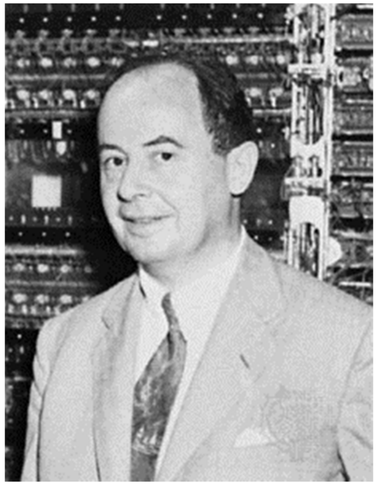
Pseudorandom numbers

John von Neumann:



Any one who considers
arithmetical methods of producing random digits
is, of course, in a state of sin.
For, as has been pointed out several times,
there is no such thing as a random number
— there are only methods to produce random numbers, and
a strict arithmetic procedure of course is not such a method.

"Various Techniques Used in Connection with Random Digits,", in *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.

Pseudorandom number generator

Random vs. pseudorandom behaviour

Random behavior -- Typically, its outcome is unpredictable and the parameters of the generating process cannot be determined by any known method.

Examples:

Parity of number of passengers in a coach in rush hour.

Weight of a book on a shelf in grams modulo 10.

Direction of movement of a particular N_2 molecule in the air in a quiet room.

Pseudo-random -- Deterministic formula,

-- Local unpredictability, *"output looks like random*",

-- Statistical tests might reveal more or less "random behaviour"

Pseudorandom integer generator

A pseudo-random integer generator is an algorithm which produces a sequence

$$\{x_n\} = x_0, x_1, x_2, \ldots$$

of non-negative integers, which manifest pseudo-random behaviour.

Pseudorandom number generator

Pseudorandom integer generator

Two important statistical properties:

- Uniformity
- Independence

Random number in a interval $[a, b]$ must be independently drawn from a uniform distribution with probability density function:

$$f(x) = \begin{cases} \frac{1}{b-a} & x \in [a, b] \\ 0 & \text{elsewhere} \end{cases}$$

Good generator

- Uniform distribution over large range of values:

  Interval $[a, b]$ is long, period = $b - a,$ generates all integers in $[a, b]$.
- Speed

  Simple generation formula.

  Modulus (if possible) equal to a power of two – fast bit operations.

Pseudorandom number generator

Random floating point number generator

Task 1: Generate (pseudo) random integer values from an interval $[a, b]$.
Task 2: Generate (pseudo) random floating point values from interval $[0,1[$.

Use the solution of Task 1 to produce the solution of Task 2.
Let $\{x_n\}$ be the sequence of values generated in Task 1.
Consider a sequence $\{y_n\} = \{(x_n - a) / (b - a - 1)\}$.

Each value of $\{y_n\}$ belongs to $[0,1[$.
"Random" real numbers are thus approximated by "random" fractions.
Large length of $[a, b]$ guarantees sufficiently dense division of $[0,1[$.

Example 1

$[a, b] = [0, 1024]$.
$\{x_n\} = \{712, \quad 84, \quad 233, \quad 269, \quad 810, \quad 944, \dots\}$

$$\{y_n\} = \{712/1023,\ 84/1023,\ 233/1023,\ 269/1023,\ 810/1023,\ 944/1023,\ \dots\}$$
$$= \{0.696,\ 0.082,\ 0.228,\ 0.263, 0.792,\ 0{,}923,\ \dots\}$$

Linear Congruential generator

Linear Congruential generator produces sequence $\{x_n\}$ defined by relations

$$0 \le x_0 < M,$$
$$x_{n+1} = (Ax_n + C) \mod M, \quad n \ge 0.$$

Modulus M, seed x_0, multiplier and increment A, C .

