.5766 | 0.9916 | AperiodicTemplate |
|  | 24 | 30 | 24 | 24 | 18 | 20 | 27 | 23 | 26 | 24 | 0.8942 | 0.9916 | AperiodicTemplate |
|  | 24 | 27 | 21 | 23 | 19 | 26 | 21 | 28 | 22 | 29 | 0.8942 | 0.9958 | AperiodicTemplate |
|  | 28 | 21 | 19 | 32 | 22 | 25 | 21 | 19 | 26 | 27 | 0.6457 | 0.9916 | AperiodicTemplate |
|  | 21 | 19 | 22 | 23 | 17 | 22 | 29 | 16 | 35 | 36 | 0.0289 | 1.0 | AperiodicTemplate |
|  | 22 | 23 | 23 | 23 | 21 | 26 | 28 | 22 | 18 | 34 | 0.6024 | 0.9916 | AperiodicTemplate |
|  | 24 | 32 | 17 | 13 | 24 | 25 | 24 | 26 | 24 | 31 | 0.2133 | 0.9916 | AperiodicTemplate |
|  | 27 | 18 | 27 | 18 | 22 | 28 | 23 | 43 | 16 | 18 | 0.0045 | 0.9958 | AperiodicTemplate |
|  | 27 | 14 | 25 | 26 | 27 | 22 | 30 | 18 | 27 | 24 | 0.4685 | 0.9958 | AperiodicTemplate |
|  | 27 | 17 | 28 | 25 | 19 | 28 | 22 | 20 | 19 | 35 | 0.2277 | 0.9791 | AperiodicTemplate |
|  | 31 | 27 | 22 | 22 | 21 | 26 | 27 | 25 | 15 | 24 | 0.6284 | 0.9791 | AperiodicTemplate |
|  | 25 | 18 | 21 | 26 | 27 | 25 | 21 | 33 | 31 | 13 | 0.1480 | 0.9916 | AperiodicTemplate |
|  | 24 | 30 | 25 | 27 | 26 | 17 | 30 | 18 | 23 | 20 | 0.5510 | 0.9916 | AperiodicTemplate |
|  | 25 | 16 | 29 | 24 | 20 | 26 | 30 | 24 | 20 | 26 | 0.6457 | 0.9916 | AperiodicTemplate |
|  | 30 | 18 | 23 | 26 | 25 | 28 | 19 | 24 | 25 | 22 | 0.8195 | 0.9916 | AperiodicTemplate |
|  | 28 | 20 | 22 | 26 | 27 | 28 | 22 | 26 | 20 | 21 | 0.9058 | 0.9625 | AperiodicTemplate |
|  | 28 | 25 | 20 | 16 | 20 | 28 | 32 | 20 | 30 | 21 | 0.3053 | 0.9916 | AperiodicTemplate |
|  | 34 | 22 | 22 | 22 | 27 | 20 | 20 | 27 | 26 | 20 | 0.5766 | 0.9875 | AperiodicTemplate |
|  | 23 | 18 | 24 | 25 | 35 | 26 | 26 | 26 | 15 | 22 | 0.2992 | 0.9875 | AperiodicTemplate |
|  | 20 | 28 | 28 | 33 | 21 | 19 | 25 | 23 | 21 | 22 | 0.5938 | 0.9916 | AperiodicTemplate |
|  | 28 | 28 | 29 | 21 | 24 | 13 | 19 | 28 | 25 | 25 | 0.3852 | 0.9791 | AperiodicTemplate |
|  | 29 | 29 | 25 | 21 | 21 | 27 | 26 | 21 | 25 | 16 | 0.6890 | 0.9875 | AperiodicTemplate |
|  | 26 | 22 | 23 | 25 | 25 | 25 | 24 | 25 | 23 | 22 | 0.9998 | 0.9875 | AperiodicTemplate |
|  | 18 | 30 | 30 | 21 | 22 | 21 | 22 | 24 | 24 | 28 | 0.7146 | 0.9958 | AperiodicTemplate |
|  | 23 | 20 | 27 | 23 | 22 | 22 | 24 | 24 | 21 | 34 | 0.7399 | 0.9958 | AperiodicTemplate |
|  | 27 | 32 | 24 | 19 | 23 | 30 | 16 | 27 | 22 | 20 | 0.3924 | 0.9875 | AperiodicTemplate |
|  | 26 | 24 | 21 | 22 | 32 | 23 | 25 | 27 | 19 | 21 | 0.8119 | 0.9791 | AperiodicTemplate |
|  | 25 | 21 | 22 | 24 | 27 | 21 | 32 | 25 | 22 | 21 | 0.8690 | 0.9916 | AperiodicTemplate |
|  | 16 | 37 | 26 | 17 | 15 | 28 | 23 | 29 | 22 | 27 | 0.0403 | 0.9958 | AperiodicTemplate |
|  | 30 | 19 | 21 | 23 | 27 | 24 | 27 | 27 | 18 | 24 | 0.7807 | 0.975 | AperiodicTemplate |
|  | 28 | 24 | 27 | 19 | 30 | 20 | 20 | 24 | 18 | 30 | 0.5425 | 0.9875 | AperiodicTemplate |
|  | 32 | 22 | 24 | 13 | 28 | 23 | 25 | 22 | 24 | 27 | 0.4220 | 0.9916 | AperiodicTemplate |
|  | 20 | 14 | 25 | 32 | 24 | 17 | 27 | 28 | 25 | 28 | 0.2535 | 0.9958 | AperiodicTemplate |
|  | 17 | 17 | 30 | 32 | 32 | 25 | 14 | 24 | 30 | 19 | 0.0392 | 0.9916 | AperiodicTemplate |
|  | 27 | 24 | 30 | 18 | 19 | 20 | 24 | 30 | 18 | 30 | 0.3852 | 0.9958 | AperiodicTemplate |
|  | 26 | 22 | 23 | 22 | 24 | 20 | 23 | 28 | 25 | 27 | 0.9850 | 0.9916 | AperiodicTemplate |
|  | 14 | 23 | 21 | 24 | 42 | 28 | 17 | 19 | 26 | 26 | 0.0083 | 0.9916 | AperiodicTemplate |
|  | 24 | 27 | 23 | 37 | 23 | 18 | 19 | 25 | 21 | 23 | 0.3115 | 0.9875 | AperiodicTemplate |
|  | 18 | 27 | 20 | 21 | 29 | 18 | 29 | 27 | 29 | 22 | 0.5257 | 0.9916 | AperiodicTemplate |
|  | 28 | 19 | 27 | 18 | 21 | 25 | 25 | 28 | 25 | 24 | 0.8555 | 0.9916 | AperiodicTemplate |
|  | 21 | 23 | 36 | 29 | 28 | 22 | 19 | 23 | 14 | 25 | 0.1379 | 0.9916 | AperiodicTemplate |
|  | 21 | 23 | 29 | 17 | 24 | 29 | 28 | 25 | 18 | 26 | 0.6457 | 0.9875 | AperiodicTemplate |
|  | 28 | 19 | 22 | 30 | 27 | 25 | 19 | 22 | 29 | 19 | 0.6284 | 0.9875 | AperiodicTemplate |
|  | 19 | 23 | 34 | 24 | 28 | 28 | 14 | 22 | 19 | 29 | 0.1626 | 0.9916 | AperiodicTemplate |
|  | 15 | 27 | 22 | 23 | 25 | 22 | 34 | 21 | 29 | 22 | 0.3572 | 0.9916 | AperiodicTemplate |
|  | 25 | 35 | 32 | 24 | 20 | 27 | 17 | 25 | 12 | 23 | 0.0500 | 1.0 | AperiodicTemplate |
|  | 18 | 22 | 28 | 29 | 32 | 21 | 17 | 26 | 25 | 22 | 0.4527 | 1.0 | AperiodicTemplate |
|  | 21 | 25 | 26 | 18 | 25 | 33 | 31 | 14 | 24 | 23 | 0.2277 | 1.0 | AperiodicTemplate |
|  | 23 | 30 | 30 | 15 | 30 | 21 | 22 | 24 | 24 | 21 | 0.4527 | 0.9833 | AperiodicTemplate |
|  | 30 | 21 | 25 | 18 | 16 | 28 | 21 | 20 | 18 | 43 | 0.0038 | 0.9875 | ApproximateEntropy |
|  | 23 | 23 | 18 | 27 | 28 | 20 | 28 | 20 | 32 | 21 | 0.5680 | 0.9958 | BlockFrequency |
|  | 21 | 29 | 20 | 23 | 22 | 15 | 24 | 31 | 25 | 30 | 0.4145 | 0.9875 | CumulativeSums |
|  | 22 | 26 | 21 | 24 | 27 | 22 | 26 | 28 | 14 | 30 | 0.5595 | 0.9875 | CumulativeSums |
|  | 20 | 25 | 22 | 27 | 24 | 16 | 38 | 28 | 19 | 21 | 0.1165 | 0.9958 | FFT |
|  | 26 | 23 | 26 | 14 | 24 | 29 | 18 | 28 | 22 | 30 | 0.3997 | 0.9875 | Frequency |
