Opening the Black Box of Random Numbers

Yutaka Nishiyama

Abstract

Random numbers are frequently used all around us, but many people do not know the basis of random numbers. After touching upon prime numbers and prime number theory, this article explains the principles of random number generation using primes and primitive roots, and explains how they are actually handled inside computers.

Keywords: Random number, Prime number, Sieve of Eratosthenes, Prime number theory, Complex prime number, Primitive root, Congruent linear generator, Mersenne number

1. Random numbers all around us

When you leave yourself logged on to a computer for a while, a screen saver is activated. This is to prevent an image from being burned onto the screen. There are a great variety of screen savers, with various different colors and patterns, but one common feature is that the patterns and coordinates are determined using random numbers. The coordinate values are likely generated using a built-in mathematical RAND function. Although many people may know how to use this function, only a few people know how it works. In mathematics, random numbers are related to the prime numbers and primitive roots discussed in elementary number theory, and this time I'd like to think about what's inside the random number black box.

Even without touching a computer, we make great use of random numbers and the concept of random numbers in daily life. Just a few examples include the toss of a coin in soccer matches to determine which team can choose to attack or defend first, the use of dice in ***sugoroku*** (a Japanese dice game) and ***mah jong***, the use of balls with numbers written on them in bingo, and rotating boards and darts used in some lottery games. The thing that can said to be common among all of these is that the outcome is not known in advance. Random numbers are not influenced by previous results, so they can be said to be independent. And since they do not have regularity, they can be said to be irregular.

Dice are often used for random numbers. Dice are regular hexahedrons (cubes), and are capable of generating random numbers between 1 and 6. Dice have dots engraved in them: 6 is opposite 1, 5 is opposite 2, and 4 is opposite 3, so the total is always 7. In order to maintain the center of mass in the exact center of a die, the dot representing 1 is slightly larger than the others. This must be intended to ensure that the numbers from 1 to 6 appear equally on average. Two dice are used for random numbers larger than 6, and other regular polyhedra (such as dodecahedra with 12 faces and icosahedra with 20) are also be used. Books full of random numbers have also been sold in order to avoid having to throw dice. There was indeed a generation when tables of random numbers were good business.

Let's think about the random numbers between 1 and 6 that dice generate. We could consider various random numbers, but we'll restrict ourselves to cycles of 6 here. Let's line up all the numbers from 1 to 6 randomly.

1, 2, 3, 4, 5, 6
6, 5, 4, 3, 2, 1
1, 3, 5, 2, 4, 6

Looking at these, they don't seem like random numbers. Random numbers must be independent; they may not be related to their successors or predecessors. However, when the numbers are lined up like so,

2, 6, 4, 5, 1, 3

they appear to be random. So why were numbers like these generated? We'll try to answer that here.

2. Primes and prime number theory

Prime numbers are the natural numbers not divisible by any number besides 1 and themselves. For example, 2, 3, 5, …… and so on, are primes.

Problem 1. Find all the primes between 1 and 100, and count how many there are.

One way to check whether or not the natural number n is prime is to try to divide n by every number from 2 to $n-1$. If none of the numbers cleanly divides it, then it must be prime. Thinking about it carefully, however, it is not necessary to try every number up to $n-1$, and in fact the numbers up to $\sqrt{n}$ are sufficient. Also, by skipping all even numbers, which are divisible by 2, the result can be obtained more efficiently. The problem of finding all the primes up to n is often used as a Visual Basic practical exercise.