Example 2

$$M = 18,\ A = 7,\ C = 5.$$

$$x_0 = 4,$$
$$x_{n+1} = (7x_n + 5) \mod 18, \quad n \ge 0.$$

$$\{x_n\} = \underbrace{4, 15, 2, 1, 12, 17, 16, 9, 14, 13, 6, 11, 10, 3, 8, 7, 0, 5,}_{\text{sequence period, length = 18}} 4, 15, 2, 1, 12, 17, 16, \ldots$$

Linear Congruential Generator

Example 3

$$M = 15, A = 11, \; C = 6.$$

$$x_0 = 8,$$
$$x_{n+1} = (11x_n + 6) \mod 15, \qquad n \geq 0.$$

$$\{x_n\} = \underbrace{8, 14, 5, 11, 2,}_{\text{sequence period, length} = 5} 8, 14, 5, 11, 2, 8, 14, \; \ldots$$

Example 4

$$M = 13, A = 5, \; C = 11.$$

$$x_0 = 7,$$
$$x_{n+1} = (5x_n + 11) \mod 13, \qquad n \geq 0.$$

$$\{x_n\} = \underbrace{7,}_{\text{sequence period, length} = 1} 7, 7, 7, 7, \; \ldots$$

Linear Congruential Generator

Misconception

Prime numbers are "more random" than composite numbers, therefore using prime numbers in a generator improves randomness.
Counterexample: Example 4, all parameters are primes:

$$x_0 = 7, \quad x_{n+1} = (5x_n + 11) \mod 13.$$

Maximum period length

Hull-Dobell Theorem:
The lenght of period is maximum, i.e. equal to M, iff conditions 1. - 3. hold:
1. C and M are coprimes.
2. $A–1$ is divisible by each prime factor of M.
3. If 4 divides M then also 4 divides $A–1$.

Example 5

1. $M = 18, \ A = 7, \ C = 6.$ Condition 1. violated
2. $M = 20, \ A = 17, \ C = 7.$ Condition 2. violated
3. $M = 17, \ A = 7, \ C = 6.$ Condition 2. violated
4. $M = 20, \ A = 11, \ C = 7.$ Condition 3. violated
5. $M = 18, \ A = 7, \ C = 5.$ All four conditions hold

Linear Congruential Generator

Randomnes issues

Example 6

$$x_0 = 4,$$
$$x_{n+1} = (7x_n + 5) \mod 18, \quad n \ge 0.$$

$$\{x_n\} = \underbrace{4, 15, 2, 1, 12, 17, 16, 9, 14, 13, 6, 11, 10, 3, 8, 7, 0, 5,}_{\text{sequence period, length = 18}} 4, 15, 2, 1, 12, 17, 16, \ldots$$

$$\{x_n \bmod 2\} = 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, \ldots$$

$$\{x_n \bmod 3\} = 1, 0, 2, 1, 0, 2, 1, 0, 2, 1, 0, 2, 1, 0, 2, 1, 0 ,2, 1, 0, 2, 1, 0, 2, 1, \ldots$$

$$\{x_n \operatorname{div} 4\} = \underbrace{0, 3, 0, 0, 3, 4, 4, 2, 3, 3, 1, 2, 2, 0, 2, 1, 0 ,1,} 0, 3, 0, 0, 3, 4, 4, \ldots$$

Trouble

Low order bits of values generated by LCG exhibit significant lack of randomness.

Remedy

Disregard the lower bits in the output (not in the generation process!).
Output the sequence $\{y_n\} = \{x_n \operatorname{div} 2^H\}$, where $\mathrm{H} \ge ¼ \log_2(\mathrm{M})$.

Sequence period

Many generators produce a sequence $\{x_n\}$ defined by the general recurrence rule

$$x_{n+1} = f(x_n) \qquad n \geq 0.$$

Therefore, if $x_n = x_{n+k}$ for some $k > 0$, then also

$$x_{n+1} = x_{n+k+1},\ x_{n+2} = x_{n+k+2},\ x_{n+3} = x_{n+k+3}, \ldots$$

Sequence period

Subsequence of minimum possible length $p > 0$, $\{x_n, x_{n+1}, x_{n+2}, \ldots x_{n+p-1}\}$ such that for any $n \geq 0$: $\quad x_n = x_{n+p}$.

Random repetitions

Values $x_n, x_{n+1}, x_{n+2}, \ldots, x_{n+p-1}$ are unique in some (simple) generators.

To increase the random-like behavior of the sequence additional operations may be applied.
Typically, it is computing $x_n \bmod W$ for some $W < \max\limits_{n \geq 0} \{x_n\}$,
often W is a power of 2 and mod is just bitwise right shift.