|  | 17 | 20 | 28 | 27 | 22 | 30 | 36 | 21 | 20 | 19 | 0.1412 | 0.9875 | LinearComplexity |
|  | 17 | 23 | 28 | 26 | 19 | 20 | 26 | 32 | 24 | 25 | 0.5852 | 0.9958 | LongestRun |
|  | 28 | 15 | 27 | 26 | 32 | 17 | 22 | 23 | 25 | 25 | 0.3852 | 0.9791 | PeriodicTemplate |
|  | 16 | 15 | 12 | 21 | 12 | 18 | 12 | 12 | 10 | 18 | 0.4915 | 0.9863 | RandomExcursion |
|  | 16 | 19 | 11 | 11 | 12 | 20 | 17 | 12 | 18 | 10 | 0.3986 | 0.9931 | RandomExcursion |
|  | 9 | 24 | 14 | 9 | 20 | 10 | 12 | 14 | 16 | 18 | 0.0639 | 0.9931 | RandomExcursion |
|  | 10 | 16 | 15 | 18 | 10 | 10 | 18 | 20 | 18 | 11 | 0.3169 | 0.9863 | RandomExcursion |
|  | 12 | 19 | 6 | 15 | 15 | 20 | 17 | 16 | 11 | 15 | 0.2757 | 0.9931 | RandomExcursion |
|  | 10 | 14 | 15 | 15 | 17 | 9 | 15 | 12 | 23 | 16 | 0.3390 | 0.9863 | RandomExcursion |
|  | 11 | 10 | 12 | 21 | 16 | 17 | 14 | 10 | 19 | 16 | 0.3741 | 0.9931 | RandomExcursion |
|  | 10 | 16 | 21 | 15 | 13 | 16 | 9 | 15 | 14 | 17 | 0.5486 | 0.9863 | RandomExcursion |
|  | 11 | 17 | 19 | 19 | 17 | 6 | 17 | 14 | 11 | 15 | 0.2474 | 0.9863 | RandomExcursion |
|  | 13 | 17 | 16 | 13 | 18 | 13 | 17 | 17 | 8 | 14 | 0.7255 | 0.9863 | RandomExcursion |
|  | 17 | 14 | 16 | 12 | 22 | 12 | 11 | 14 | 15 | 13 | 0.6519 | 0.9931 | RandomExcursion |
|  | 17 | 13 | 13 | 17 | 18 | 17 | 15 | 11 | 13 | 12 | 0.8916 | 0.9794 | RandomExcursion |
|  | 12 | 15 | 13 | 22 | 22 | 17 | 13 | 7 | 14 | 11 | 0.1078 | 0.9726 | RandomExcursion |
|  | 15 | 15 | 22 | 16 | 16 | 9 | 16 | 13 | 9 | 15 | 0.4113 | 0.9931 | RandomExcursion |
|  | 15 | 16 | 20 | 16 | 14 | 17 | 11 | 11 | 13 | 13 | 0.8087 | 0.9794 | RandomExcursion |
|  | 14 | 16 | 25 | 15 | 11 | 10 | 15 | 14 | 16 | 10 | 0.1976 | 0.9931 | RandomExcursion |
|  | 13 | 14 | 11 | 26 | 6 | 11 | 25 | 18 | 11 | 11 | 0.0012 | 0.9931 | RandomExcursion |
|  | 9 | 21 | 16 | 11 | 15 | 17 | 19 | 12 | 17 | 9 | 0.2474 | 0.9931 | RandomExcursion |
|  | 9 | 15 | 16 | 13 | 16 | 21 | 20 | 9 | 15 | 12 | 0.2958 | 0.9794 | RandomExcursion |
|  | 11 | 17 | 17 | 17 | 10 | 15 | 11 | 19 | 18 | 11 | 0.5341 | 0.9863 | RandomExcursion |
|  | 18 | 18 | 13 | 17 | 12 | 14 | 18 | 13 | 13 | 10 | 0.7681 | 0.9794 | RandomExcursion |
|  | 18 | 10 | 15 | 17 | 20 | 11 | 19 | 14 | 8 | 14 | 0.3062 | 0.9931 | RandomExcursion |
|  | 13 | 17 | 21 | 15 | 9 | 17 | 13 | 18 | 13 | 10 | 0.4241 | 0.9931 | RandomExcursion |
|  | 14 | 14 | 17 | 13 | 7 | 21 | 20 | 14 | 11 | 15 | 0.2757 | 0.9931 | RandomExcursion |
|  | 10 | 13 | 14 | 19 | 17 | 15 | 17 | 16 | 12 | 13 | 0.8343 | 0.9794 | RandomExcursion |
|  | 15 | 16 | 14 | 16 | 17 | 8 | 15 | 15 | 11 | 19 | 0.6963 | 0.9931 | RandomExcursion |
|  | 19 | 25 | 27 | 21 | 24 | 29 | 30 | 19 | 26 | 20 | 0.7146 | 0.9875 | Rank |
|  | 26 | 27 | 18 | 36 | 21 | 20 | 19 | 28 | 24 | 21 | 0.2644 | 0.9916 | Runs |
|  | 21 | 18 | 16 | 26 | 16 | 33 | 32 | 26 | 20 | 32 | 0.0500 | 0.9875 | Serial |
|  | 22 | 28 | 23 | 28 | 20 | 18 | 20 | 27 | 28 | 26 | 0.7807 | 0.9958 | Serial |
|  | 14 | 22 | 27 | 24 | 22 | 25 | 31 | 21 | 24 | 30 | 0.4527 | 0.9958 | Universal |

Результати тестування вихідної псевдовипадкової послідовності конгруентного генератора.

Статистичний портрет – 240 двійкових послідовностей по 1000000 біт кожна

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Гістограма значень ймовірностей Р | | | | | | | | | | | Значення імовірності Р | Доля проходження тесту | Статистичний тест |
| №  Тесту | 0..0,1 | 0,1..0,2 | 0,2..0,3 | 0,3..0,4 | 0,4..0,5 | 0,5..0,6 | 0,6..0,7 | 0,7..0,8 | 0,8..0,9 | 0,9..1,0 |
|  | 5 | 12 | 7 | 18 | 19 | 24 | 37 | 40 | 33 | 45 | 0.0 | 1.0 | AperiodicTemplate |
|  | 3 | 3 | 12 | 22 | 18 | 17 | 39 | 36 | 38 | 52 | 0.0 | 1.0 | AperiodicTemplate |
|  | 5 | 16 | 11 | 22 | 22 | 27 | 31 | 23 | 39 | 44 | 0.0 | 1.0 | AperiodicTemplate |
|  | 12 | 9 | 15 | 26 | 22 | 20 | 23 | 36 | 37 | 40 | 0,000002 | 0.9958 | AperiodicTemplate |
|  | 6 | 12 | 19 | 20 | 13 | 33 | 23 | 33 | 38 | 43 | 0.0 | 1.0 | AperiodicTemplate |
|  | 10 | 7 | 15 | 18 | 30 | 18 | 27 | 28 | 32 | 55 | 0.0 | 1.0 | AperiodicTemplate |
|  | 10 | 15 | 20 | 18 | 17 | 29 | 27 | 32 | 31 | 41 | 9.2e-0 | 0.9958 | AperiodicTemplate |
|  | 1 | 12 | 13 | 24 | 18 | 23 | 25 | 34 | 46 | 44 | 0.0 | 1.0 | AperiodicTemplate |
|  | 5 | 22 | 21 | 19 | 26 | 25 | 22 | 24 | 29 | 47 | 0,000007 | 1.0 | AperiodicTemplate |
|  | 6 | 15 | 20 | 21 | 20 | 31 | 29 | 38 | 22 | 38 | 1.6e-0 | 1.0 | AperiodicTemplate |
|  | 6 | 13 | 8 | 24 | 18 | 27 | 31 | 34 | 31 | 48 | 0.0 | 0.9958 | AperiodicTemplate |
|  | 9 | 9 | 19 | 19 | 25 | 30 | 28 | 25 | 28 | 48 | 0.0 | 1.0 | AperiodicTemplate |
|  | 12 | 16 | 15 | 28 | 28 | 21 | 20 | 27 | 35 | 38 | 0.0009 | 1.0 | AperiodicTemplate |
|  | 6 | 14 | 18 | 15 | 19 | 20 | 24 | 38 | 38 | 48 | 0.0 | 0.9958 | AperiodicTemplate |
|  | 13 | 19 | 18 | 23 | 18 | 23 | 21 | 31 | 29 | 45 | 0.0002 | 1.0 | AperiodicTemplate |
|  | 14 | 14 | 13 | 21 | 21 | 27 | 33 | 37 | 37 | 23 | 0.0001 | 0.9958 | AperiodicTemplate |
|  | 7 | 14 | 15 | 17 | 21 | 29 | 29 | 31 | 32 | 45 | 0.0 | 1.0 | AperiodicTemplate |
|  | 15 | 16 | 28 | 14 | 23 | 29 | 37 | 25 | 27 | 26 | 0.0206 | 1.0 | AperiodicTemplate |