A method for finding primes has been known since the age of the ancient Greeks, by using the 'sieve of Eratosthenes' (Figure 1). For example, the way to find the prime numbers up to 30 is as follows. Write out the natural numbers from 2 to 30. Since 2 is prime, fill in the multiples of 2 in a diagonal line, 4, 6, 8, …… Next, since 3 is prime, fill in the multiples of 3 in a diagonal line, 6, 9, 12, …… By working in this way, the numbers that remain will be prime. While this method is primitive, it does have the advantage that it prevents oversights. It also wields power when it comes to large computations such as finding all the primes up to 10^7, or counting their number.

$$\begin{array}{cccccccccc}
2 & 3 & \cancel{4} & 5 & \xcancel{6} & 7 & \cancel{8} & \bcancel{9} & \cancel{10} & 11 & \xcancel{12} & 13 \\
\cancel{14} & \bcancel{15} & \cancel{16} & 17 & \xcancel{18} & 19 & \cancel{20} & \bcancel{21} & \cancel{22} & 23 \\
\xcancel{24} & 25 & \cancel{26} & \bcancel{27} & \cancel{28} & 29 & \xcancel{30}
\end{array}$$

Figure 1. The sieve of Eratosthenes

Prime numbers may be applied in various ways. When I used to work for IBM, I knew that prime factorizations have an application in the fast Fourier transform (FFT). When seismic waves are analyzed in terms of frequency, the waves may be converted into sine and cosine

Fourier series. If there are 1000 points of sample data, it is necessary to perform 1000 calculations. However, the value of 1024, which is in this neighborhood, has the prime factorization of 2^{10}, and while 1024 calculations would normally be necessary, the fast Fourier transform, which only requires 10 calculations, may be used. I don't know the details of the algorithm, but I can say that it makes practical use of prime numbers. The fast Fourier transform was discovered by J. W. Cooley and John Tukey in 1965.

Many aspects of prime numbers have been researched over a long period. One of these is the prime number theorem, which states what proportion of the natural numbers are prime, and was predicted by Gauss in 1792. It was later proven independently by both Hadamard and Poussin in 1896. It is expressed explicitly by the following formula. Regarding the number of primes between 1 and x, denoted $\pi(x)$,

$$\pi(x) \approx \frac{x}{\log x} \quad (\log x \text{ indicates the natural logarithm}).$$

Calculating the actual value for $x < 1000$ reveals that this formula is relatively accurate, but for $x > 10^5$ however, it is insufficient. A more accurate formula can be obtained using the logarithmic integral,

$$Li(x) = \frac{x}{\log x} + \frac{1!x}{\log^2 x} + \cdots\cdots + \frac{(k-1)!x}{\log^k x} + O\left(\frac{x}{\log^{k+1} x}\right).$$

The actual number of primes up to the natural number x, the approximate number according to the prime distribution function, and the approximate number according to the improved formula are shown in Table 1. $Li(x)$ is taken as the sum up to the 9th term for $x = 10^5$, up to the 10th term for $x = 10^6$, and up to the 12th term for $x = 10^7$. It is thought that this formula is usually accurate for large values of x.

x	Actual number	$\frac{x}{\log x}$	$Li(x)$
10	4	4	
100	25	22	
1000	168	145	
10^4	1229	1086	
10^5	9592	8686	9628
10^6	78498	72382	78627
10^7	664579	620421	664915

Table 1. Number of primes, actual and approximated

3. Complex factors and prime numbers

In our discussion of prime numbers, let us touch upon complex factors. Numbers that are only divisible by 1 and themselves are called prime numbers, and we know that 2, 3, 5 $\cdots\cdots$ are prime. How about expanding numbers in the complex realm? From 2, 3, 5, $\cdots\cdots$ the numbers 2 and 5 can be factorized as

$2=(1+i)(1-i)$,
$5=(2+i)(2-i)$,

so 2 and 5 are no longer prime. It feels a bit strange but this is the truth according to complex numbers. There is such a thing as a prime factorization under complex numbers. $32+22i$ is factorized as follows.

$$32+22i$$
$$=2(16+11i)=(1+i)(1-i)(2+3i)(5-2i)$$

In this case the complex numbers $1+i$, $1-i$, $2+3i$, and $5-2i$ are the prime factors.

Among the real numbers, the fundamental number is 1. In the case of complex numbers, there are four fundamental numbers, 1, -1, i, and $-i$. Numbers which can only be divided by these four factors and themselves are considered prime. By making a computer program using the method of the sieve of Eratosthenes introduced above, it is possible to construct a two dimensional diagram of the complex numbers that are prime (Figure 2). If you actually make the diagram, you will see that it has a pattern like a tablecloth, and has a rather nice mathematical appearance. Consult reference (see Izumori, 1991) for information regarding the construction of a suitable BASIC program.

