ЛАБОРАТОРНАЯ РАБОТА № 2

Генерирование равномерно распределенных псевдослучайных последовательностей

Цель работы

Освоить основные алгоритмы программного генерирования равномерно распределенных псевдослучайных последовательностей.

Конгруэнтные генераторы.

Линейным конгруэнтным генератором (ЛКГ) с параметрами (x_0, a, c, N) называется программный генератор РРСП, порождающий псевдослучайную последовательность $x_1, x_2, \ldots \in A$, $A = \{0,1, \ldots, N-1\}$ с помощью рекуррентного соотношения:

$$x_{t+1} = (ax_t + c) \bmod N, \; t = 0,1\ldots \qquad \textbf{(1)}$$

Параметры этого генератора (1) имеют следующий смысл: $x_0 \in A$ – начальное, или стартовое, значение; $a \in A \setminus \{0\}$ – ненулевой множитель; $c \in A$ – приращение; N – модуль, равный мощности алфавита A.

Если приращение $c = 0$, то генератор (1) называется мультипликативным конгруэнтным генератором (МКГ), а если $c \neq 0$, то смешанным конгруэнтным генератором (СКГ).

Перечислим свойства псевдослучайной последовательности, порождаемой ЛКГ:

1. Псевдослучайная последовательность (1), порождаемая ЛКГ, достигает максимального значения периода $T_{\max} = N$ тогда и только тогда, когда выполнены следующие три условия:
   a) c, N – взаимно простые, т.е. $НОД(c, N) = 1$;
   b) число $b = a - 1$ кратно p для любого простого числа $p < N$, являющегося делителем N;
   c) число b кратно *4*, если N кратно *4*.
2. Для МКГ, если x_0, N – взаимно простые, a – первообразный элемент по модулю N, а $\varphi(N)$ – максимально возможный порядок по модулю N, то псевдослучайная последовательность имеет максимальный период $T_{\max}$.

3. Для МКГ, если $N = 2^q$, $q \geq 4$, то максимально возможное значение периода $T_{\max} = 2^{q-2} = \frac{N}{4}$ псевдослучайной последовательности достигается, если $x_0 \geq 1$ – нечетно и вычет $a \bmod 8 \in \{3,5\}$.

4. «Слабость» ЛКГ и МКГ заключается в том, что если рассматривать последовательные биграммы $\left(z_1^{(t)}, z_2^{(t)}\right)$: $z_1^{(t)} = x_t$, $z_2^{(t)} = x_{t-1}$, то точки $z^t = \left(z_1^{(t)}, z_2^{(t)}\right)$, $t = 1,2\ldots$ на плоскости R^2 будут лежать на прямых из семейства $z_2 = az_1 + c - kN$, $k = 0,1\ldots$

Нелинейные конгруэнтные генераторы.

Четвертое свойство линейного и мультипликативного конгруэнтных генераторов псевдослучайных последовательностей представляет «слабость» этих генераторов и может активно использоваться для построения криптоатак в целях оценки параметров a, c, x_0. Для устранения этого недостатка используют нелинейные конгруэнтные генераторы псевдослучайных последовательностей. Наибольшее распространение получили три подхода, описание которых приводится ниже.

Квадратичные конгруэнтные генераторы.

Этот алгоритм генерации псевдослучайной последовательности $x_t \in A = \{0,1,\ldots,N-1\}$ определяется квадратичным рекуррентным соотношением:

$$x_{t+1} = \left(dx_t^2 + ax_t + c\right) \bmod N,\ t = 0,1\ldots \qquad \textbf{(2)}$$

где $x_0, a, c, d \in A$ – параметры генератора. Выбор этих параметров осуществляется на основе следующих двух свойств последовательности (2):

1. Квадратичная конгруэнтная последовательность (2) имеет наибольший период $T_{\max} = N$ тогда и только тогда, когда выполнены следующие условия:

a) c, N – взаимно простые числа;

b) $d, a - 1$ – кратны p, где p — любой нечетный простой делитель N;

c) d – четное число, причем

$$d = \begin{cases} (a-1) \bmod 4, & \text{если N кратно 4} \\ (a-1) \bmod 2, & \text{если N кратно 2} \end{cases}$$

d) если N кратно 9, то либо $d \bmod 9 = 0$, либо $d \bmod 9 = 1$ и $cd \bmod 9 = 6$.

e) Если $N = 2^q, q \geq 2$ то наибольший период $T_{\max} = 2^q$ тогда и только тогда, когда c – нечетно, d – четно, a – нечетное число, удовлетворяющее соотношению: $a = (d + 1) \bmod 4$.

Генератор Эйхенауэра – Лена с обращением.

Псевдослучайная нелинейная конгруэнтная последовательность Эйхенауэра – Лена с обращением определяется следующим нелинейным рекуррентным соотношением:

$$x_{t+1} = \begin{cases} \left(ax_t^{-1} + c\right) \bmod N, & \text{если } x_t \geq 1 \\ c, & \text{если } x_t = 0 \end{cases} \qquad \textbf{(3)}$$

где x_t^{-1} – обратный к x_t элемент по модулю N, т.е. $x_t x_t^{-1} \equiv 1 (\bmod N)$; $x_0, a, c \in A$ – параметры генератора.

Выбор параметров осуществляется на основании свойства:

1. Если $N = 2^q$, a, x_0 – нечетны, c — четно, то генератор (3) имеет максимально возможный период $T_{\max} = 2^{q-1}$ тогда и только тогда, когда $a \equiv 1 (\bmod 4)$, $c \equiv 2 (\bmod 4)$

Конгруэнтный генератор, использующий умножение с переносом.