Combined Linear Congruential Generator

Definition Let there be r linear congruential generators defined by relations

$$0 \le y_{k,0} < M_k$$
$$y_{k,n+1} = \left(A_k y_{k,n} + C_k\right) \bmod M_k, \quad n > 0.,$$
$$1 \le k \le r.$$

The combined linear congruential generator is a sequence $\{x_n\}$ defined by

$$x_n = (y_{1,n} - y_{2,n} + y_{3,n} - y_{4,n} + \ldots (-1)^{r-1} \cdot y_{r,n}) \bmod (M_1 - 1), \quad n \ge 0.$$

Fact Maximum possible period length (not always attained!) is
$(M_1 - 1)(M_2 - 1) \ldots (M_r - 1) \ / \ 2^{r-1}$.

Example 7 $\mathrm{r} = 2, \quad 1 \le y_{1,0} \le 2147483562, \quad 1 \le y_{2,0} \le 2147483398$

$$y_{1,n+1} = \left(40014 y_{1,n} + 0\right) \bmod 2147483563, \quad n \ge 0,$$
$$y_{2,n+1} = \left(40692 y_{2,n} + 0\right) \bmod 2147483399, \quad n \ge 0,$$
$$x_n = (y_{1,n} - y_{2,n}) \bmod 2147483562, \quad n \ge 0.$$

Period length is $\frac{(M_1 - 1)(M_2 - 1)}{2} = 2305842648436451838$.

Combined Linear Congruential Generator

Example 8 $\quad r = 3, \quad y_{1,0} = y_{2,0} = y_{3,0} = 1,$

$$y_{1,n+1} = \left(9y_{1,n} + 11\right) \bmod 16, \quad n \geq 0,$$
$$y_{2,n+1} = \left(7y_{2,n} + 5\right) \bmod 18, \quad n \geq 0,$$
$$y_{3,n+1} = \left(4y_{3,n} + 8\right) \bmod 27, \quad n \geq 0,$$
$$x_n = (y_{1,n} - y_{2,n} + y_{3,n}) \bmod 15, \quad n \geq 0.$$

$\{x_n\}$ = 1, 4, 0, 2, 7, 12, 2, 2, 6, 6, 7, 7, 5, 2, 0, 9, 1, 1, 9, 11, 7, 9, 2, 8, 9, 12, 1, 1, 14, 2, 12, 9, 7, 4, 9, 8,
1, 6, 14, 5, 9, 0, 1, 4, 8, 8, 6, 9, 4, 4, 3, 11, 4, 3, 11, 14, 9, 12, 1, 7, 11, 11, 0, 0, 1, 1, 0, 11, 10, 3, 11, 11,
3, 6, 1, 4, 11, 2, 3, 6, 10, 10, 9, 11, 7, 3, 2, 14, 3, 3, 10, 1, 8, 14, 3, 9, 10, 13, 3, 2, 1, 3, 14, 14, 12, 6, 13,
13, 5, 8, 3, 6, 10, 1, 6, 5, 10, 9, 11, 11, 9, 6, 4, 13, 5, 5, 12, 0, 10, 13, 6, 11, 13, 0, 5, 5, 3, 6, 1, 13, 11, 8,
12, 12, 4, 10, 3, 8, 13, 3, 5, 8, 12, 12, 10, 13, 8, 8, 6, 0, 7, 7, 0, 2, 13, 0, 5, 11, 0, 0, 4, 4, 5, 5, 3, 0, 13, 7,
0, 14, 7, 9, 5, 8, 0, 6, 7, 10, 14, 14, 12, 0, 10, 7, 6, 2, 7, 6, 14, 5, 12, 3, 7, 13, 14, 2, 6, 6, 4, 7, 3, 2, 1, 9,
2, 2, 9, 12, 7, 10, 14, 5, 9, 9, 13, 13, 0, 14, 13, 9, 8, 2, 9, 9, 1, 4, 14, 2, 9, 0, 1, 4, 9, 8, 7, 9, 5, 2, 0, 12, 1,
1, 8, 14, 6, 12, 1, 7, 9, 11, 1, 0, 14, 2, 12, 12, 10, 4, 11, 11, 3, 6, 1, 4, 9, 14, 4, 3, 8, 8, 9, 9, 7, 4, 2, 11, 3,
3, 10, 13, 9, 11, 4, 9, 11, 14, 3, 3, 1, 4, 14, 11, 9, 6, 10, 10, 3, 8, 1, 6, 11, 2, 3, 6, 10, 10, 8, 11, 6, 6, 4,
13, 6, 5, 13, 0, 11, 14, 3, 9, 13, 13, 2, 2, 3, 3, 1, 13, 12, 5, 13, 12, 5, 8, 3, 6, 13, 4, 5, 8, 12, 12, 10, 13,
9, 5, 4, 0, 5, 5, 12, 3, 10, 1, 5, 11, 12, 0, 4, 4, 3, 5, 1, 0, 14, 8, 0, 0, 7, 10, 5, 8, 12, 3, 7, 7, 12, 11, 13, 12,
11, 8, 6, 0, 7, 7, 14, 2, 12, 0, 7, 13, 0, 2, 7, 6, 5, 8, 3, 0, 13, 10, 14, 14, 6, 12, 4, 10, 0, 5, 7, 9, 14, 14, 12,
0, 10, 10, 8, 2, 9, 9, (sequence restarts:) 1, 4, 0, 2, 7, 12, 2, 2, 7, 7, 5, ...