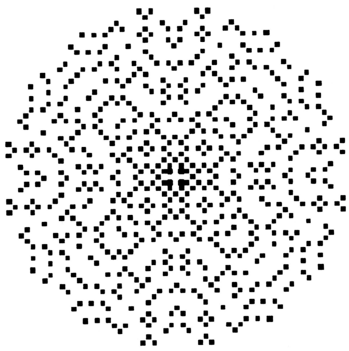
Figure 2. Complex prime numbers

4. Primes and primitive roots

Leaving the discussion of primes at this point, I'd now like to explain about the pseudo-random numbers that were mentioned at the outset. The numbers 1 to 6 are marked on the faces of dice, and if a sequence like

2, 6, 4, 5, 1, 3

is generated by some method, they are well suited for use as random numbers. So what is involved in such a method? For primes there are such things as 'primitive roots'. For example, 3 is a primitive root of the prime number 7. Let's use this relationship to try and generate some pseudo-random numbers.

$3^1=3$
$3^2=9\equiv 2 \bmod 7$
$3^3=2\times 3=6$
$3^4=6\times 3=18\equiv 4 \bmod 7$
$3^5=4\times 3=12\equiv 5 \bmod 7$
$3^6=5\times 3=15\equiv 1 \bmod 7$

The following sequence of random numbers is thus generated in this way.

3, 2, 6, 4, 5, 1

The manner of calculation is repeated multiplication by 3. When the result is greater than 7, a multiple of 7 is subtracted and the remainder is taken as the answer. After repeating this calculation 6 times, the result returns to 1.

This is called a linear congruential generator. In this case, the last number is 1, so to avoid this, if the initial value is taken as $3^2 \equiv 2 \bmod 7$ then the sequence becomes

2, 6, 4, 5, 1, 3.

The random numbers from 1 to 6 can be generated from the prime number 7. Putting it another way, the numbers 3^1, 3^2, 3^3, 3^4, 3^5, 3^6 with 7 as a modulus can be said to constitute a cyclic group.

For a prime number p and coprime number r, the relationship $r^{p-1} \equiv 1 \bmod p$ is satisfied. This is known as Fermat's little theorem. The reason that the theorem is described as 'little' is to distinguish it from the famous Fermat's last theorem. This congruence expression may have solutions of the form $r^i \equiv 1 \bmod p$ for $i < p-1$, but r is only a primitive root of p when the expression is satisfied for $i = p-1$. For the prime number 7, besides 3, 5 is also a primitive root. The primitive roots of the primes 3, 5, 7, 11 and 13 are shown in Table 2.

Problem 2. Using the fact that 6 is a primitive root of the prime number 11, generate a pseudo-random sequence from 1 to 10.

p (prime)	r (primitive root)
3	2
5	2, 3
7	3, 5
11	2, 6, 7, 8
13	2, 6, 7, 11

Table 2. Primes and primitive roots

5. Random numbers on computers

The explanations above deal with the principles of random number generation, but how are random numbers actually handled inside computers? Computers have built-in functions such as RAND for generating random numbers in the interval (0, 1). Integer random numbers are generated first, and then divided by the largest possible number to calculate a random real number in the interval (0, 1). I don't know the details of the algorithm used in the Visual BASIC built in RAND function, but one well-known pseudo-random number generation algorithm is as follows. All such random number algorithms involve the concept represented by the following congruence equation.

$$X_i = (aX_{i-1} + c) \bmod M \quad (a \text{ and } c \text{ are constants})$$

After setting X_{i-1} to some value, it is multiplied by a. If this value exceeds M, it is divided by M and the remainder is given as the value of X_i. Random numbers are generated one after the other using this iterative formula.