|  | 19 | 17 | 26 | 21 | 15 | 26 | 27 | 26 | 32 | 31 | 0.1908 | 0.9958 | AperiodicTemplate |
|  | 3 | 22 | 18 | 18 | 17 | 29 | 31 | 33 | 35 | 34 | 0,00001 | 1.0 | AperiodicTemplate |
|  | 6 | 13 | 11 | 21 | 19 | 33 | 31 | 32 | 26 | 48 | 0.0 | 1.0 | AperiodicTemplate |
|  | 5 | 15 | 14 | 17 | 21 | 14 | 37 | 24 | 49 | 44 | 0.0 | 1.0 | AperiodicTemplate |
|  | 8 | 12 | 8 | 24 | 16 | 24 | 35 | 30 | 42 | 41 | 0.0 | 1.0 | AperiodicTemplate |
|  | 5 | 6 | 22 | 21 | 19 | 15 | 33 | 32 | 35 | 52 | 0.0 | 1.0 | AperiodicTemplate |
|  | 12 | 15 | 18 | 21 | 24 | 29 | 23 | 28 | 34 | 36 | 0.0058 | 0.9958 | AperiodicTemplate |
|  | 15 | 18 | 15 | 23 | 23 | 22 | 32 | 24 | 38 | 30 | 0.0134 | 1.0 | AperiodicTemplate |
|  | 9 | 12 | 17 | 14 | 26 | 23 | 22 | 41 | 40 | 36 | 0.0 | 1.0 | AperiodicTemplate |
|  | 5 | 6 | 15 | 20 | 13 | 24 | 25 | 39 | 47 | 46 | 0.0 | 1.0 | AperiodicTemplate |
|  | 5 | 10 | 15 | 14 | 21 | 18 | 19 | 49 | 31 | 58 | 0.0 | 1.0 | AperiodicTemplate |
|  | 6 | 9 | 12 | 18 | 23 | 26 | 29 | 35 | 32 | 50 | 0.0 | 1.0 | AperiodicTemplate |
|  | 9 | 12 | 14 | 21 | 18 | 32 | 33 | 34 | 27 | 40 | 0,000002 | 1.0 | AperiodicTemplate |
|  | 12 | 13 | 16 | 25 | 28 | 21 | 25 | 31 | 38 | 31 | 0.0013 | 1.0 | AperiodicTemplate |
|  | 10 | 15 | 14 | 22 | 31 | 31 | 29 | 26 | 28 | 34 | 0.0020 | 1.0 | AperiodicTemplate |
|  | 15 | 17 | 14 | 21 | 20 | 33 | 32 | 24 | 36 | 28 | 0.0054 | 0.9958 | AperiodicTemplate |
|  | 6 | 14 | 15 | 19 | 27 | 18 | 34 | 42 | 35 | 30 | 0.0 | 1.0 | AperiodicTemplate |
|  | 9 | 23 | 23 | 14 | 15 | 29 | 28 | 30 | 30 | 39 | 0.0002 | 0.9958 | AperiodicTemplate |
|  | 20 | 15 | 19 | 30 | 23 | 16 | 34 | 24 | 34 | 25 | 0.0392 | 0.9916 | AperiodicTemplate |
|  | 7 | 20 | 21 | 19 | 23 | 33 | 32 | 23 | 33 | 29 | 0.0033 | 1.0 | AperiodicTemplate |
|  | 6 | 11 | 13 | 27 | 20 | 30 | 27 | 31 | 33 | 42 | 0.0 | 1.0 | AperiodicTemplate |
|  | 9 | 18 | 11 | 16 | 22 | 22 | 23 | 21 | 44 | 54 | 0.0 | 1.0 | AperiodicTemplate |
|  | 6 | 12 | 17 | 19 | 24 | 24 | 25 | 38 | 36 | 39 | 0,000001 | 0.9958 | AperiodicTemplate |
|  | 5 | 13 | 17 | 15 | 21 | 19 | 30 | 27 | 39 | 54 | 0.0 | 0.9958 | AperiodicTemplate |
|  | 6 | 7 | 18 | 21 | 25 | 33 | 25 | 27 | 32 | 46 | 0.0 | 1.0 | AperiodicTemplate |
|  | 8 | 15 | 21 | 17 | 17 | 34 | 28 | 29 | 36 | 35 | 0,00005 | 0.9958 | AperiodicTemplate |
|  | 10 | 17 | 22 | 26 | 16 | 22 | 23 | 28 | 28 | 48 | 1.3e-0 | 1.0 | AperiodicTemplate |
|  | 10 | 12 | 20 | 25 | 22 | 25 | 22 | 39 | 29 | 36 | 0.0002 | 1.0 | AperiodicTemplate |
|  | 8 | 10 | 12 | 16 | 17 | 21 | 30 | 28 | 49 | 49 | 0.0 | 1.0 | AperiodicTemplate |
|  | 7 | 14 | 18 | 18 | 29 | 33 | 24 | 29 | 28 | 40 | 0,00004 | 1.0 | AperiodicTemplate |
|  | 7 | 14 | 23 | 16 | 24 | 21 | 26 | 37 | 35 | 37 | 1.4e-0 | 1.0 | AperiodicTemplate |
|  | 7 | 14 | 17 | 13 | 27 | 30 | 31 | 27 | 29 | 45 | 0.0 | 1.0 | AperiodicTemplate |
|  | 14 | 28 | 20 | 19 | 25 | 26 | 26 | 18 | 36 | 28 | 0.0886 | 1.0 | AperiodicTemplate |
|  | 7 | 9 | 11 | 16 | 20 | 27 | 27 | 40 | 32 | 51 | 0.0 | 1.0 | AperiodicTemplate |
|  | 2 | 12 | 23 | 15 | 17 | 23 | 21 | 32 | 37 | 58 | 0.0 | 1.0 | AperiodicTemplate |
|  | 6 | 15 | 15 | 19 | 26 | 26 | 20 | 34 | 38 | 41 | 0.0 | 0.9958 | AperiodicTemplate |
|  | 4 | 8 | 17 | 21 | 16 | 18 | 25 | 34 | 44 | 53 | 0.0 | 1.0 | AperiodicTemplate |
|  | 8 | 11 | 13 | 18 | 19 | 23 | 31 | 35 | 39 | 43 | 0.0 | 1.0 | AperiodicTemplate |
|  | 8 | 13 | 20 | 20 | 24 | 31 | 27 | 30 | 24 | 43 | 0,00004 | 1.0 | AperiodicTemplate |
|  | 11 | 9 | 5 | 15 | 29 | 18 | 38 | 36 | 28 | 51 | 0.0 | 0.9958 | AperiodicTemplate |
|  | 5 | 7 | 17 | 12 | 25 | 26 | 25 | 29 | 47 | 47 | 0.0 | 1.0 | AperiodicTemplate |
|  | 8 | 10 | 11 | 22 | 20 | 30 | 34 | 17 | 41 | 47 | 0.0 | 0.9916 | AperiodicTemplate |
|  | 6 | 9 | 15 | 22 | 26 | 27 | 29 | 36 | 27 | 43 | 0.0 | 0.9916 | AperiodicTemplate |
|  | 6 | 6 | 11 | 18 | 22 | 22 | 30 | 29 | 45 | 51 | 0.0 | 1.0 | AperiodicTemplate |
|  | 16 | 19 | 20 | 13 | 21 | 31 | 27 | 33 | 34 | 26 | 0.0184 | 0.9958 | AperiodicTemplate |
|  | 4 | 14 | 23 | 21 | 27 | 31 | 29 | 16 | 37 | 38 | 0,000003 | 1.0 | AperiodicTemplate |
|  | 13 | 9 | 14 | 19 | 23 | 27 | 34 | 26 | 39 | 36 | 0,000008 | 0.9916 | AperiodicTemplate |
|  | 5 | 15 | 23 | 22 | 24 | 21 | 23 | 29 | 32 | 46 | 0,000002 | 1.0 | AperiodicTemplate |
|  | 7 | 16 | 16 | 17 | 23 | 21 | 30 | 32 | 31 | 47 | 0.0 | 0.9958 | AperiodicTemplate |
|  | 11 | 12 | 14 | 22 | 22 | 17 | 35 | 27 | 31 | 49 | 0.0 | 0.9916 | AperiodicTemplate |
|  | 2 | 10 | 10 | 21 | 26 | 21 | 20 | 24 | 44 | 62 | 0.0 | 1.0 | AperiodicTemplate |
|  | 3 | 14 | 20 | 22 | 28 | 21 | 34 | 23 | 36 | 39 | 0,000001 | 1.0 | AperiodicTemplate |
|  | 4 | 11 | 22 | 18 | 23 | 28 | 22 | 43 | 26 | 43 | 0.0 | 1.0 | AperiodicTemplate |
|  | 3 | 22 | 16 | 14 | 33 | 25 | 31 | 30 | 33 | 33 | 1.1e-0 | 1.0 | AperiodicTemplate |
|  | 10 | 24 | 16 | 16 | 25 | 25 | 24 | 22 | 34 | 44 | 7.1e-0 | 1.0 | AperiodicTemplate |
|  | 7 | 12 | 24 | 27 | 11 | 27 | 24 | 35 | 32 | 41 | 0,000001 | 0.9958 | AperiodicTemplate |
|  | 10 | 13 | 13 | 21 | 31 | 28 | 24 | 33 | 24 | 43 | 0,000008 | 1.0 | AperiodicTemplate |