Period length is $432 < 15 \cdot 17 \cdot 26 / 2$.

Lehmer generator produces sequence $\{x_n\}$ defined by relations

$$0 < x_0 < M, \quad x_0 \text{ coprime to } M.$$

$$x_{n+1} = Ax_n \bmod M, \quad n \geq 0.$$

Modulus M, seed x_0, multiplier A.

Example 9

$$x_0 = 1,$$

$$x_{n+1} = 6x_n \bmod 13.$$

$$\{x_n\} = \underbrace{1, 6, 10, 8, 9, 2, 12, 7, 3, 5, 4, 11,}_{\text{sequence period, length = 12}} 1, 6, 10, 8, 9, 2, 12, \dots$$

Example 10

$$x_0 = 2,$$

$$x_{n+1} = 5x_n \bmod 13.$$

$$\{x_n\} = \underbrace{2, 10, 11, 3,}_{\text{sequence period, length = 4}} 2, 10, 11, 3, 2, 10, 11, 3, \dots$$

$$0 < x_0 < M, \qquad x_0 \text{ coprime to } M.$$

$$x_{n+1} = Ax_n \bmod M, \qquad n \geq 0.$$

Fact

The sequence period length is maximal and equal to $M-1$ if
M is prime and
A is a primitive root of the multiplicative group of integers modulo M,

Primitive root

G is a primitive root of the multiplicative group of integers modulo M if
$\{G, G^2, G^3, ..., G^{M-1}\} = \{1, 2, 3 \ldots, M-1\}$ (all powers are taken modulo M)

Example 11

$M = 13$, $G = 6$,
$\{G, G^2, G^3, ..., G^{12}\} = \{6, 10, 8, 9, 2, 12, 7, 3, 5, 4, 11, 1\}$
$= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$, G is a primitive root.
$M = 13$, $G = 2$,
$\{G, G^2, G^3, ..., G^{12}\} = \{2, 4, 8, 3, 6, 12, 11, 9, 5, 10, 7, 1\}$
$= \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$, G is a primitive root.
$M = 13$, $G = 5$,
$(G, G^2, G^3, ..., G^{12}) = \{5, 12, 8, 1, 5, 12, 8, 1, 5, 12, 8, 1\}$, G is not a primitive root.

Finding group primitive roots

No elementary and effective method is known.
Special cases has been studied in detail.

Multiplicative group of integers modulo $M_{31} = 2^{31}–1 = 2\ 147\ 483\ 647$.

G is a primitive root iff $G \equiv 7^b \pmod{M_{31}}$ where b is coprime to $M_{31}–1$.
The prime factors $M_{31}–1$ are 2, 3, 7,11, 31, 151, 331.
($M_{31}–1 = 2·3^2·7·11·31·151·331$)

Example 12

$G = 7^5 = 16807$ is a primitive root because 5 is coprime to $M_{31}–1$.

$G = 7^{1116395447} = 48271$ is a primitive root because 1116395447 is a prime, therefore it is coprime to $M_{31}–1$.

$G = 7^{1058580763} = 69621$ is a primitive root because $69621 = 19·41·61·22277$ therefore 69621 is coprime to $M_{31}–1$.

Blum Blum Shub Generator

Blum Blum Shub generator produces sequence $\{x_n\}$ defined by relations

$$2 \leq x_0 < M, \quad x_0 \text{ coprime to } M.$$
$$x_{n+1} = x_n^2 \bmod M$$

Modulus M, seed x_0.