The variables handled by programs include real number types and integer types. Integer type random numbers use 4 bytes (*i.e.*, 32 bits). The number of different values that can be represented with 32 bits is a power of 2, but one bit at the head is the 'sign bit'. A 0 signifies a positive number, while a 1 signifies a negative number. The actual values are expressed using the 31 bits that remain after excluding this one sign bit. Assuming that the random numbers are positive, the largest number that can be represented by 31 bits is $2^{31}-1=2147483647$. By generating random numbers in the interval 1 to $2^{31}-1$ using the congruence expression explained above and dividing them by this largest number, random numbers can be generated in the interval $(0, 1)$.

The problem is then how to select the prime p and the primitive root r. The value of the largest integer-type variable is $2^{31}-1$. The primes that can be expressed in the form 2^p-1 are known as Mersenne numbers. There are 23 Mersenne numbers existing in the range $p<11400$, including

$$p=2, 3, 5, 7, 13, 17, 19, 31, 61, \cdots\cdots$$

These values of p do not include all the primes.

Fortunately, the largest number that can be expressed with 31 bits, $2^{31}-1$, is a Mersenne number and is prime. It is therefore useful to obtain the primitive roots of $2^{31}-1$. As shown in Table 2, the number of primitive roots is not just 1. For small prime numbers, the primitive roots may be obtained easily, but for a number like $2^{31}-1$ which has 10 digits in base 10, it's not so easy to obtain them and we must rely on the help of a computer.

S. K. Park and K. W. Miller used $M=2^{31}-1$ as a prime, and $a=7^5$ as a primitive root in the formula

$$X_i=7^5X_{i-1} \bmod (2^{31}-1).$$

As mentioned in reference (see Park and Miller, 1988), finding suitable pairs of primes and primitive roots is difficult. $2^{31}-1=2147483647$ is prime, but the primitive root $7^5=16807$ is not prime. The numbers $M=2^{31}-1$ and $a=7^5$ used in the congruence formula are coprime, so the above stated Fermat's little theorem is satisfied. That is,

$$(7^5)^{2^{31}-2}\equiv 1 \bmod (2^{31}-1).$$

This is satisfied without the need for calculation, but actually verifying it with a computer confirms that after $2^{31}-2$ iterations, the value returns to 1 for the first time. Because the result does not return to 1 at any point before $2^{31}-2$ iterations, it passes through every possible value and is thus an excellent primitive root.

The method above has a cycle length of $2^{31}-2$ which prompts no complaints, but its weak point is that the division calculation required by the $\bmod (2^{31}-1)$ operation takes time. This issue is addressed by IBM's well-known improved RANDU subroutine (1970). The RANDU congruence expression is given by the following formula.

$$X_i=65539X_{i-1} \bmod 2^{31}$$

$M=2^{31}(=2147483648)$ is not prime, so Fermat's little theorem does not apply, but $a=2^{16}+3(=65539)$ is prime, and M is coprime with a, so the generation of random numbers with a large cycle length can be expected. Confirming this with a computer revealed that

$$(2^{16}+3)^{2^{29}}\equiv 1 \bmod 2^{31}.$$

The cycle length is a quarter of the maximum cycle length, which is a relatively long cycle.

From a mathematical perspective it is beneficial to choose a prime for M. In the case of RANDU however, the divisor 2^{31} is not prime. The reason it was chosen is surely because it yields an efficient division calculation. Inside a computer, a division by 2^{31} is performed using a machine instruction that slides a binary number 31 bits to the right. This uses a shift register.

Leaving the details for another time, suffice it to say that this method has the advantage of being an especially fast computation. But RANDU belongs to the year 1970. As computer technology develops, the speed of computation increases and this kind of ingenuity is no longer necessary. Due to the lack of precision in the random numbers generated by RANDU, other methods are now in use.

Random numbers are generated using primes and primitive roots. Perhaps by knowing this mechanism, one may understand and appreciate the wonder of mathematics.

References

Izumori, H. (1991). *PC de Tanoshimu Koko Sugaku* [Enjoying High School Maths with a Computer], Tokyo: Saiensusya, 47-51.

Park, S.K., Miller, K.W. (1988). Random number generators: Good ones are hard to find, Communication of the ACM, 31, 1192-1201.