|  | 7 | 15 | 10 | 23 | 29 | 17 | 26 | 31 | 34 | 48 | 0.0 | 1.0 | AperiodicTemplate |
|  | 10 | 11 | 19 | 25 | 15 | 17 | 32 | 39 | 33 | 39 | 0.0 | 0.9958 | AperiodicTemplate |
|  | 11 | 16 | 10 | 18 | 22 | 36 | 26 | 36 | 29 | 36 | 1.3e-0 | 1.0 | AperiodicTemplate |
|  | 6 | 8 | 15 | 13 | 23 | 29 | 28 | 23 | 43 | 52 | 0.0 | 1.0 | AperiodicTemplate |
|  | 14 | 12 | 13 | 15 | 20 | 24 | 26 | 28 | 43 | 45 | 0.0 | 1.0 | AperiodicTemplate |
|  | 4 | 9 | 6 | 12 | 18 | 28 | 32 | 31 | 43 | 57 | 0.0 | 1.0 | AperiodicTemplate |
|  | 13 | 10 | 22 | 21 | 21 | 27 | 32 | 31 | 26 | 37 | 0.0017 | 1.0 | AperiodicTemplate |
|  | 9 | 18 | 15 | 21 | 29 | 29 | 29 | 29 | 30 | 31 | 0.0078 | 1.0 | AperiodicTemplate |
|  | 12 | 29 | 23 | 15 | 24 | 24 | 29 | 29 | 27 | 28 | 0.1379 | 0.9958 | AperiodicTemplate |
|  | 5 | 17 | 17 | 17 | 24 | 32 | 16 | 32 | 42 | 38 | 0.0 | 0.9958 | AperiodicTemplate |
|  | 6 | 10 | 16 | 23 | 26 | 24 | 25 | 24 | 31 | 55 | 0.0 | 1.0 | AperiodicTemplate |
|  | 6 | 4 | 14 | 18 | 24 | 32 | 23 | 34 | 36 | 49 | 0.0 | 1.0 | AperiodicTemplate |
|  | 7 | 16 | 15 | 16 | 19 | 27 | 26 | 34 | 42 | 38 | 0.0 | 1.0 | AperiodicTemplate |
|  | 12 | 17 | 18 | 15 | 29 | 24 | 32 | 31 | 22 | 40 | 0.0005 | 0.9958 | AperiodicTemplate |
|  | 7 | 14 | 15 | 17 | 19 | 18 | 20 | 38 | 42 | 50 | 0.0 | 1.0 | AperiodicTemplate |
|  | 19 | 10 | 14 | 27 | 16 | 27 | 40 | 25 | 28 | 34 | 0.0001 | 0.9958 | AperiodicTemplate |
|  | 4 | 7 | 16 | 22 | 24 | 30 | 30 | 34 | 28 | 45 | 0.0 | 1.0 | AperiodicTemplate |
|  | 12 | 17 | 20 | 18 | 23 | 27 | 24 | 30 | 35 | 34 | 0.0112 | 1.0 | AperiodicTemplate |
|  | 9 | 16 | 11 | 22 | 29 | 25 | 24 | 31 | 38 | 35 | 4.7e-0 | 1.0 | AperiodicTemplate |
|  | 13 | 15 | 21 | 14 | 18 | 23 | 26 | 37 | 40 | 33 | 4.4e-0 | 1.0 | AperiodicTemplate |
|  | 10 | 13 | 14 | 16 | 25 | 30 | 19 | 34 | 40 | 39 | 0.0 | 1.0 | AperiodicTemplate |
|  | 3 | 13 | 14 | 24 | 20 | 21 | 26 | 32 | 43 | 44 | 0.0 | 1.0 | AperiodicTemplate |
|  | 8 | 22 | 22 | 18 | 20 | 35 | 27 | 15 | 35 | 38 | 5.6e-0 | 1.0 | AperiodicTemplate |
|  | 2 | 10 | 9 | 15 | 18 | 28 | 28 | 33 | 51 | 46 | 0.0 | 1.0 | AperiodicTemplate |
|  | 11 | 10 | 11 | 22 | 21 | 20 | 27 | 28 | 43 | 47 | 0.0 | 0.9958 | AperiodicTemplate |
|  | 8 | 22 | 12 | 24 | 22 | 15 | 26 | 27 | 39 | 45 | 0.0 | 1.0 | AperiodicTemplate |
|  | 3 | 12 | 16 | 15 | 32 | 26 | 28 | 26 | 38 | 44 | 0.0 | 1.0 | AperiodicTemplate |
|  | 19 | 20 | 15 | 20 | 24 | 26 | 30 | 22 | 30 | 34 | 0.1516 | 0.9875 | AperiodicTemplate |
|  | 4 | 9 | 12 | 22 | 24 | 15 | 25 | 43 | 40 | 46 | 0.0 | 1.0 | AperiodicTemplate |
|  | 5 | 4 | 16 | 25 | 18 | 21 | 29 | 28 | 42 | 52 | 0.0 | 1.0 | AperiodicTemplate |
|  | 16 | 17 | 26 | 15 | 20 | 29 | 22 | 28 | 31 | 36 | 0.0266 | 0.9958 | AperiodicTemplate |
|  | 9 | 13 | 15 | 17 | 14 | 34 | 26 | 34 | 35 | 43 | 0.0 | 1.0 | AperiodicTemplate |
|  | 6 | 12 | 11 | 21 | 22 | 25 | 23 | 32 | 46 | 42 | 0.0 | 0.9958 | AperiodicTemplate |
|  | 9 | 11 | 18 | 16 | 25 | 13 | 25 | 41 | 32 | 50 | 0.0 | 0.9958 | AperiodicTemplate |
|  | 8 | 12 | 11 | 16 | 21 | 19 | 28 | 38 | 36 | 51 | 0.0 | 1.0 | AperiodicTemplate |
|  | 8 | 14 | 16 | 18 | 16 | 27 | 27 | 36 | 36 | 42 | 0.0 | 0.9958 | AperiodicTemplate |
|  | 10 | 14 | 16 | 18 | 24 | 21 | 35 | 32 | 38 | 32 | 0,00005 | 0.9958 | AperiodicTemplate |
|  | 5 | 12 | 21 | 23 | 20 | 21 | 24 | 35 | 36 | 43 | 0.0 | 1.0 | AperiodicTemplate |
|  | 14 | 15 | 12 | 22 | 23 | 22 | 27 | 28 | 39 | 38 | 0.0001 | 0.9916 | AperiodicTemplate |
|  | 4 | 12 | 16 | 14 | 27 | 33 | 16 | 35 | 33 | 50 | 0.0 | 1.0 | AperiodicTemplate |
|  | 4 | 10 | 15 | 14 | 21 | 35 | 33 | 33 | 35 | 40 | 0.0 | 1.0 | AperiodicTemplate |
|  | 5 | 7 | 19 | 18 | 12 | 29 | 36 | 32 | 38 | 44 | 0.0 | 1.0 | AperiodicTemplate |
|  | 5 | 13 | 16 | 15 | 23 | 22 | 26 | 17 | 40 | 63 | 0.0 | 1.0 | AperiodicTemplate |
|  | 3 | 11 | 17 | 12 | 17 | 25 | 37 | 30 | 38 | 50 | 0.0 | 1.0 | AperiodicTemplate |
|  | 4 | 8 | 17 | 21 | 16 | 19 | 25 | 33 | 44 | 53 | 0.0 | 1.0 | AperiodicTemplate |
|  | 5 | 12 | 7 | 19 | 18 | 25 | 36 | 40 | 34 | 44 | 0.0 | 1.0 | AperiodicTemplate |
|  | 5 | 8 | 15 | 20 | 21 | 18 | 26 | 36 | 31 | 60 | 0.0 | 1.0 | AperiodicTemplate |
|  | 8 | 7 | 11 | 12 | 19 | 20 | 26 | 34 | 47 | 56 | 0.0 | 0.9958 | AperiodicTemplate |
|  | 11 | 16 | 22 | 18 | 17 | 30 | 28 | 31 | 29 | 38 | 0.0014 | 1.0 | AperiodicTemplate |
|  | 9 | 15 | 20 | 21 | 16 | 19 | 24 | 35 | 31 | 50 | 0.0 | 0.9958 | AperiodicTemplate |
|  | 10 | 11 | 13 | 18 | 22 | 22 | 37 | 35 | 29 | 43 | 0.0 | 1.0 | AperiodicTemplate |
|  | 8 | 7 | 17 | 15 | 22 | 23 | 30 | 29 | 33 | 56 | 0.0 | 1.0 | AperiodicTemplate |
|  | 10 | 16 | 20 | 19 | 25 | 19 | 19 | 36 | 33 | 43 | 1.1e-0 | 1.0 | AperiodicTemplate |
|  | 11 | 14 | 22 | 22 | 24 | 24 | 32 | 30 | 34 | 27 | 0.0164 | 1.0 | AperiodicTemplate |
|  | 8 | 18 | 9 | 12 | 24 | 26 | 22 | 39 | 35 | 47 | 0.0 | 0.9958 | AperiodicTemplate |
|  | 6 | 7 | 17 | 23 | 24 | 24 | 21 | 31 | 41 | 46 | 0.0 | 1.0 | AperiodicTemplate |
|  | 3 | 11 | 15 | 19 | 26 | 23 | 30 | 23 | 36 | 54 | 0.0 | 1.0 | AperiodicTemplate |