При этом нелинейная конгруэнтная псевдослучайная последовательность определяется рекуррентным соотношением:

$$x_{t+1} = (ax_t + c_t) \bmod N \qquad \textbf{(4)}$$

где, в отличие от (2), «приращение» $c_t = c(x_{t-1}, x_{t-2}, \ldots, x_0)$ изменяется во времени и зависит от указанных аргументов нелинейно:

$$c_t = \left\lfloor \frac{ax_{t-1} + c_{t-1}}{N} \right\rfloor^{1} \qquad \textbf{(5)}$$

Параметрами нелинейного конгруэнтного генератора (4), (5) являются x_0, c_0, a, N.

Рекуренты в конечном поле

Обобщением мультипликативной конгруэнтной последовательности является линейная рекуррентная последовательность порядка $k \geq 1$ над конечным полем $GF(p^k)$:

$$x_{t+1} = (a_1 x_t + a_2 x_{t-1} + \ldots + a_k x_{t-k+1}) \bmod p \qquad \textbf{(6)}$$

где $a_1, \ldots, a_k \in A = \{0,1\ldots, p-1\}$ – коэффициенты рекурренты, а $x_0, \ldots, x_{-k+1} \in A$ – начальные значения рекурренты.

Параметры генератора псевдослучайной последовательности (6): $p, k, a_1, \ldots, a_{-k+1}$. Начальные значения $x_0, \ldots, x_{-k+1} \in A$ выбираются произвольно так, чтобы не обращались в ноль одновременно. Коэффициенты рекурренты $a_1, \ldots, a_k \in A$ выбираются таким образом, чтобы порождающий полином

$$f(x) = x^k - a_1 x^{k-1} - \ldots - a_{k-1}x - a_k \qquad \textbf{(7)}$$

являлся примитивным многочленом по модулю p, т.е. многочлен (7) имел корень x_*, являющийся первообразным элементом поля $GF(p^k)$. При таком

---

[1] Наибольшее целое, меньшее или равное числу в скобках.

выборе параметров достигается максимально возможный период $T_{max} = p^k - 1$ псевдослучайной последовательности (6).

Последовательности, порождаемые линейными регистрами сдвига с обратной связью.

Линейным регистром сдвига с обратной связью (Linear Feedback Shift Register, сокращенно LFSR) называется логическое устройство, схема которого изображена на рис. 1.

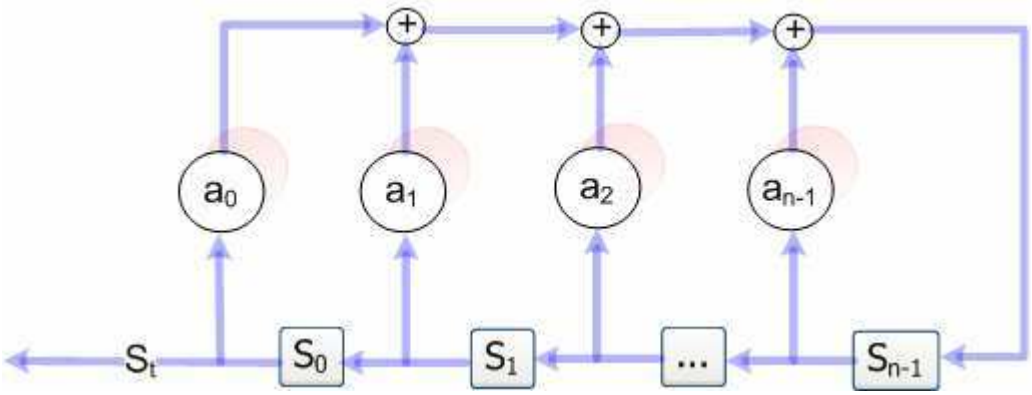

Рис. 1. Блок-схема LFSR.

LFSR состоит из n ячеек памяти, двоичные состояния которых в момент времени $t = 0,1,\ldots$ характеризуются значениями $S_0(t)$, $S_1(t)$, $\ldots$, $S_{n-1}(t) \in A = \{0,1\}$. Выходы ячеек памяти связаны не только последовательно друг с другом, но и с сумматорами $\oplus$ в соответствии с коэффициентами передачи $a_0, a_1, \ldots, a_{n-1} \in A$: если $a_i = 1$, то значение $S_i(t)$ i-ой ячейки передается на один из входов i-го сумматора; если же $a_i = 0$, то такая передача отсутствует. Полагается $a_i \equiv 0$. Состояние LFSR в текущий момент времени t задается двоичным n-вектор-столбцом $S(t) = (S_{n-1}(t), \ldots, S_0(t))'$.

Содержание ячеек LFSR с течением времени изменяется следующим образом, определяя тем самым динамику состояний LFSR:

$$S_i(t+1) = \begin{cases} S_{i+1}(t), & \text{если} \quad i \in \overline{0, n-2} \\ \sum_{j=0}^{n-1} a_j S_j(t), & \text{если} \quad i = n-1 \end{cases} \qquad \textbf{(8)}$$

Текущие значения нулевой ячейки регистра используются в качестве элементов порождаемой LFSR двоичной псевдослучайной последовательности $s_y = S_0(t)$ (см. рис. 1).

