ЛЕКЦИЯ 2. ГЕНЕРАТОРЫ СЛУЧАЙНЫХ ЧИСЕЛ.

ПОСТАНОВКА ЗАДАЧИ ГЕНЕРАЦИИ СЛУЧАЙНЫХ ЧИСЕЛ

Во многих статистических методах и практических задачах программирования используется генерация случайных чисел, которые должны быть равномерно распределены в интервале (0; 1).

Если генератор выдает числа, смещенные в какую-то часть интервала (одни числа выпадают чаще других), то результат решения задачи, решаемой статистическим методом, может оказаться неверным. Поэтому задача создания хорошего генератора действительно случайных и действительно равномерно распределенных имеет исключительную важность.

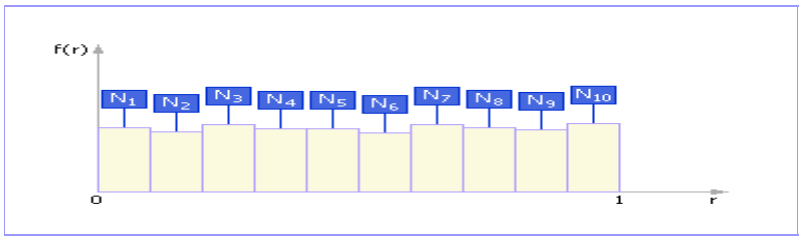
Математическое ожидание m_r и дисперсия D_r такой последовательности, состоящей из n случайных чисел r_i, должны быть следующими (если это действительно равномерно распределенные случайные числа в интервале от 0 до 1):

$$m_r = \frac{\sum_{i=1}^{n} r_i}{n} = 0.5$$

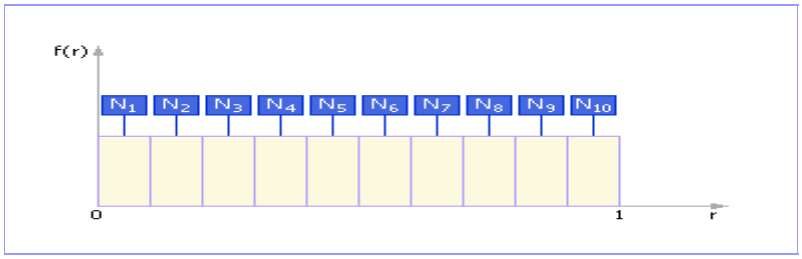
$$D_r = \frac{\sum_{i=1}^{n} (r_i - m_r)^2}{n} = \frac{1}{12}$$

Если в задаче потребуется, чтобы случайное число x находилось в интервале $(a; b)$, отличном от (0; 1), нужно воспользоваться формулой $x = a + (b - a) \cdot r$, где r — случайное число из интервала (0; 1).

За **эталон генератора случайных чисел** принят такой генератор, который порождает **последовательность** случайных чисел с *равномерным* законом распределения в интервале (0; 1). За одно обращение данный генератор возвращает одно случайное число. Если наблюдать такой ГСЧ достаточно длительное время, то окажется, что, например, в каждый из десяти интервалов (0; 0.1), (0.1; 0.2), (0.2; 0.3), …, (0.9; 1) попадет практически одинаковое количество случайных чисел — то есть они будут распределены равномерно по всему интервалу (0; 1). Если изобразить на графике $k = 10$ интервалов и частоты N_i попаданий в них, то получится примерно следующая экспериментальная кривая плотности распределения случайных чисел (см. рисунок).



Конечно же, в идеале кривая плотности распределения случайных чисел выглядела бы так, как показано на следующем рисунке, то есть в идеальном случае в каждый интервал попадает одинаковое число точек: $N_i = N/k$, где N — общее число точек, k — количество интервалов, $i = 1, \ldots, k$.



Следует помнить, что генерация произвольного случайного числа состоит из двух этапов:

- генерация нормализованного случайного числа (то есть равномерно распределенного от 0 до 1);
- преобразование нормализованных случайных чисел r_i в случайные числа x_i, которые распределены по необходимому пользователю (произвольному) закону распределения или в необходимом интервале.

Генераторы случайных чисел по способу получения чисел делятся на:

- физические;
- табличные;
- алгоритмические.

ФИЗИЧЕСКИЕ ГЕНЕРАТОРЫ СЛУЧАЙНЫХ ЧИСЕЛ

Примером физических генераторов случайных чисел могут служить: монета («орел» — 1, «решка» — 0); игральные кости; поделенный на секторы с цифрами барабан со стрелкой; аппаратурный генератор шума, в качестве которого используют шумящее тепловое устройство, например, транзистор.

Вот, как, например, можно получить случайное трехразрядное число, распределенное по равномерному закону в интервале от 0 до 1, с помощью монеты. Точность — три знака после запятой.

Подбросим монету 9 раз. Если монета упала решкой, то запишем «0», если орлом, то «1». Итак, допустим, что в результате эксперимента получили случайную последовательность 100110100.