|  | 8 | 19 | 22 | 20 | 30 | 27 | 13 | 38 | 28 | 35 | 0.0001 | 1.0 | AperiodicTemplate |
|  | 12 | 19 | 13 | 21 | 21 | 28 | 24 | 31 | 36 | 35 | 0.0016 | 1.0 | AperiodicTemplate |
|  | 5 | 12 | 18 | 12 | 31 | 32 | 25 | 31 | 31 | 43 | 0.0 | 1.0 | AperiodicTemplate |
|  | 9 | 11 | 10 | 18 | 22 | 33 | 25 | 32 | 33 | 47 | 0.0 | 0.9958 | AperiodicTemplate |
|  | 10 | 15 | 29 | 14 | 16 | 22 | 31 | 25 | 34 | 44 | 0,000003 | 1.0 | AperiodicTemplate |
|  | 6 | 13 | 17 | 16 | 29 | 20 | 28 | 29 | 31 | 51 | 0.0 | 1.0 | AperiodicTemplate |
|  | 10 | 16 | 8 | 25 | 18 | 34 | 36 | 27 | 34 | 32 | 0,000006 | 1.0 | AperiodicTemplate |
|  | 8 | 10 | 23 | 17 | 23 | 22 | 30 | 27 | 36 | 44 | 0,000001 | 1.0 | AperiodicTemplate |
|  | 12 | 21 | 23 | 15 | 24 | 23 | 25 | 25 | 36 | 36 | 0.0091 | 1.0 | AperiodicTemplate |
|  | 6 | 12 | 22 | 22 | 26 | 17 | 23 | 43 | 34 | 35 | 0,000001 | 1.0 | AperiodicTemplate |
|  | 8 | 13 | 10 | 23 | 24 | 20 | 25 | 31 | 37 | 49 | 0.0 | 1.0 | AperiodicTemplate |
|  | 11 | 16 | 17 | 24 | 12 | 27 | 30 | 37 | 35 | 31 | 9.9e-0 | 1.0 | AperiodicTemplate |
|  | 4 | 13 | 9 | 26 | 21 | 25 | 29 | 27 | 35 | 51 | 0.0 | 1.0 | AperiodicTemplate |
|  | 19 | 22 | 23 | 29 | 24 | 26 | 18 | 21 | 31 | 27 | 0.6631 | 0.9875 | AperiodicTemplate |
|  | 13 | 7 | 18 | 20 | 31 | 30 | 24 | 31 | 26 | 40 | 4.5e-0 | 1.0 | AperiodicTemplate |
|  | 9 | 11 | 17 | 17 | 20 | 30 | 27 | 21 | 47 | 41 | 0.0 | 1.0 | AperiodicTemplate |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 240 | 0.0 | 1.0 | ApproximateEntropy |
|  | 0 | 0 | 0 | 2 | 2 | 7 | 3 | 8 | 31 | 187 | 0.0 | 1.0 | BlockFrequency |
|  | 7 | 10 | 9 | 18 | 18 | 30 | 31 | 31 | 36 | 50 | 0.0 | 0.9958 | CumulativeSums |
|  | 6 | 9 | 13 | 21 | 18 | 21 | 29 | 31 | 34 | 58 | 0.0 | 0.9958 | CumulativeSums |
|  | 240 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0 | 0.0 | FFT |
|  | 7 | 13 | 15 | 20 | 31 | 36 | 20 | 34 | 31 | 33 | 0,00001 | 0.9958 | Frequency |
|  | 16 | 23 | 26 | 19 | 27 | 35 | 21 | 15 | 27 | 31 | 0.0780 | 0.9916 | LinearComplexity |
|  | 18 | 22 | 26 | 18 | 20 | 26 | 27 | 27 | 25 | 31 | 0.6371 | 1.0 | LongestRun |
|  | 37 | 20 | 28 | 27 | 24 | 19 | 17 | 24 | 29 | 15 | 0.0618 | 0.9875 | PeriodicTemplate |
|  | 14 | 16 | 19 | 16 | 20 | 28 | 14 | 18 | 14 | 15 | 0.3600 | 0.9885 | RandomExcursion |
|  | 17 | 13 | 17 | 21 | 18 | 25 | 22 | 17 | 14 | 10 | 0.3049 | 0.9827 | RandomExcursion |
|  | 18 | 22 | 23 | 12 | 15 | 16 | 14 | 17 | 16 | 21 | 0.6432 | 0.9942 | RandomExcursion |
|  | 18 | 22 | 23 | 14 | 19 | 19 | 17 | 12 | 14 | 16 | 0.6677 | 0.9885 | RandomExcursion |
|  | 19 | 20 | 21 | 15 | 19 | 11 | 16 | 20 | 15 | 18 | 0.8186 | 0.9827 | RandomExcursion |
|  | 15 | 18 | 18 | 18 | 15 | 17 | 26 | 17 | 13 | 17 | 0.7041 | 0.9885 | RandomExcursion |
|  | 17 | 15 | 18 | 16 | 14 | 17 | 19 | 11 | 23 | 24 | 0.5106 | 0.9885 | RandomExcursion |
|  | 20 | 14 | 15 | 15 | 12 | 13 | 22 | 21 | 20 | 22 | 0.4990 | 0.9827 | RandomExcursion |
|  | 23 | 12 | 9 | 12 | 24 | 15 | 25 | 23 | 15 | 16 | 0.0338 | 0.9885 | RandomExcursion |
|  | 21 | 16 | 7 | 19 | 22 | 15 | 19 | 25 | 17 | 13 | 0.1309 | 0.9885 | RandomExcursion |
|  | 17 | 14 | 22 | 23 | 25 | 17 | 15 | 18 | 12 | 11 | 0.2269 | 0.9885 | RandomExcursion |
|  | 14 | 19 | 23 | 26 | 19 | 19 | 17 | 13 | 17 | 7 | 0.0925 | 0.9942 | RandomExcursion |
|  | 15 | 18 | 20 | 19 | 20 | 20 | 19 | 17 | 16 | 10 | 0.8079 | 0.9942 | RandomExcursion |
|  | 18 | 15 | 20 | 17 | 23 | 16 | 17 | 19 | 13 | 16 | 0.8952 | 0.9885 | RandomExcursion |
|  | 16 | 18 | 28 | 20 | 16 | 17 | 13 | 16 | 9 | 21 | 0.1400 | 0.9885 | RandomExcursion |
|  | 14 | 22 | 23 | 18 | 17 | 17 | 16 | 17 | 14 | 16 | 0.8494 | 0.9942 | RandomExcursion |
|  | 21 | 17 | 9 | 16 | 15 | 21 | 14 | 19 | 23 | 19 | 0.4318 | 0.9942 | RandomExcursion |
|  | 17 | 17 | 12 | 15 | 18 | 22 | 15 | 21 | 13 | 24 | 0.5106 | 0.9942 | RandomExcursion |
|  | 15 | 17 | 19 | 16 | 16 | 17 | 16 | 24 | 21 | 13 | 0.7970 | 0.9885 | RandomExcursion |
|  | 17 | 18 | 21 | 21 | 19 | 11 | 17 | 27 | 17 | 6 | 0.0443 | 0.9942 | RandomExcursion |
|  | 22 | 20 | 16 | 13 | 22 | 14 | 12 | 21 | 18 | 16 | 0.5821 | 1.0 | RandomExcursion |
|  | 15 | 15 | 18 | 24 | 21 | 17 | 14 | 15 | 24 | 11 | 0.3410 | 0.9770 | RandomExcursion |
|  | 19 | 16 | 9 | 25 | 21 | 16 | 14 | 21 | 19 | 14 | 0.2716 | 0.9885 | RandomExcursion |
|  | 21 | 16 | 17 | 12 | 7 | 21 | 25 | 17 | 14 | 24 | 0.0576 | 0.9885 | RandomExcursion |
|  | 15 | 18 | 16 | 16 | 16 | 14 | 24 | 17 | 21 | 17 | 0.8494 | 0.9827 | RandomExcursion |
|  | 24 | 20 | 13 | 7 | 17 | 15 | 18 | 17 | 27 | 16 | 0.0576 | 0.9770 | RandomExcursion |
|  | 23 | 23 | 20 | 24 | 28 | 34 | 16 | 25 | 27 | 20 | 0.4070 | 0.9916 | Rank |
|  | 3 | 14 | 17 | 12 | 26 | 30 | 36 | 35 | 27 | 40 | 0.0 | 1.0 | Runs |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 240 | 0.0 | 1.0 | Serial |
|  | 14 | 23 | 17 | 25 | 16 | 23 | 29 | 41 | 26 | 26 | 0.0076 | 0.9958 | Serial |
|  | 48 | 38 | 22 | 22 | 35 | 13 | 17 | 15 | 15 | 15 | 0.0 | 0.9625 | Universal |

Результати тестування вихідної псевдовипадкової послідовності LFSR генератора.