Модель (8) является частным случаем модели (7) линейной рекурренты над полем $GF(2^n)$, поэтому коэффициенты $\{a_i\}$ выбираются согласно методике, приведенной в предыдущем пункте. То есть многочлен, по которому строится LFSR, должен быть примитивным по модулю *2*. Степень многочлена является длиной сдвигового регистра. Примитивный(базовый) многочлен степени n по модулю *2* – это неприводимый многочлен, который является делителем $x^{2^n-1} - 1$, но не является делителем $x^d - 1$ для всех d, являющихся делителями $2^n - 1$. Неприводимый многочлен степени n нельзя представить в виде умножения многочленов кроме него самого и единичного.

Генераторы Фибонначи.

Общий вид рекуррентного соотношения, определяющего генератор Фибоначчи, задается уравнением

$$x_t = x_{t-r} \oplus x_{t-s}, \; t = r, r+1, r+2 \ldots \qquad \textbf{(9)}$$

где $r, s \in N(r > s)$ – параметры генератора; элемент $x_t \in V_k$ представляет собой двоичный k-вектор и действие $\oplus$ выполняется покомпонентно.

Криптостойкие генераторы на основе односторонних функций.

Для повышения стойкости алгоритмов генерации псевдослучайных последовательностей к криптоанализу в последнее время предлагается синтезировать алгоритмы на основе известных в криптографии односторонних функций. Характерное свойство односторонних (one-way) функций состоит в

том, что для вычисления значения функции по заданному значению аргумента существует полиномиально-сложный алгоритм, в то время как для вычисления аргумента по заданному значению функции полиномиально-сложного алгоритма не существует (или он не известен). Доказательство свойства односторонности функции является трудной математической задачей, поэтому в настоящее время в криптосистемах часто используются «кандидаты в односторонние функции», для которых показано лишь, что в настоящее время не известны полиномиально-сложные алгоритмы вычисления обратной функции. Примерами таких «кандидатов» являются некоторые известные криптоалгоритмы (например, DES) и хэш-функции (например, SHA-1).

Генераторы, основанные на математическом аппарате односторонних функций: ANSI X9.17, FIPS-186, Yarrow-160.

Криптостойкие генераторы, основанные на проблемах теории чисел.

Стойкость данных генераторов псевдослучайных последовательностей основывается на неразрешимости с полиномиальной сложностью (на данный момент) некоторых известных проблем теории чисел: факторизации больших чисел и дискретного логарифмирования.

Примерами, генераторов основанных на данных проблемах являются RSA-алгоритм генерации псевдослучайных последовательностей, модификация Микали-Шнорра RSA-алгоритм генерации псевдослучайных последовательностей, BBS (Blum–Blum–Shub) – алгоритм генерации псевдослучайных последовательностей.

Методы «улучшения» элементарных псевдослучайных последовательностей.

Пусть $\xi_1, \xi_2, \ldots$ – некоторая двоичная псевдослучайная последовательность, сгенерированная одним из простейших методов и называемая поэтому (в данном пункте) элементарной. Для того чтобы

построить псевдослучайную последовательность со свойствами, более близкими к свойствам РРСГТ, чем элементарная последовательность, осуществим функциональное преобразование:

$$x_1 = f_1(\xi_1, \xi_2, \ldots), \; x_2 = f_2(\xi_1, \xi_2, \ldots), \ldots$$

где $f_1(\xi_1, \xi_2, \ldots), \; f_2(\xi_1, \xi_2, \ldots), \ldots$ – некоторые функционалы, которые следует подбирать так, чтобы преобразованная последовательность $\{x_k\}$ имела вероятностное распределение, более близкое к распределению РРСП, чем распределение $\{\xi_i\}$.

Выбирая различные функционалы и метрики в пространстве вероятностных распределений, можно разработать множество методов и алгоритмов «улучшения» элементарных псевдослучайных последовательностей.

Например, алгоритм симметризации псевдослучайных последовательностей.

Комбинирование алгоритмов генерации методом Макларена – Марсальи.

Пусть имеется два простейших генератора псевдослучайных последовательностей: G_1 и G_2. Генератор G_1 порождает «элементарную» последовательность над алфавитом мощности N: $x_0, x_1, \ldots \in A(N) = \{0,1, \ldots, N-1\}$, а генератор G_2 – над алфавитом мощности K: $y_0, y_1, \ldots \in A(K) = \{0,1, \ldots, K-1\}$.

Пусть имеется вспомогательная таблица $T = \{T(0), T(1), \ldots, T(K-1)\}$, из K целых чисел (память из K ячеек).

Метод Макларена – Марсальи комбинирования последовательностей $\{x_i\}$, $\{y_i\}$ для получения выходной псевдослучайной последовательности $\{z_k\}$ состоит в следующем. Сначала T-таблица заполняется K первыми членами последовательности $\{x_i\}$. Элементы выходной последовательности

вычисляются следующим образом:

$s \leftarrow y_k,\ z_k = T(s),\ T(s) \leftarrow x_{K+k},\ k = 0,1,\ldots$

Таким образом, генератор G_2 осуществляет «случайный» выбор из T-таблицы, а также ее «случайное» заполнение «случайными» числами, порождаемыми генератором G_1.

Метод комбинирования Макларена – Марсальи позволяет ослабить зависимость между членами $\{z_k\}$ и увеличить период псевдослучайной последовательности.

Комбинирование LFSR-генераторов.