Seed x_0 coprime to M.
Modulus M is a product of two big primes P and Q.
$P \bmod 4 = \mathrm{Q} \bmod 4 = 3$,
$\gcd(\varphi(\mathrm{P}-1), \varphi(\mathrm{Q}-1))$ should be small, (cannot be 1).

Example 13 $x_0 = 4, \ \ M = 11 \cdot 47,$ $\gcd(\varphi(10), \varphi(46)) = \gcd(4, 22) = 2,$

$$x_{n+1} = x_n^2 \bmod 517.$$

$$\{x_n\} = \underbrace{\underline{4,}\ 16, 256, 394, 136, 401, 14, 196, 158, 148, 190, 427, 345, 115, 300, 42, 213, 390, 102, 64, 477, 49, 333, 251, 444, 159, 465, 119, 202, 478, 487, 383, 378, 192, 157, 350, 488, 324, 25, 108, 290, 346, 289, 284,}_{\text{sequence period, length = 44}}\ \underline{4,}\ 16, 256, 394, 136, \ldots$$

Primes related notions

Prime counting function $\pi(n)$

Counts the number of prime numbers less than or equal to n.

Example 14

$\pi(10)$ = 4. Primes less than or equal to 10: 2, 3, 5, 7.

$\pi(37)$ = 12. Primes less than or equal to 37: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37.

$\pi(100)$ = 25. Primes less than or equal to 100: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97.

Estimate

$$\frac{n}{\ln n} < \pi(n) < 1.25506 \frac{n}{\ln n} \quad \text{for } n > 16.$$

Example 15

$$\frac{100}{\ln 100} < \pi(100) < 1.25506 \frac{100}{\ln 100} \qquad \frac{10^6}{\ln 10^6} < \pi(10^6) < 1.25506 \frac{10^6}{\ln 10^6}$$

$$21.715 < \pi(100) = 25 < 27.253 \qquad 72382.4 < \pi(10^6) = 78498 < 90844.3$$

Limit behaviour

$$\lim_{n \to \infty} \frac{\pi(n)}{\frac{n}{\ln n}} = 1$$

Primes related notions

Euler's totient function $\varphi(n)$

Counts the positive integers less than or equal to n that are relatively prime to n.

Example 16

$n = 21$, $\varphi(21) = 12$.
coprimes to 21, smaller than 21: 1, 2, 4, 5, 8, 10, 11 13, 16, 17, 19, 20.
$n = 24$, $\varphi(24) = 8$.
coprimes to 24, smaller than 24: 1, 5, 7, 11, 13, 17, 19, 23.

Mersenne prime M_n

Mersenne prime M_n is a prime in the form $2^n–1$, for some $n > 1$.

Example 17

$n = 3$, $M_3 = 2^3–1 = 7$,
$n = 7$, $M_7 = 2^7–1 = 127$,
$n = 31$, $M_{31} = 2^{31}–1 = 2147483647$.

Sieve of Eratosthenes

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Sieve of Eratosthenes

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Sieve of Eratosthenes

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Sieve of Eratosthenes

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Sieve of Eratosthenes

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Sieve of Eratosthenes

Algorithm

EratosthenesSieve (n)

Let A be an array of Boolean values, indexed by integers 2 to n, initially all set to **true**

for i = 2 **to** $\sqrt{n}$

if $A[i]$ = **true then**

for $j = i^2, i^2+i, i^2+2i, i^2+3i$, ..., not exceeding n

$A[i]$:= **false**

end

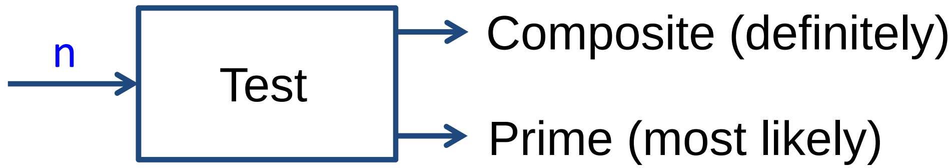
output all i such that $A[i]$ is **true**

end

Time complexity: $O(n \log \log n)$.

Randomized primality tests

General scheme



Fermat (little) theorem

If p is prime and $0 < a <$ p, then $a^{p-1} \equiv 1 \pmod{p}$.