Статистичний портрет – 240 двійкових послідовностей по 1000000 біт кожна

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Гістограма значень ймовірностей Р | | | | | | | | | | | Значення імовірності Р | Доля проходження тесту | Статистичний тест |
| №  Тесту | 0..0,1 | 0,1..0,2 | 0,2..0,3 | 0,3..0,4 | 0,4..0,5 | 0,5..0,6 | 0,6..0,7 | 0,7..0,8 | 0,8..0,9 | 0,9..1,0 |
|  | 22 | 23 | 24 | 27 | 29 | 32 | 24 | 22 | 21 | 16 | 0.5852 | 0.9916 | AperiodicTemplate |
|  | 34 | 32 | 29 | 16 | 19 | 18 | 22 | 20 | 25 | 25 | 0.1223 | 0.9833 | AperiodicTemplate |
|  | 33 | 29 | 23 | 23 | 22 | 19 | 23 | 22 | 22 | 24 | 0.7315 | 0.9875 | AperiodicTemplate |
|  | 28 | 28 | 21 | 18 | 25 | 19 | 27 | 27 | 25 | 22 | 0.8119 | 0.9791 | AperiodicTemplate |
|  | 22 | 29 | 26 | 22 | 21 | 26 | 27 | 21 | 24 | 22 | 0.9642 | 0.9916 | AperiodicTemplate |
|  | 27 | 30 | 22 | 22 | 17 | 19 | 31 | 29 | 20 | 23 | 0.4296 | 0.9791 | AperiodicTemplate |
|  | 32 | 28 | 25 | 29 | 24 | 25 | 17 | 21 | 20 | 19 | 0.4765 | 0.9875 | AperiodicTemplate |
|  | 32 | 24 | 25 | 19 | 25 | 20 | 26 | 25 | 20 | 24 | 0.8043 | 0.9833 | AperiodicTemplate |
|  | 17 | 18 | 22 | 24 | 27 | 23 | 28 | 31 | 27 | 23 | 0.6111 | 0.9958 | AperiodicTemplate |
|  | 31 | 17 | 20 | 33 | 20 | 30 | 32 | 22 | 17 | 18 | 0.0541 | 0.9916 | AperiodicTemplate |
|  | 16 | 24 | 25 | 21 | 20 | 18 | 25 | 31 | 31 | 29 | 0.3178 | 0.9875 | AperiodicTemplate |
|  | 26 | 25 | 18 | 23 | 32 | 19 | 24 | 22 | 22 | 29 | 0.6544 | 0.9916 | AperiodicTemplate |
|  | 25 | 30 | 26 | 29 | 22 | 14 | 28 | 26 | 22 | 18 | 0.3852 | 0.9875 | AperiodicTemplate |
|  | 26 | 28 | 22 | 27 | 16 | 22 | 31 | 24 | 28 | 16 | 0.3852 | 0.9916 | AperiodicTemplate |
|  | 19 | 28 | 25 | 28 | 26 | 18 | 17 | 28 | 20 | 31 | 0.3924 | 0.9916 | AperiodicTemplate |
|  | 22 | 24 | 32 | 23 | 20 | 24 | 19 | 28 | 24 | 24 | 0.8119 | 0.9916 | AperiodicTemplate |
|  | 25 | 18 | 34 | 26 | 24 | 18 | 23 | 24 | 26 | 22 | 0.5595 | 0.9958 | AperiodicTemplate |
|  | 16 | 28 | 15 | 22 | 32 | 22 | 22 | 21 | 41 | 21 | 0.0069 | 0.9916 | AperiodicTemplate |
|  | 20 | 24 | 26 | 26 | 26 | 21 | 20 | 15 | 31 | 31 | 0.3781 | 0.9958 | AperiodicTemplate |
|  | 31 | 22 | 24 | 15 | 22 | 25 | 30 | 29 | 25 | 17 | 0.3178 | 0.9916 | AperiodicTemplate |
|  | 25 | 29 | 22 | 17 | 30 | 28 | 22 | 29 | 21 | 17 | 0.4296 | 0.9916 | AperiodicTemplate |
|  | 17 | 29 | 21 | 32 | 25 | 20 | 29 | 18 | 22 | 27 | 0.3572 | 0.9833 | AperiodicTemplate |
|  | 23 | 22 | 20 | 22 | 19 | 27 | 28 | 23 | 26 | 30 | 0.8485 | 0.9916 | AperiodicTemplate |
|  | 21 | 25 | 15 | 22 | 26 | 26 | 27 | 28 | 26 | 24 | 0.7887 | 0.9958 | AperiodicTemplate |
|  | 26 | 25 | 29 | 23 | 24 | 27 | 21 | 21 | 26 | 18 | 0.9058 | 0.9791 | AperiodicTemplate |
|  | 27 | 15 | 36 | 24 | 21 | 20 | 33 | 24 | 23 | 17 | 0.0618 | 0.9833 | AperiodicTemplate |
|  | 18 | 30 | 28 | 21 | 26 | 20 | 24 | 25 | 25 | 23 | 0.8343 | 1.0 | AperiodicTemplate |
|  | 19 | 18 | 20 | 31 | 33 | 18 | 30 | 21 | 24 | 26 | 0.2040 | 0.9916 | AperiodicTemplate |
|  | 30 | 24 | 28 | 20 | 22 | 28 | 25 | 29 | 18 | 16 | 0.4450 | 0.9958 | AperiodicTemplate |
|  | 26 | 14 | 20 | 23 | 18 | 31 | 21 | 26 | 33 | 28 | 0.1552 | 0.975 | AperiodicTemplate |
|  | 28 | 32 | 25 | 23 | 29 | 17 | 16 | 24 | 22 | 24 | 0.4070 | 0.9916 | AperiodicTemplate |
|  | 24 | 25 | 20 | 22 | 19 | 22 | 26 | 27 | 21 | 34 | 0.6197 | 0.9958 | AperiodicTemplate |
|  | 26 | 20 | 29 | 24 | 27 | 24 | 24 | 25 | 24 | 17 | 0.8881 | 0.9958 | AperiodicTemplate |
|  | 24 | 25 | 20 | 23 | 22 | 28 | 35 | 19 | 22 | 22 | 0.5341 | 0.9875 | AperiodicTemplate |
|  | 23 | 23 | 20 | 32 | 29 | 25 | 18 | 20 | 30 | 20 | 0.4527 | 0.9916 | AperiodicTemplate |
|  | 20 | 28 | 24 | 23 | 22 | 18 | 28 | 23 | 23 | 31 | 0.7564 | 0.9875 | AperiodicTemplate |
|  | 17 | 14 | 32 | 24 | 30 | 27 | 29 | 18 | 27 | 22 | 0.1283 | 0.9791 | AperiodicTemplate |
|  | 15 | 22 | 23 | 26 | 35 | 15 | 26 | 31 | 21 | 26 | 0.0932 | 1.0 | AperiodicTemplate |
|  | 19 | 20 | 24 | 20 | 35 | 18 | 21 | 22 | 30 | 31 | 0.1626 | 0.9833 | AperiodicTemplate |
|  | 20 | 19 | 16 | 29 | 30 | 28 | 22 | 16 | 26 | 34 | 0.0980 | 0.9875 | AperiodicTemplate |
|  | 17 | 20 | 25 | 34 | 27 | 22 | 26 | 22 | 15 | 32 | 0.1283 | 0.9833 | AperiodicTemplate |
|  | 25 | 19 | 26 | 39 | 33 | 21 | 14 | 13 | 28 | 22 | 0.0036 | 0.9833 | AperiodicTemplate |
|  | 16 | 20 | 23 | 28 | 22 | 20 | 30 | 26 | 23 | 32 | 0.4145 | 0.9958 | AperiodicTemplate |
|  | 24 | 26 | 23 | 18 | 28 | 30 | 24 | 17 | 31 | 19 | 0.4372 | 0.9833 | AperiodicTemplate |
|  | 27 | 19 | 21 | 23 | 34 | 19 | 20 | 25 | 21 | 31 | 0.3371 | 1.0 | AperiodicTemplate |
|  | 20 | 25 | 23 | 26 | 19 | 20 | 26 | 31 | 32 | 18 | 0.4372 | 1.0 | AperiodicTemplate |
|  | 23 | 21 | 30 | 20 | 26 | 31 | 19 | 20 | 29 | 21 | 0.5425 | 0.9958 | AperiodicTemplate |
|  | 20 | 13 | 27 | 31 | 34 | 24 | 22 | 24 | 18 | 27 | 0.1109 | 0.9916 | AperiodicTemplate |
|  | 31 | 26 | 23 | 27 | 29 | 17 | 23 | 18 | 18 | 28 | 0.3997 | 0.9875 | AperiodicTemplate |
|  | 23 | 15 | 25 | 19 | 24 | 22 | 31 | 29 | 27 | 25 | 0.5174 | 0.9875 | AperiodicTemplate |
|  | 21 | 24 | 20 | 30 | 28 | 19 | 22 | 27 | 27 | 22 | 0.8043 | 0.9833 | AperiodicTemplate |
|  | 20 | 14 | 25 | 18 | 25 | 28 | 26 | 30 | 26 | 28 | 0.3852 | 0.9875 | AperiodicTemplate |
|  | 22 | 22 | 22 | 35 | 20 | 19 | 22 | 29 | 29 | 20 | 0.3371 | 0.9916 | AperiodicTemplate |
|  | 20 | 22 | 17 | 26 | 28 | 22 | 24 | 23 | 29 | 29 | 0.7399 | 0.9958 | AperiodicTemplate |
|  | 27 | 24 | 27 | 18 | 19 | 25 | 33 | 22 | 28 | 17 | 0.3852 | 0.9916 | AperiodicTemplate |
|  | 31 | 29 | 24 | 23 | 28 | 20 | 24 | 18 | 22 | 21 | 0.6890 | 0.9916 | AperiodicTemplate |
|  | 20 | 28 | 24 | 21 | 28 | 25 | 12 | 28 | 29 | 25 | 0.3371 | 0.9916 | AperiodicTemplate |
|  | 21 | 25 | 13 | 23 | 23 | 25 | 23 | 33 | 28 | 26 | 0.3641 | 0.9958 | AperiodicTemplate |
|  | 32 | 22 | 36 | 18 | 26 | 19 | 24 | 16 | 22 | 25 | 0.1082 | 0.9666 | AperiodicTemplate |