LFSR-генераторы часто используются в качестве генераторов элементарных псевдослучайных последовательностей и применяются для комбинирования генераторов. Прежде всего отметим, что LFSR-генераторы можно использовать в качестве G_1, G_2 в генераторе Макларена – Марсальи. Например, одним из способов комбинирования LFSR-генераторов является полиномиальное комбинирование элементарных последовательностей. Общая модель комбинирования LFSR-генераторов представлена на рис. 2.

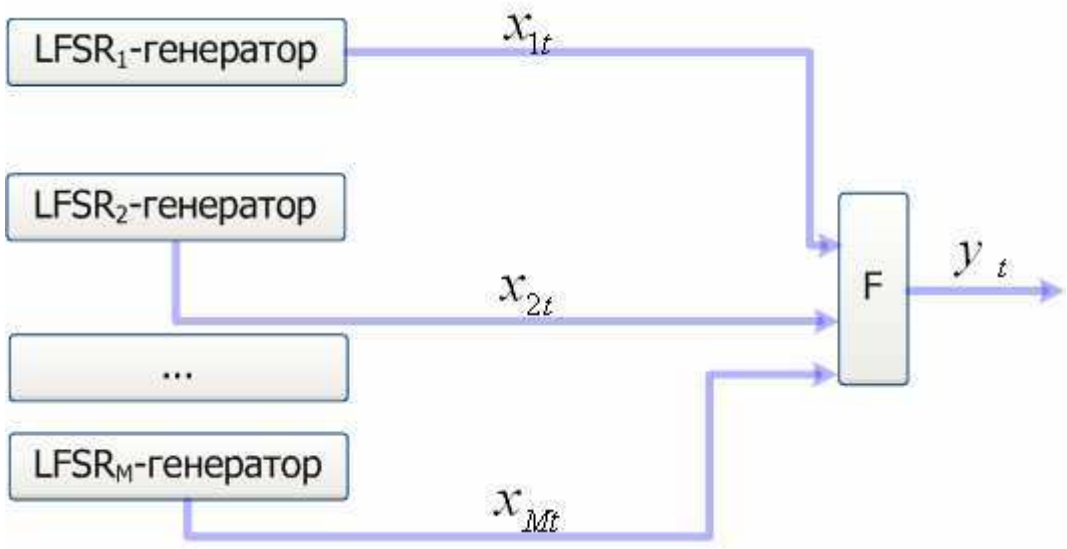

Рис. 2. Общая модель комбинирования LFSR-генераторов.

Здесь функция F общего полиномиального вида:

$$y = F(x) = \left(a_0 + \sum_{1 \le i \le M} a_i x_i + \sum_{1 \le i < j \le M} a_{ij} x_i x_j + \ldots + a_{12\ldots M} x_1 x_2 \ldots x_M \right) \bmod 2.$$

Комбинирование с помощью псевдослучайного прореживания

Рассмотрим еще один способ комбинирования двух LFSR-генераторов G_1, G_2. Пусть LFSR-генератор G_1 порождает «элементарную» двоичную последовательность $\{a_t\}$, а LFSR-генератор G_2 – двоичную «селектирующую» последовательность $\{s_t\}$. С помощью этих двух последовательностей $\{a_t\}$, $\{s_t\}$ строится выходная последовательность $\{x_t\}$, включающая те биты a_t, для которых соответствующее значение селектора $s_t = 1$; если $s_t = 0$, то значение a_t игнорируется. Такой генератор двоичной псевдослучайной последовательности называется SG-генератором.

Свойство SG-генератора выражается следующим утверждением: пусть T_a, T_s – соответственно периоды последовательностей $\{a_t\}$ и $\{s_t\}$. Если генераторы G_1, G_2 используют примитивные порождающие многочлены степеней n и m соответственно, а периоды T_a, T_s – взаимно простые числа, то выходная последовательность $\{x_t\}$ имеет период $T = (2^n - 1)2^{m-1}$.

Конгруэнтный генератор со случайными параметрами

Еще один способ комбинирования двух генераторов G_1, G_2 заключается в том, что G_2 изменяет параметры генератора G_1 с течением времени. Проиллюстрируем это в случае, когда G_2 — линейный конгруэнтный генератор:

$$x_t = (a_t x_{t-1} + b_t) \bmod N, \; t = 1, 2, \ldots \qquad \textbf{(10)}$$

где $x_0 \in A$ – некоторое стартовое значение, а $A_t = \begin{pmatrix} a_t \\ b_t \end{pmatrix} \in B, t = 1,2,\ldots$ есть некоторая псевдослучайная последовательность векторов, равномерно распределенных в B.
Доказано, что если $|B| = 3$, то наибольшее приближение распределения $\{x_t\}$ равномерному достигается, если множество параметров B имеет следующий вид: $B = \left\{ \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix} \right\}$.

Статистические тесты

Статистические тесты необходимы для того, чтобы определить меру близости сгенерированных псевдослучайных последовательностей к случайным. Случайность – это вероятностная характеристика, поэтому свойства случайных последовательностей можно сформулировать в терминах вероятностей. Возможные выходные значения статистических тестов, в случае их применения к истинно случайным последовательностям известны априори и описываются в терминах вероятности. Существует бесчисленное количество возможных статистических тестов, которые определяют, является последовательность случайной или нет. Однако, не существует конечного набора тестов, который бы считался полным. Более того, необходимо с осторожностью интерпретировать результаты тестирования, чтобы избежать неверных выводов о специфике исследуемого генератора.