Fermat primality test

FermatTest (n, k)
    for i = 1 **to** k
        a = random integer in [2, n-2]
        if $a^{n-1} \not\equiv 1 \pmod{n}$ **then return** *Composite*
    end
    return *Prime*
end

Flaw: There are infinitely many composite numbers for which the test always fails.
(Carmichael numbers: 561, 1105, 1729, 2465, …)

Miller-Rabin primality test

Lemma: If p is prime and $x^2 \equiv 1 \pmod{p}$ then $x \equiv 1 \pmod{p}$ or $x \equiv -1 \pmod{p}$.

⇨ Let $n > 2$ be prime, $n-1 = 2^r \cdot d$ where d is odd, $1 < a < n\text{-}1$.
Then either $a^d \equiv 1 \pmod{n}$ or $a^{2^s \cdot d} \equiv -1 \pmod{n}$ for some $0 \le s \le r-1$.

MillerRabinTest (n, k)
    compute r, d such that d is odd and $2^r \cdot d = n\text{-}1$
    for $i = 1$ **to** k // WitnessLoop
        a = random integer in $[2, n\text{-}2]$
        $x = a^d \bmod n$
        if $x = 1$ or $x = n-1$ **then goto** EndOfLoop
        for $j = 1$ **to** $r\text{-}1$
            $x = x^2 \bmod n$
            if $x = 1$ **then return** *Composite*
            if $x = n-1$ **then goto** EndOfLoop
        end
        return *Composite*
        EndOfLoop:
    end
    return *Prime*
end

Examples:

$n = 1105 = 2^4 \cdot 69 + 1$
$a = 389$
$x_0 = 1039$
$x_1 = 1041$
$x_2 = 781$
$x_3 = 1$ -> Composite

$n = 1105 = 2^4 \cdot 69 + 1$
$a = 390$
$x_0 = 539$
$x_1 = 1011$
$x_2 = 1101$
$x_3 = 16$
-> Composite

$n = 13 = 2^2 \cdot 3 + 1$
$a = 7$
$x_0 = 5$
$x_1 = 12 \equiv -1 \pmod{13}$
WitnessLoop passes

Randomized primality tests

Miller-Rabin primality test

- Time complexity: $\mathrm{O}(k \log^3 n)$.
- If n is composite then the test declares n prime with a probability at most 4^{-k}.
- A deterministic variant exists, however it relies on unproven generalized Riemann hypothesis.

AKS primality test

- First known deterministic polynomial-time primality test.
- Agrawal, Kayal, Saxena, 2002 - Gödel Prize in 2006.
- Time complexity: $\mathrm{O}(\log^6 n)$.
- The algorithm is of immense theoretical importance, but not used in practice.

Difficulty of the problem

- No efficient algorithm is known.
- The presumed difficulty is at the heart of widely used algorithms in cryptography (RSA).

Pollard's rho algorithm

- Effective for a composite number having a small prime factor.

```
PollardRho (n)
    x = y = 2; d = 1
    while d = 1
        x = g(x) mod n
        y = g(g(y)) mod n
        d = gcd (|x-y|, n)
    end
    if d = n return Failure
    else return d
end
```

g(x) .. a suitable polynomial function

For example, $g(x) = x^2 - 1$

gcd .. the greatest common divisor

Pollard's rho algorithm – analysis

- Assume $n = pq$.
- Values of x and y form two sequences $\{x_k\}$ and $\{y_k\}$, respectively, where y_k=x_{2k} for each k. Both sequences enter a cycle. This implies there is t such that y_t=x_t.
- Sequences $\{x_k \bmod p\}$ and $\{y_k \bmod p\}$ typically enter a cycle of shorter length. If, for some $s < t$, $x_s \equiv y_s \pmod p$, then p divides $|x_s$-$y_s|$ and the algorithm halts.

- The expected number of iterations is O($\sqrt{p}$)=O($n^{1/4}$) .

References

T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein: Introduction to Algorithms, 3rd ed., MIT Press, 2009, Chapter 31 Number-Theoretic Algorithms

Stephen K. Park, Keith W. MIller: Random number generators: good ones are hard to find, Communications of the ACM, Volume 31 Issue 10, Oct. 1988

Pierre L'Ecuyer: Efficient and portable combined random number generators, Communications of the ACM, Volume 31 Issue 6, June 1988