|  | 18 | 28 | 18 | 30 | 26 | 32 | 20 | 24 | 23 | 21 | 0.4296 | 0.9916 | AperiodicTemplate |
|  | 21 | 24 | 25 | 15 | 27 | 30 | 23 | 24 | 29 | 22 | 0.6457 | 0.9916 | AperiodicTemplate |
|  | 26 | 26 | 29 | 24 | 17 | 18 | 28 | 23 | 28 | 21 | 0.6717 | 0.9791 | AperiodicTemplate |
|  | 29 | 22 | 20 | 15 | 24 | 31 | 27 | 21 | 26 | 25 | 0.5091 | 0.9833 | AperiodicTemplate |
|  | 23 | 24 | 29 | 23 | 33 | 15 | 30 | 27 | 14 | 22 | 0.1193 | 0.9875 | AperiodicTemplate |
|  | 23 | 24 | 23 | 33 | 23 | 19 | 23 | 27 | 24 | 21 | 0.8043 | 0.9875 | AperiodicTemplate |
|  | 22 | 22 | 29 | 23 | 32 | 19 | 21 | 24 | 22 | 26 | 0.7564 | 0.9916 | AperiodicTemplate |
|  | 24 | 19 | 24 | 35 | 20 | 24 | 19 | 29 | 21 | 25 | 0.4145 | 0.9875 | AperiodicTemplate |
|  | 19 | 21 | 20 | 22 | 23 | 22 | 16 | 41 | 32 | 24 | 0.0189 | 0.9958 | AperiodicTemplate |
|  | 35 | 21 | 20 | 24 | 23 | 23 | 30 | 19 | 17 | 28 | 0.2482 | 0.9833 | AperiodicTemplate |
|  | 20 | 23 | 18 | 34 | 24 | 20 | 25 | 28 | 23 | 25 | 0.5510 | 0.9916 | AperiodicTemplate |
|  | 27 | 30 | 23 | 24 | 24 | 30 | 21 | 20 | 14 | 27 | 0.4372 | 0.9875 | AperiodicTemplate |
|  | 37 | 24 | 32 | 20 | 31 | 23 | 22 | 23 | 19 | 9 | 0.0060 | 0.9791 | AperiodicTemplate |
|  | 22 | 26 | 27 | 16 | 21 | 18 | 29 | 31 | 28 | 22 | 0.4220 | 0.9916 | AperiodicTemplate |
|  | 29 | 31 | 34 | 11 | 17 | 26 | 20 | 20 | 21 | 31 | 0.0164 | 0.9791 | AperiodicTemplate |
|  | 22 | 25 | 33 | 27 | 17 | 26 | 18 | 23 | 28 | 21 | 0.4606 | 0.9875 | AperiodicTemplate |
|  | 34 | 23 | 15 | 27 | 21 | 26 | 23 | 21 | 24 | 26 | 0.4296 | 0.9833 | AperiodicTemplate |
|  | 22 | 24 | 24 | 29 | 19 | 29 | 25 | 20 | 31 | 17 | 0.5257 | 1.0 | AperiodicTemplate |
|  | 28 | 15 | 20 | 23 | 31 | 22 | 14 | 35 | 25 | 27 | 0.0556 | 0.9958 | AperiodicTemplate |
|  | 32 | 23 | 19 | 25 | 25 | 25 | 24 | 23 | 23 | 21 | 0.8881 | 0.9958 | AperiodicTemplate |
|  | 35 | 25 | 27 | 26 | 29 | 20 | 28 | 16 | 17 | 17 | 0.0980 | 0.975 | AperiodicTemplate |
|  | 24 | 27 | 20 | 27 | 31 | 26 | 26 | 21 | 19 | 19 | 0.7146 | 0.9916 | AperiodicTemplate |
|  | 30 | 19 | 38 | 16 | 22 | 18 | 25 | 18 | 23 | 31 | 0.0281 | 0.9833 | AperiodicTemplate |
|  | 24 | 23 | 32 | 18 | 24 | 18 | 23 | 28 | 27 | 23 | 0.6544 | 0.9916 | AperiodicTemplate |
|  | 20 | 24 | 25 | 22 | 22 | 35 | 24 | 24 | 21 | 23 | 0.6890 | 0.9916 | AperiodicTemplate |
|  | 18 | 20 | 34 | 26 | 20 | 31 | 20 | 25 | 27 | 19 | 0.2535 | 0.9916 | AperiodicTemplate |
|  | 24 | 24 | 27 | 22 | 20 | 25 | 27 | 23 | 27 | 21 | 0.9830 | 0.9958 | AperiodicTemplate |
|  | 27 | 20 | 28 | 20 | 19 | 23 | 36 | 23 | 23 | 21 | 0.3572 | 0.9875 | AperiodicTemplate |
|  | 23 | 21 | 19 | 29 | 27 | 31 | 20 | 25 | 23 | 22 | 0.7564 | 0.9958 | AperiodicTemplate |
|  | 21 | 27 | 24 | 19 | 22 | 28 | 22 | 21 | 28 | 28 | 0.8755 | 0.9958 | AperiodicTemplate |
|  | 20 | 28 | 19 | 22 | 24 | 23 | 22 | 28 | 30 | 24 | 0.8415 | 0.9875 | AperiodicTemplate |
|  | 29 | 22 | 22 | 34 | 19 | 25 | 20 | 26 | 24 | 19 | 0.4846 | 0.9916 | AperiodicTemplate |
|  | 20 | 22 | 35 | 26 | 18 | 13 | 33 | 26 | 28 | 19 | 0.0371 | 0.9875 | AperiodicTemplate |
|  | 21 | 24 | 24 | 26 | 24 | 21 | 27 | 22 | 26 | 25 | 0.9957 | 0.9916 | AperiodicTemplate |
|  | 26 | 24 | 25 | 19 | 19 | 20 | 20 | 36 | 20 | 31 | 0.1951 | 0.9916 | AperiodicTemplate |
|  | 16 | 26 | 13 | 27 | 23 | 25 | 27 | 33 | 25 | 25 | 0.2040 | 1.0 | AperiodicTemplate |
|  | 30 | 14 | 23 | 26 | 23 | 24 | 24 | 35 | 21 | 20 | 0.2133 | 0.9875 | AperiodicTemplate |
|  | 23 | 20 | 23 | 21 | 22 | 24 | 31 | 28 | 22 | 26 | 0.8881 | 0.9958 | AperiodicTemplate |
|  | 16 | 31 | 21 | 21 | 18 | 32 | 16 | 24 | 27 | 34 | 0.0513 | 0.9791 | AperiodicTemplate |
|  | 20 | 27 | 25 | 19 | 20 | 23 | 28 | 22 | 26 | 30 | 0.8043 | 0.9833 | AperiodicTemplate |
|  | 13 | 19 | 22 | 18 | 26 | 26 | 30 | 29 | 27 | 30 | 0.1865 | 0.9958 | AperiodicTemplate |
|  | 29 | 28 | 18 | 20 | 25 | 23 | 29 | 26 | 24 | 18 | 0.6717 | 0.9791 | AperiodicTemplate |
|  | 30 | 32 | 21 | 21 | 19 | 28 | 17 | 26 | 29 | 17 | 0.2180 | 0.9791 | AperiodicTemplate |
|  | 18 | 31 | 21 | 22 | 28 | 21 | 29 | 25 | 22 | 23 | 0.6976 | 0.9958 | AperiodicTemplate |
|  | 23 | 19 | 29 | 25 | 19 | 33 | 27 | 27 | 23 | 15 | 0.2932 | 0.9875 | AperiodicTemplate |
|  | 19 | 18 | 27 | 29 | 30 | 29 | 24 | 24 | 27 | 13 | 0.2180 | 0.9916 | AperiodicTemplate |
|  | 29 | 21 | 26 | 25 | 18 | 17 | 33 | 23 | 31 | 17 | 0.1782 | 0.9916 | AperiodicTemplate |
|  | 31 | 27 | 24 | 34 | 22 | 19 | 21 | 21 | 21 | 20 | 0.3852 | 0.9833 | AperiodicTemplate |
|  | 33 | 19 | 28 | 22 | 28 | 26 | 22 | 21 | 26 | 15 | 0.3371 | 0.9791 | AperiodicTemplate |
|  | 18 | 28 | 28 | 23 | 30 | 19 | 26 | 24 | 24 | 20 | 0.7146 | 0.9916 | AperiodicTemplate |
|  | 24 | 24 | 25 | 31 | 19 | 25 | 26 | 16 | 24 | 26 | 0.7231 | 0.9791 | AperiodicTemplate |
|  | 15 | 34 | 27 | 28 | 18 | 27 | 22 | 18 | 29 | 22 | 0.1480 | 0.9875 | AperiodicTemplate |
|  | 20 | 29 | 17 | 19 | 28 | 31 | 27 | 19 | 32 | 18 | 0.1588 | 0.9916 | AperiodicTemplate |
|  | 28 | 31 | 18 | 15 | 28 | 24 | 29 | 26 | 23 | 18 | 0.2757 | 0.9916 | AperiodicTemplate |
|  | 27 | 37 | 26 | 22 | 22 | 24 | 19 | 17 | 25 | 21 | 0.2482 | 0.9958 | AperiodicTemplate |
|  | 19 | 22 | 29 | 26 | 19 | 24 | 29 | 25 | 27 | 20 | 0.7807 | 0.9916 | AperiodicTemplate |
|  | 30 | 20 | 25 | 29 | 29 | 18 | 28 | 26 | 14 | 21 | 0.2644 | 0.975 | AperiodicTemplate |
|  | 19 | 19 | 17 | 22 | 23 | 31 | 29 | 21 | 27 | 32 | 0.2873 | 0.9958 | AperiodicTemplate |
|  | 25 | 16 | 22 | 27 | 21 | 25 | 29 | 30 | 21 | 24 | 0.6804 | 0.9791 | AperiodicTemplate |
|  | 20 | 29 | 16 | 32 | 25 | 24 | 17 | 28 | 22 | 27 | 0.3241 | 0.9875 | AperiodicTemplate |
|  | 27 | 24 | 27 | 18 | 19 | 25 | 33 | 22 | 28 | 17 | 0.3852 | 0.9916 | AperiodicTemplate |
|  | 22 | 23 | 23 | 28 | 29 | 32 | 24 | 22 | 21 | 16 | 0.5510 | 0.9916 | AperiodicTemplate |
|  | 25 | 20 | 19 | 17 | 36 | 26 | 28 | 24 | 19 | 26 | 0.2228 | 0.9875 | AperiodicTemplate |
|  | 22 | 22 | 29 | 29 | 20 | 24 | 29 | 22 | 26 | 17 | 0.6890 | 1.0 | AperiodicTemplate |