Разобьем полученную двоичную последовательность 100110100 на триады: 100, 110, 100. После перевода этих двоичных чисел в десятичные получаем: 4, 6, 4. Подставив спереди «0.», получим: 0.464. Таким методом могут получаться только числа от 0.000 до 0.777 (так как максимум, что можно «выжать» из трех двоичных разрядов — это $111_2 = 7_8$) — то есть, по сути, эти числа представлены в восьмеричной системе счисления. Для перевода *восьмеричного* числа в *десятичное* представление выполним: $0.464_8 = 4 \cdot 8^{-1} + 6 \cdot 8^{-2} + 4 \cdot 8^{-3} = 0.6015625_{10} = 0.602_{10}$.

Итак, искомое число равно: 0.602.

ТАБЛИЧНЫЕ ГЕНЕРАТОРЫ СЛУЧАЙНЫХ ЧИСЕЛ

Табличные генераторы случайных чисел в качестве источника случайных чисел используют специальным образом составленные таблицы, содержащие проверенные некоррелированные, то есть никак не зависящие друг от друга, цифры. Ниже приведен небольшой фрагмент такой таблицы. Обходя таблицу слева направо и сверху вниз, можно получать равномерно распределенные от 0 до 1 случайные числа с нужным числом знаков после запятой (в нашем примере мы используем для каждого числа по три знака). Так как цифры в таблице не зависят друг от друга, то таблицу можно обходить разными способами, например, сверху вниз, или справа налево, или, скажем, можно выбирать цифры, находящиеся на четных позициях.

Случайные цифры								**Равномерно распределенные от 0 до 1 случайные числа**
9	2	9	2	0	4	2	6	0.929
9	5	7	3	4	9	0	3	0.204
5	9	1	6	6	5	7	6	0.269
…								…

Достоинство данного метода в том, что он дает действительно случайные числа, так как таблица содержит проверенные некоррелированные цифры. Недостатки метода: для хранения большого количества цифр требуется много памяти; большие трудности порождения и проверки такого рода таблиц, повторы при использовании таблицы уже не гарантируют случайности числовой последовательности, а значит, и надежности результата.

АЛГОРИТМИЧЕСКИЕ ГЕНЕРАТОРЫ СЛУЧАЙНЫХ ЧИСЕЛ

Числа, генерируемые с помощью этих генераторов случайных чисел, всегда являются псевдослучайными (или квазислучайными), то есть каждое последующее сгенерированное число зависит от предыдущего:

$$r_{i+1} = f(r_i).$$

Последовательности, составленные из таких чисел, образуют петли, то есть обязательно существует цикл, повторяющийся бесконечное число раз. Повторяющиеся циклы называются **периодами.**

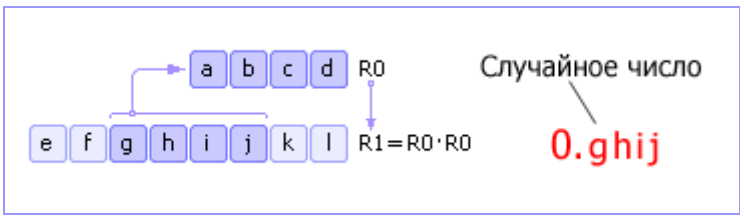
Достоинством данных генераторов является быстродействие; генераторы практически не требуют ресурсов памяти, компактны. Недостатки: числа нельзя в полной мере назвать случайными, поскольку между ними имеется зависимость, а также наличие периодов в последовательности квазислучайных чисел.

Рассмотрим несколько алгоритмических методов получения:

- метод серединных квадратов;
- метод серединных произведений;
- метод перемешивания;
- линейный конгруэнтный метод.

Метод серединных квадратов

Имеется некоторое четырехзначное число $R0$. Это число возводится в квадрат и заносится в $R1$. Далее из $R1$ берется середина (четыре средних цифры) — новое случайное число — и записывается в $R0$. Затем процедура повторяется (см. рисунок ниже). Отметим, что на самом деле в качестве случайного числа необходимо брать не ghij, а 0.ghij — с приписанным слева нулем и десятичной точкой. Этот факт как раз и отражен на рисунке.



Недостатки метода:

1) если на некоторой итерации число $R0$ станет равным нулю, то генератор вырождается, поэтому важен правильный выбор начального значения $R0$;