Каждый статистический тест разработан для проверки конкретной нулевой гипотезы. В данной работе в качестве нулевой гипотезы выдвинем предположение о том, что тестируемая последовательность является случайной. Если же тестируемая последовательность не случайна, тогда будет справедлива альтернативная гипотеза. Каждый тест получает в качестве входного значения порожденную генератором последовательность и делает вывод о принятии или отклонении нулевой гипотезы, то есть определяет, случайные значения выдает генератор или нет.

Для каждого теста необходимо выбрать статистику, подходящую для принятия или отклонения нулевой гипотезы. В предположении о случайности последовательности такая статистика имеет распределение возможных значений. Теоретическое распределение этой статистики при условии справедливости нулевой гипотезы определяется математическими методами. После этого определяется пороговое значение. Во время тестирования последовательности вычисляется статистическое значение, которое затем сравнивается с пороговым. Если статистическое значение превышает пороговое значение, то нулевая гипотеза отвергается, и тогда принимается альтернативная. Если пороговое значение не превышено, то нулевая гипотеза принимается.

Допустим, последовательность в действительности является случайной, но в

результате тестирования нулевая гипотеза была отклонена. Такой результат носит название ошибки I рода. А если последовательность в действительности не является случайной, и при этом была принята нулевая гипотеза, то говорят об ошибке II рода.

Вероятность ошибки I рода часто называют уровнем значимости теста и обозначают символом α. Данную вероятность можно задать перед проведением теста. В данной работе α это вероятность того, что тест определит, что последовательность не случайна, а на самом деле она случайна. То есть последовательность демонстрирует неслучайное поведение, несмотря на то что была порождена «хорошим» генератором. Обычно в криптографии α присваивают значения близкие к 0,01.

Вероятность ошибки II рода обозначают символом β. Для каждого теста β это вероятность того, что тест определит неслучайную последовательность как случайную, то есть что «плохой» генератор породил последовательность, которая ведет себя как случайная. В отличии от α, β не является фиксированной величиной. β может принимать много различных значений, так как поток данных может оказаться неслучайным в бесконечном числе случаев, и в каждом случае β окажется разным. Вычисление ошибки II рода гораздо сложнее, чем вычисление ошибки I рода, по причине разнообразия неслучайных последовательностей.

Одной из основных задач приведенных ниже статистических тестов является минимизация вероятности принятия последовательности порожденной «плохим» генератором как порожденной «хорошим» генератором. Вероятности α и β связаны между собой, а также с длиной исследуемой последовательности n таким образом, что если две из них заданы, то можно определить и третью. Таким образом, выбираются значения α и n, а затем пороговое значение выбирается таким образом, чтобы ошибка β была минимальной.

Каждый тест основан на вычисленном статистическом значении, которое является функцией от последовательности. Пусть статистическое значение теста S, а пороговое значение t, тогда вероятность ошибки I рода будет равна

$$P\,(S > t \;||\; H_0 \text{ верна}) = P(H_0 \text{ отвергнута} \mid H_0 \text{ верна}),$$

а вероятность ошибки II рода будет равна

$$P(S \leq t \;||\; H_0 \text{ не верна}) = P\,(H_0 \text{ принята} \mid H_0 \text{ не верна}).$$

В результате статистического теста вычисляется значение *P-value*, которое представляет собой сумму аргументов против нулевой гипотезы. В данном случае *P-value* это вероятность того, что идеальный генератор породит последовательность менее случайную, чем тестируемая последовательность (вид случайности для каждого теста определяется индивидуально). Если в результате теста значение *P-value* равно 1, то

последовательность является абсолютно случайной. Значение *P-value*, равное 0, говорит о том, что последовательность полностью неслучайна. Для тестов может быть выбран уровень значимости α. Если $P\text{-}value \geq \alpha$, то нулевая гипотеза принимается, то есть последовательность является случайной. Если $P\text{-}value < \alpha$, то нулевая гипотеза отвергается, то есть последовательность не является случайной. Обычно α выбирается из отрезка [0,001, 0,01]. Если $P\text{-}value \geq 0{,}001$, то последовательность считается случайной с вероятностью 0,999. Если $P\text{-}value < 0{,}001$, последовательность не является случайной с вероятностью 0,999. Если $P\text{-}value \geq 0{,}01$, то вероятность того, что последовательность случайна, равна 0,99. Если $P\text{-}value < 0{,}01$, то вероятность того, что последовательность не случайна, равна 0,99.

Существует большое количество различных статистических тестов для псевдослучайных последовательностей. Наиболее известными являются NIST, DIEHARD и тесты Дональда Кнута. NIST STS (Statistical Test Suite) имеет большую, чем у остальных, криптографическую направленность, которая достигается путем введения в пакет таких тестов как проверка линейной сложности и универсального теста Маурера.

Частотный побитовый тест. Суть данного теста заключается в определении соотношения нулей и единиц во всей последовательности. Цель теста: выяснить, является ли число единиц и нулей в последовательности примерно одинаковым, что характерно для истинно случайной последовательности. Для этого определяется, насколько частота единиц близка к ½. Все последующие тесты проводятся при условии, что пройден данный тест.

Частотный блочный тест. Суть теста состоит в определении доли единиц внутри блоков длины M бит. Цель теста: узнать, является ли частота единиц внутри M-битного блока приблизительно равной $M/2$, как следовало бы ожидать в случае случайной последовательности. Если длина блока M равна единице, то тест переходит в частотный побитовый тест.