|  | 19 | 35 | 28 | 16 | 33 | 24 | 25 | 23 | 9 | 28 | 0.0063 | 0.9875 | AperiodicTemplate |
|  | 21 | 21 | 20 | 28 | 24 | 27 | 25 | 25 | 26 | 23 | 0.9733 | 0.9916 | AperiodicTemplate |
|  | 30 | 24 | 26 | 21 | 15 | 26 | 20 | 20 | 31 | 27 | 0.4070 | 0.9916 | AperiodicTemplate |
|  | 24 | 22 | 23 | 28 | 24 | 19 | 21 | 27 | 30 | 22 | 0.8881 | 0.9875 | AperiodicTemplate |
|  | 32 | 19 | 23 | 20 | 29 | 25 | 22 | 19 | 25 | 26 | 0.6457 | 0.9666 | AperiodicTemplate |
|  | 14 | 29 | 24 | 28 | 24 | 20 | 23 | 18 | 30 | 30 | 0.2700 | 0.9916 | AperiodicTemplate |
|  | 26 | 22 | 22 | 29 | 18 | 23 | 39 | 21 | 11 | 29 | 0.0130 | 0.9875 | AperiodicTemplate |
|  | 20 | 23 | 27 | 23 | 21 | 30 | 24 | 23 | 28 | 21 | 0.9058 | 0.9958 | AperiodicTemplate |
|  | 21 | 22 | 29 | 22 | 27 | 30 | 23 | 26 | 24 | 16 | 0.6890 | 0.9875 | AperiodicTemplate |
|  | 22 | 26 | 27 | 24 | 20 | 25 | 25 | 27 | 20 | 24 | 0.9808 | 0.9958 | AperiodicTemplate |
|  | 23 | 30 | 21 | 20 | 28 | 22 | 20 | 33 | 24 | 19 | 0.4846 | 0.9916 | AperiodicTemplate |
|  | 24 | 24 | 30 | 28 | 24 | 16 | 29 | 25 | 16 | 24 | 0.4765 | 0.9916 | AperiodicTemplate |
|  | 27 | 25 | 18 | 25 | 29 | 24 | 25 | 22 | 21 | 24 | 0.9366 | 0.9916 | AperiodicTemplate |
|  | 17 | 26 | 17 | 17 | 33 | 32 | 25 | 25 | 24 | 24 | 0.1908 | 1.0 | AperiodicTemplate |
|  | 26 | 22 | 26 | 18 | 20 | 34 | 26 | 20 | 24 | 24 | 0.5680 | 0.9875 | AperiodicTemplate |
|  | 24 | 18 | 23 | 23 | 27 | 27 | 21 | 16 | 26 | 35 | 0.3053 | 0.9875 | AperiodicTemplate |
|  | 20 | 31 | 21 | 28 | 28 | 23 | 28 | 21 | 21 | 19 | 0.6457 | 0.9875 | AperiodicTemplate |
|  | 22 | 28 | 22 | 21 | 18 | 23 | 26 | 22 | 32 | 26 | 0.7315 | 0.9833 | AperiodicTemplate |
|  | 25 | 26 | 14 | 27 | 27 | 28 | 26 | 29 | 18 | 20 | 0.4220 | 0.9875 | AperiodicTemplate |
|  | 17 | 29 | 23 | 24 | 19 | 28 | 21 | 21 | 30 | 28 | 0.5595 | 0.9958 | AperiodicTemplate |
|  | 23 | 26 | 26 | 26 | 24 | 19 | 21 | 27 | 23 | 25 | 0.9830 | 0.9916 | AperiodicTemplate |
|  | 27 | 31 | 21 | 25 | 27 | 25 | 20 | 22 | 23 | 19 | 0.8195 | 0.9958 | AperiodicTemplate |
|  | 24 | 27 | 30 | 23 | 28 | 29 | 18 | 16 | 23 | 22 | 0.5341 | 0.9833 | AperiodicTemplate |
|  | 23 | 24 | 30 | 21 | 22 | 19 | 24 | 23 | 32 | 22 | 0.7399 | 0.9833 | AperiodicTemplate |
|  | 22 | 27 | 27 | 20 | 23 | 17 | 24 | 22 | 28 | 30 | 0.7399 | 0.9916 | AperiodicTemplate |
|  | 29 | 26 | 18 | 26 | 33 | 26 | 15 | 15 | 23 | 29 | 0.1137 | 0.9875 | ApproximateEntropy |
|  | 17 | 32 | 25 | 24 | 24 | 29 | 24 | 19 | 27 | 19 | 0.5091 | 0.9875 | BlockFrequency |
|  | 28 | 27 | 17 | 22 | 24 | 27 | 25 | 20 | 25 | 25 | 0.8819 | 0.9875 | CumulativeSums |
|  | 27 | 20 | 23 | 27 | 19 | 25 | 36 | 28 | 16 | 19 | 0.1664 | 0.9833 | CumulativeSums |
|  | 34 | 27 | 22 | 30 | 25 | 16 | 26 | 22 | 17 | 21 | 0.2327 | 0.9833 | FFT |
|  | 26 | 20 | 24 | 21 | 28 | 34 | 14 | 25 | 29 | 19 | 0.1951 | 0.9875 | Frequency |
|  | 24 | 27 | 26 | 26 | 18 | 25 | 12 | 23 | 31 | 28 | 0.2757 | 0.9916 | LinearComplexity |
|  | 27 | 24 | 29 | 27 | 23 | 17 | 30 | 15 | 27 | 21 | 0.3924 | 0.9916 | LongestRun |
|  | 27 | 27 | 22 | 24 | 21 | 25 | 24 | 22 | 26 | 22 | 0.9938 | 0.9958 | PeriodicTemplate |
|  | 16 | 15 | 11 | 15 | 16 | 13 | 13 | 12 | 15 | 15 | 0.9925 | 0.9787 | RandomExcursion |
|  | 14 | 15 | 12 | 17 | 9 | 16 | 14 | 16 | 11 | 17 | 0.8642 | 0.9787 | RandomExcursion |
|  | 9 | 16 | 17 | 14 | 20 | 15 | 9 | 13 | 12 | 16 | 0.5558 | 1.0 | RandomExcursion |
|  | 12 | 21 | 13 | 15 | 10 | 17 | 16 | 11 | 11 | 15 | 0.5999 | 1.0 | RandomExcursion |
|  | 20 | 15 | 10 | 15 | 13 | 9 | 16 | 14 | 18 | 11 | 0.5558 | 1.0 | RandomExcursion |
|  | 20 | 11 | 18 | 13 | 11 | 10 | 21 | 18 | 5 | 14 | 0.0546 | 1.0 | RandomExcursion |
|  | 15 | 16 | 23 | 11 | 13 | 13 | 14 | 14 | 12 | 10 | 0.4985 | 0.9929 | RandomExcursion |
|  | 10 | 18 | 18 | 18 | 18 | 15 | 14 | 9 | 10 | 11 | 0.4049 | 0.9929 | RandomExcursion |
|  | 11 | 15 | 21 | 14 | 15 | 20 | 14 | 15 | 10 | 6 | 0.1794 | 0.9858 | RandomExcursion |
|  | 13 | 16 | 18 | 11 | 19 | 13 | 14 | 18 | 9 | 10 | 0.5269 | 0.9929 | RandomExcursion |
|  | 16 | 8 | 18 | 12 | 15 | 13 | 10 | 15 | 19 | 15 | 0.5852 | 0.9716 | RandomExcursion |
|  | 16 | 14 | 18 | 9 | 14 | 9 | 15 | 18 | 13 | 15 | 0.7037 | 0.9787 | RandomExcursion |
|  | 13 | 15 | 23 | 12 | 13 | 11 | 11 | 17 | 14 | 12 | 0.4846 | 0.9858 | RandomExcursion |
|  | 9 | 20 | 13 | 23 | 10 | 14 | 12 | 15 | 15 | 10 | 0.1658 | 0.9858 | RandomExcursion |
|  | 13 | 17 | 10 | 13 | 17 | 12 | 12 | 15 | 15 | 17 | 0.9160 | 0.9787 | RandomExcursion |
|  | 19 | 10 | 12 | 15 | 8 | 15 | 18 | 17 | 16 | 11 | 0.4708 | 0.9787 | RandomExcursion |
|  | 18 | 8 | 20 | 14 | 11 | 13 | 18 | 11 | 14 | 14 | 0.4572 | 0.9929 | RandomExcursion |
|  | 12 | 15 | 18 | 15 | 13 | 16 | 16 | 11 | 9 | 16 | 0.8404 | 1.0 | RandomExcursion |
|  | 12 | 13 | 18 | 12 | 22 | 15 | 12 | 11 | 14 | 12 | 0.5704 | 0.9716 | RandomExcursion |
|  | 14 | 17 | 16 | 10 | 12 | 18 | 15 | 12 | 14 | 13 | 0.9160 | 0.9787 | RandomExcursion |
|  | 13 | 14 | 19 | 11 | 12 | 19 | 17 | 14 | 13 | 9 | 0.6296 | 0.9929 | RandomExcursion |
|  | 10 | 12 | 19 | 15 | 13 | 16 | 18 | 15 | 11 | 12 | 0.7611 | 0.9929 | RandomExcursion |
|  | 6 | 14 | 12 | 19 | 12 | 12 | 16 | 15 | 18 | 17 | 0.4049 | 1.0 | RandomExcursion |
|  | 11 | 17 | 11 | 16 | 11 | 11 | 19 | 14 | 12 | 19 | 0.5999 | 0.9929 | RandomExcursion |
|  | 12 | 11 | 17 | 21 | 12 | 15 | 14 | 13 | 12 | 14 | 0.7611 | 1.0 | RandomExcursion |
|  | 16 | 8 | 12 | 15 | 16 | 26 | 13 | 17 | 10 | 8 | 0.0327 | 0.9929 | RandomExcursion |
|  | 26 | 16 | 28 | 25 | 21 | 23 | 25 | 25 | 27 | 24 | 0.8819 | 0.9833 | Rank |
|  | 29 | 24 | 24 | 18 | 19 | 27 | 30 | 14 | 23 | 32 | 0.1951 | 0.9916 | Runs |
|  | 27 | 32 | 22 | 23 | 27 | 20 | 21 | 27 | 23 | 18 | 0.6804 | 0.9916 | Serial |
|  | 24 | 17 | 34 | 25 | 21 | 20 | 22 | 26 | 23 | 28 | 0.5009 | 0.9958 | Serial |
|  | 25 | 36 | 25 | 24 | 17 | 31 | 23 | 23 | 16 | 20 | 0.1379 | 0.9916 | Universal |