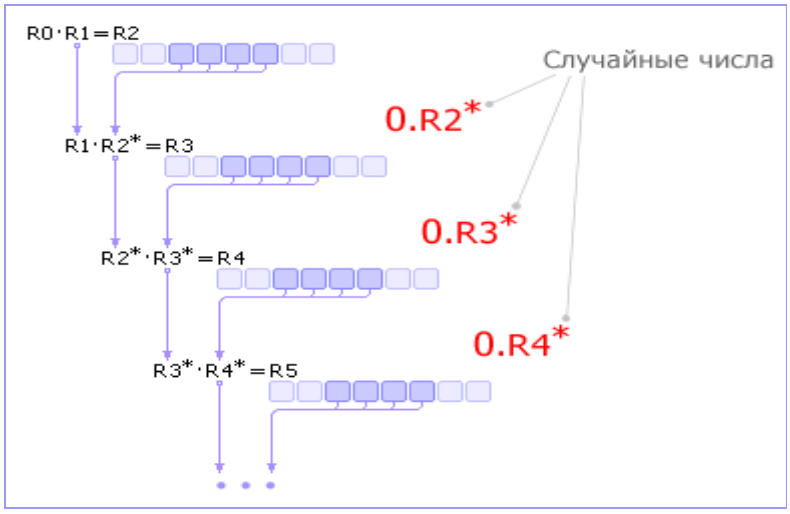
2) генератор будет повторять последовательность через M^n шагов (в лучшем случае), где n — разрядность числа $R0$, M — основание системы счисления.

Для примера из верхнего рисунка: если число $R0$ будет представлено в двоичной системе счисления, то последовательность псевдослучайных чисел повторится через $2^4 = 16$ шагов. Заметим, что повторение последовательности может произойти и раньше, если начальное число будет выбрано неудачно.

Описанный способ был предложен Джоном фон Нейманом и относится к 1946 году. Поскольку этот способ оказался ненадежным, от него очень быстро отказались.

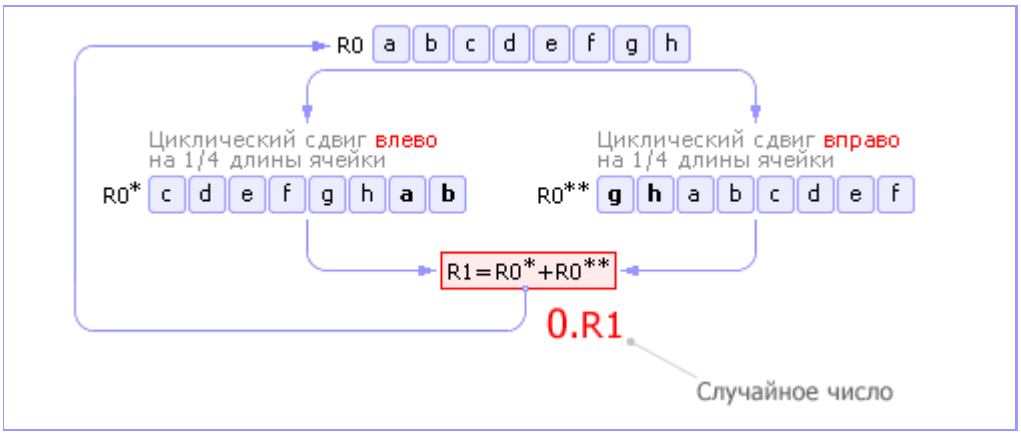
Метод серединных произведений

Число $R0$ умножается на $R1$, из полученного результата $R2$ извлекается середина $R2^*$ (это очередное случайное число) и умножается на $R1$. По этой схеме вычисляются все последующие случайные числа (см. рисунок).



Метод перемешивания

В методе перемешивания используются операции циклического сдвига содержимого ячейки влево и вправо. Идея метода состоит в следующем. Пусть в ячейке хранится начальное число $R0$. Циклически сдвигая содержимое ячейки влево на 1/4 длины ячейки, получаем новое число $R0^*$. Точно так же, циклически сдвигая содержимое ячейки $R0$ вправо на 1/4 длины ячейки, получаем второе число $R0^{**}$. Сумма чисел $R0^*$ и $R0^{**}$ дает новое случайное число $R1$. Далее $R1$ заносится в $R0$, и вся последовательность операций повторяется (см. рисунок).



Обратите внимание, что число, полученное в результате суммирования $R0^{*}$ и $R0^{**}$, может не уместиться полностью в ячейке $R1$. В этом случае от полученного числа должны быть отброшены лишние разряды.

Линейный конгруэнтный метод

Линейный конгруэнтный метод является одной из простейших и наиболее употребительных в настоящее время процедур, имитирующих случайные числа. В этом методе используется операция $\mathrm{mod}(x, y)$, возвращающая остаток от деления первого аргумента на второй. Каждое последующее случайное число рассчитывается на основе предыдущего случайного числа по следующей формуле:

$$r_{i+1} = \mathrm{mod}(k \cdot r_i + b, M).$$

Где M – большое число; k – множитель, такой, что 0<k<M; b – приращение, такое что 0<b<M. Для качественного генератора требуется подобрать подходящие коэффициенты. Необходимо, чтобы число M было довольно большим, так как период не может иметь больше M элементов.

ЗАДАНИЕ К ЛЕКЦИИ 6

Описать функцию MyRandom(int n), возвращающую случайное вещественное число в интервале от 0 до n. Использовать табличный генератор случайных чисел из приведенной в лекции таблицы. Продемонстрировать работу функции.