Тест серий. Данный тест заключается в том, чтобы определить суммарное число серий в последовательности, где серия – это непрерывная последовательность одинаковых бит. Серия длины k состоит ровно из k одинаковых бит и ограничена слева и справа битами противоположными по значению. Цель теста серий: выяснить, соответствует ли количество серий нулей и единиц исследуемой последовательности предполагаемому количеству серий истинно случайной последовательности для любой выбранной длины серии. В частности, тест проверяет, не являются ли переходы между нулями и единицами слишком быстрыми или, наоборот, слишком медленными.

Тест на самую длинную серию единиц в блоке. Суть данного теста состоит в нахождении самой длинной серии единиц внутри блока длины *M* бит. Цель теста: узнать, согласуются ли количество бит самой длинной серии единиц тестируемой последовательности и предполагаемое количество бит самой длинной серии единиц случайной последовательности. Стоит учитывать, что непостоянство предполагаемой длины максимальной серии единиц подразумевает аналогичное непостоянство и в случае серий нулей. Поэтому данный достаточно проводить только для единиц.

Тест рангов бинарных матриц. В качестве объекта для рассмотрения здесь выступают непересекающиеся подматрицы всей последовательности. Цель теста: проверить наличие в исходной последовательности линейной зависимости среди подстрок фиксированной длины.

Спектральный тест. В данном тесте оценивается высота пиков дискретного преобразования Фурье последовательности. Цель теста: обнаружить периодические свойства тестируемой последовательности (то есть повторяющиеся шаблоны, которые находятся рядом друг с другом), которые являются признаком отклонения характера последовательности от случайного. Необходимо, чтобы число пиков превышающих порог в 95 % значительно превышало 5 %.

Тест на совпадение неперекрывающихся шаблонов. В основе теста лежит факт появления заранее выбранных строк, называемых шаблонами. Цель теста: выявить генераторы, которые выдают непериодический шаблон слишком часто. Для этого теста и для теста на совпадение перекрывающихся шаблонов, для поиска m-битных шаблонов используется специальное m-битное окно. Если шаблон не найден, окно сдвигается на одну позицию. Если шаблон найден, окно переходит к биту, следующему за найденным шаблоном, и поиск продолжается.

Тест на совпадение перекрывающихся шаблонов. Данный тест, как и предыдущий, заключается в поиске заранее выбранных строк, называемых шаблонами. От предыдущего данный тест отличается тем, что если шаблон найден, окно сдвигается только на один бит, и поиск продолжается.

Универсальный статистический Маурера. Суть теста заключается в определении количества бит между пересекающимися шаблонами (мера, которая связана с длиной сжатой последовательности). Цель теста: узнать, может последовательность быть значительно сжать без потерь информации или нет. Последовательность, которую можно значительно сжать, не является случайной.

Тест на линейную сложность. В основе теста лежит регистр сдвига с линейной обратной связью (linear feedback shift register, РСЛОС, LFSR). Цель теста: определить,

является ли последовательность достаточно сложной, чтобы считаться случайной. Случайные последовательности характеризуются длинным регистром сдвига. Слишком короткий регистр сдвига свидетельствует о том, что последовательность неслучайна. Линейная сложность в данном тесте определяется с помощью алгоритма Берлекэмпа-Мэсси.

Тест на периодичность. Суть данного теста – частота всех возможных пересекающихся *m*-битных шаблонов во всей последовательности. Цель теста: проверить, будет ли число появлений 2^m *m*-битных пересекающихся шаблонов близким к числу появлений таких шаблонов в истинно случайной последовательности. Случайным последовательностям свойственна равномерность, поэтому все шаблоны длины *m* бит имеют одинаковую вероятность появления. В случае, если $m = 1$, тест переходит в частотный побитовый.

Тест приблизительной энтропии. Суть данного теста, так же как и предыдущего, заключается в определении частоты всех возможных пересекающихся *m*-битных шаблонов во всей последовательности. Цель теста: вычислить разность частот пересекающихся блоков длин *m* и *m+1* и сравнить результат с предполагаемым для случайной последовательности.

Тест кумулятивных сумм. В данном тесте исследуется кумулятивная сумма, больше всего отстающая от нуля (для этого в последовательности все «0» заменяются на « -1»). Цель теста: выяснить, является ли кумулятивная сумма частичных отрезков тестируемой последовательности слишком большой или, наоборот, слишком маленькой относительно ожидаемого поведения аналогичной кумулятивной суммы, вычисленной для случайной последовательности. Кумулятивная сумма может рассматриваться как случайное блуждание. Для случайной последовательности, результат случайного блуждания колеблется около нуля. Для некоторых типов неслучайных последовательностей отклонение случайных блужданий от нуля будет велико.

Тест на произвольные отклонения (вариант 1). В данном тесте интерес представляет количество циклов, имеющих ровно K посещений при случайном блуждании кумулятивной суммы. Кумулятивная сумма образуется из частичных сумм, после того как все «0» и «1» последовательности изменяются на «-1» и «+1» соответственно. Цикл случайного блуждания состоит из последовательных шагов единичной длины, которые носят случайных характер. Цикл заканчивается при возвращении в точку, в которой он начался. Цель теста: проверить, отклоняется ли число посещений для определенного состояния цикла от предполагаемого для истинно случайной последовательности. Этот

тест, вообще говоря, состоит из восьми тестов (по одному на каждое состояние: -4, -3, -2, -1 и +1, +2, +3, +4).

Тест на произвольные отклонения (вариант 2). Суть теста заключается в определении общего числа посещений определенного состояния при случайном блуждании кумулятивной суммы. Цель теста: выявить отклонения от предполагаемого числа посещений различных состояний. Данный тест, вообще говоря, состоит из восемнадцати тестов (по одному на каждое состояние: -9, -8, …, -1 и +1, +2, …, +9.

Таблица 1 – Контрольные последовательности: бинарное разложение числа *e* и числа π

Название теста	**Параметры**	$\boldsymbol{\pi}$	$\boldsymbol{e}$
Частотный побитовый	$n = 10^6$	0,953749	0,578211
Частотный блочный	$n = 10^6$ *block_length* = 128	0,211072	0,380615
Серий	$n = 10^6$	0,561917	0,419268
Самая длинная серия единиц в блоке	$n = 10^6$	0,718945	0,024390
Рангов бинарных матриц	$n = 10^6$	0,306156	0,083553
Спектральный	$n = 10^6$	0,847187	0,010186
Совпадение неперекрывающихся шаблонов	$n = 10^6$ *block_length* = 125 000 *temlplate_length* = 9 *template* = 000000001	0,078790	0,165757
Совпадение перекрывающихся шаблонов	$n = 10^6$ *temlplate_length* = 9 *template* = 111111111	0,110434	0,296897
Универсальный Маурера	$n = 10^6$	0,282568	0,669012
На линейную сложность	$n = 10^6$ *block_length* = 500	0,826335	0,255475
На периодичность	$n = 10^6$ *block_length* = 16	0,766182	0,143005
Приблизительной энтропии	$n = 10^6$ *block_length* = 10	0,700073	0,361595

Кумулятивных сумм	$n = 10^6$	*mode* = forward	0,628308	0,669886
		mode = reverse	0,663369	0,724265
Произвольные отклонения (вар. 1)	$n = 10^6$ $x = +1$		0,786868	0,844143
Произвольные отклонения (вар. 2)	$n = 10^6$ $x = -1$		0,826009	0,760966

Задание

1. Согласно варианту реализовать приложение, генерирующие псевдослучайную равномерно распределенную последовательность произвольной длины из заданного алфавита.

2. Реализовать программу, вычисляющую период последовательности. Подобрать параметры генератора таким образом, чтобы период последовательности имел максимальное значение.

3. Сгенерировать последовательность длиной 1 миллион бит. Реализовать три статистических теста (согласно варианту) и исследовать сгенерированную последовательность (тест пройден / не пройден). Предварительно проверить работоспособность тестов, взяв в качестве входной последовательности бинарное разложение числа π или числа *e* (таблица 1).

Варианты заданий

Легкий уровень

№	Генератор	Размер алфавита	Дополнение	Статистические тесты
1	Линейный конгруэнтный генератор	N=15	*Рассмотреть следующие случаи:* 1. Мультипликативный генератор Лемера: $c = 0$ 2. Смешанный генератор	1. Частотный побитовый 2. Частотный блочный 3. Самая длинная серия единиц
2	Аддитивный генератор	N=30	$x_{t+1} = (x_t + x_{t-1}) \bmod N$	1. Частотный побитовый 2. Серий 3. Самая длинная серия единиц
3	Генераторы Фибоначчи	N=2		1. Частотный блочный 2. Серий 3. Самая длинная серия единиц

Средний уровень

№	Генератор	Размер алфавита	Дополнение	Статистические тесты
4	Комбинирование алгоритмов методом Макларена-Марсальи	N=10 K=5		1. Серий 2. Рангов бинарных матриц 3. На линейную сложность
5	Генератор Эйхенауэра–Лена с обращением	N=20		1. Неперекрывающихся шаблонов 2. Кумулятивных сумм 3. На периодичность
6	LFSR		*Рассмотреть следующие случаи генераторов, основанных на примитивных многочленах:* 1. $f(x)=x^7+x+1$ 2. $f(x)=x^7+x^5+x^3+1$ 3. $f(x)=x^7+x^5+x^3+1$ *Какие из этих многочленов примитивны по модулю 2?*	1. Рангов бинарных матриц 2. Универсальный Маурера 3. Самая длинная серия единиц
7	LFSR		*Рассмотреть следующие случаи генераторов, основанных на примитивных многочленах:* 1. $f(x)=x^6+x^3+1$ 2. $f(x)=x^6+x^2+x+1$ 3. $f(x)=x^6+x+1$ *Какие из этих многочленов примитивны по модулю 2?*	1. Неперекрывающихся шаблонов 2. На линейную сложность 3. Самая длинная серия единиц
8	Квадратичный конгруэнтный генератор	N=10	*Рассмотреть следующие случаи:* 1. Генератор Ковью: $c=0$ 2. $c\neq 0$	1. Рангов бинарных матриц 2. Приблизительной энтропии 3. Кумулятивных сумм
9	LFSR		*Рассмотреть следующие случаи генераторов, основанных на примитивных многочленах:* 1. $f(x)=x^5+x^2+1$ 2. $f(x)=x^5+1$ 3. $f(x)=x^5+x^4+x^2+1$ *Какие из этих многочленов примитивны по модулю 2?*	1. Серий 2. Неперекрывающихся шаблонов 3. Универсальный Маурера
10	Конгруэнтный генератор, использующий умножение с переносом	N=30		1. Рангов бинарных матриц 2. Универсальный Маурера 3. Самая длинная

				серия единиц
11	LFSR		*Рассмотреть следующие случаи генераторов, основанных на примитивных многочленах:* 1. $f(x)=x^4+x$ 2. $f(x)=x^4+x^3+x^2+x+1$ 3. $f(x)=x^4+x+1$ *Какие из этих многочленов примитивны по модулю 2?*	1. Рангов бинарных матриц 2. Неперекрывающихся шаблонов 3. На периодичность
12	Генератор Эйхенауэра–Лена с обращением	N=15		1. Рангов бинарных матриц 2. Перекрывающихся шаблонов 3. Кумулятивных сумм

Сложный уровень

№	Генератор	Размер алфавита	Дополнение	Статистические тесты
13	Комбинирование с прореживающим генератором		*Рассмотреть следующие случаи генераторов, основанных на примитивных многочленах:* 1. $f(x)=x^7+x^3+x^2+x+1$ 2. $f(x)=x^7+x+1$ 3. $f(x)=x^7+x^3+1$ *Какие из этих многочленов примитивны по модулю 2?*	1. Спектральный 2. Перекрывающихся шаблонов 3. На линейную сложность
14	Комбинирование с селектирующим генератором		*Рассмотреть следующие случаи генераторов, основанных на примитивных многочленах:* 1. $f(x)=x^7+x+1$ 2. $f(x)=x^7+x^5+x^3+1$ 3. $f(x)=x^7+x^5+x^3+1$ *Какие из этих многочленов примитивны по модулю 2?*	1. Рангов бинарных матриц 2. Кумулятивных сумм 3. Произвольные отклонения (вар. 1)
15	Комбинирование с мажоритарным голосованием		*Рассмотреть следующие случаи генераторов, основанных на примитивных многочленах:* 1. $f(x)=x^6+x^3+1$ 2. $f(x)=x^6+x^2+x+1$ 3. $f(x)=x^6+x+1$ *Какие из этих многочленов примитивны по модулю 2?*	1. Спектральный 2. Неперекрывающихся шаблонов 3. Универсальный Маурера
16	Комбинирование с неравномерным движением регистров		*Рассмотреть следующие случаи генераторов, основанных на примитивных многочленах:*	1. Универсальный Маурера 2. На линейную сложность

			1. $f(x)=x^5+x^2+1$ 2. $f(x)=x^5+1$ 3. $f(x)=x^5+x^4+x^2+1$ *Какие из этих многочленов примитивны по модулю 2?*	3. Произвольные отклонения (вар. 1)
17	Комбинирование с неравномерным движением регистров и мажоритарным голосованием		*Рассмотреть следующие случаи генераторов, основанных на примитивных многочленах:* 1. $f(x)=x^4+x$ 2. $f(x)=x^4+x^3+x^2+x+1$ 3. $f(x)=x^4+x+1$ *Какие из этих многочленов примитивны по модулю 2?*	1. Спектральный 2. Приблизительной энтропии 3. Произвольные отклонения (вар. 2)
18	Комбинирование с прореживающим генератором		*Рассмотреть следующие случаи генераторов, основанных на примитивных многочленах:* 1. $f(x)=x^5+x^2+1$ 2. $f(x)=x^5+x^4+x^3+x^2+1$ 3. $f(x)=x^5+1$ *Какие из этих многочленов примитивны по модулю 2?*	1. Универсальный Маурера 2. Кумулятивных сумм 3. Произвольные отклонения (вар. 2)
19	Конгруэнтный генератор со случайными параметрами	N=15	1. $B=\left\{\binom{3}{0},\binom{3}{1},\binom{3}{2}\right\}$ 2. B выбрать произвольно, $\lvert B\rvert=5$	1. Спектральный 2. Перекрывающихся шаблонов 3. Произвольные отклонения (вар. 1)
20	Генератор Blum-Blum-Shub			1. Универсальный Маурера 2. Приблизительной энтропии 3. Произвольные отклонения (вар. 2)

Приложение 1

Генератор Blum-Blum-Shub

Данный алгоритм генерирует псевдослучайную последовательность бит $z_1, z_2, \ldots, z_l$ длины l:

1) Выбираются два достаточно больших секретных случайных (и различных) простых числа P и Q, каждое из которых сравнимо с 3 по модулю 4.

2) Вычисляется $N = P \cdot Q$.

3) Выбирается случайное целое число s (значение инициализации) из отрезка $[1, N-1]$, такое что $НОД(s, N) = 1$.

4) Вычисляется $x_0 \leftarrow s^2 \bmod N$.

5) Для i от 1 до l:

   i. $x_i \leftarrow x_{i-1}^2 \bmod N$;

   ii. $z_i \leftarrow$ *наименее значимый бит* x_i.

6) Формируется выходная последовательность $z_1, z_2, \ldots, z_l$.

Пример выполнения алгоритма:

1) Выберем два простых числа $P = 47$ и $Q = 67$. Оба сравнимы с 3 по модулю 4 в том смысле, что $47 \bmod 4 = 3$ и $67 \bmod 4 = 3$.

2) $N = 47 \cdot 67 = 3149$.

3) Выберем начальное значение $s = 7$, $НОД(7, 3149) = 1$.

4) $x_0 = 7^2 \bmod 3149 = 49$.

5) Запускаем цикл от 1 до 16 и оставшиеся числа вычисляем согласно алгоритму.

6) Получаем последовательность длиной 16 бит: 1110101011010010.