Sage Quick Reference: Elementary Number Theory

William Stein (modified by nu)
Sage Version 3.4
http://wiki.sagemath.org/quickref

以下 $m, n, a, b, \ldots$ は `ZZ` の元とする.
`ZZ` $= \mathbb{Z} =$ 全ての整数

ORGINAL TEXT
Everywhere $m, n, a, b, etc.$ are elements of `ZZ`
`ZZ` $= \mathbb{Z} =$ all integers

整数 Integers

$$\ldots, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, \ldots$$

n を m で割ると余りは `n % m`
`gcd(n,m)`, `gcd(`*list*`)`
拡張された公約数 $g = sa + tb = \gcd(a, b)$: `g,s,t=xgcd(a,b)`
`lcm(n,m)`, `lcm(`*list*`)`
二項係数 $\binom{m}{n}$ = `binomial(m,n)`
base 進法による表示: `n.digits(`*base*`)`
base 進法による桁数: `n.ndigits(`*base*`)`
(*base* は省略可, デフォルトは 10)
割り切る. $n \mid m$: `n.divides(m)`, $nk = m$ を満たす k があるか.
約数 – $d \mid n$ を満たす d 達 : `n.divisors()`
階乗 – $n!$ = `n.factorial()`

ORGINAL TEXT
n divided by m has *remainder* `n % m`
`gcd(n,m)`, `gcd(`*list*`)`
extended gcd $g = sa + tb = \gcd(a, b)$: `g,s,t=xgcd(a,b)`
`lcm(n,m)`, `lcm(`*list*`)`
binomial coefficient $\binom{m}{n}$ = `binomial(m,n)`
digits in a given base: `n.digits(`*base*`)`
number of digits: `n.ndigits(`*base*`)`
(*base* is optional and defaults to 10)
divides $n \mid m$: `n.divides(m)` if $nk = m$ some k
divisors – all d with $d \mid n$: `n.divisors()`
factorial – $n!$ = `n.factorial()`

素数 Prime Numbers

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, \ldots$$

素因数分解: `factor(n)`
素数判定: `is_prime(n)`, `is_pseudoprime(n)`
素冪判定: `is_prime_power(n)`
$\pi(x) = \#\{p : p \leq x \text{ is prime}\}$ = `prime_pi(x)`
素数の集合: `Primes()`
$\{p : m \leq p < n \text{ and } p \text{ prime}\}$ =`prime_range(m,n)`
n 以上 m 以下の素冪の集合: `prime_powers(m,n)`
最初の n 個の素数: `primes_first_n(n)`
次の素数, ひとつ前の素数: `next_prime(n)`,
`previous_prime(n)`, `next_probable_prime(n)`
次の素冪, ひとつ前の素冪: `next_prime_power(n)`,
`pevious_prime_power(n)`
$2^p - 1$ の素数性に関する Lucas-Lehmer テスト

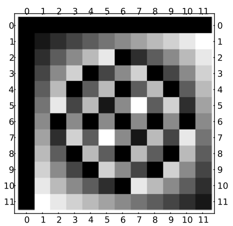
```
def is_prime_lucas_lehmer(p):
    s = Mod(4, 2^p - 1)
    for i in range(3, p+1): s = s^2 - 2
    return s == 0
```

ORGINAL TEXT
factorization: `factor(n)`
primality testing: `is_prime(n)`, `is_pseudoprime(n)`
prime power testing: `is_prime_power(n)`
$\pi(x) = \#\{p : p \leq x \text{ is prime}\}$ = `prime_pi(x)`
set of prime numbers: `Primes()`
$\{p : m \leq p < n \text{ and } p \text{ prime}\}$ =`prime_range(m,n)`
prime powers: `prime_powers(m,n)`
first n primes: `primes_first_n(n)`
next and previous primes: `next_prime(n)`,
`previous_prime(n)`, `next_probable_prime(n)`
prime powers: `next_prime_power(n)`,
`pevious_prime_power(n)`
Lucas-Lehmer test for primality of $2^p - 1$

```
def is_prime_lucas_lehmer(p):
    s = Mod(4, 2^p - 1)
    for i in range(3, p+1): s = s^2 - 2
    return s == 0
```

合同式, モジュラ計算 Modular Arithmetic and Congruences

```
k=12; m = matrix(ZZ, k, [(i*j)%k for i in [0..k-1] for j in [0..k-1]]); m.plot(cmap='gray')
```

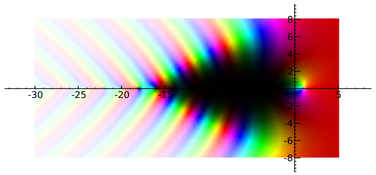


オイラーの $\phi(n)$ 関数: `euler_phi(n)`
クロネッカーシンボル $\left(\frac{a}{b}\right)$ = `kronecker_symbol(a,b)`
平方剰余: `quadratic_residues(n)`
平方非剰余: `quadratic_residues(n)`
環 $\mathbb{Z}/n\mathbb{Z}$ = `Zmod(n) = IntegerModRing(n)`
$\mathbb{Z}/n\mathbb{Z}$ の元としての a ($a \bmod n$): `Mod(a, n)`
$\mathbb{Z}/n\mathbb{Z}$ での原始根 = `primitive_root(n)`
$\mathbb{Z}/n\mathbb{Z}$ での逆元: `n.inverse_mod(m)`
$\mathbb{Z}/n\mathbb{Z}$ での冪 $a^n \pmod{m}$: `power_mod(a, n, m)`
中国の剰余定理: `x = crt(a,b,m,n)`
$x \equiv a \pmod{m}$ かつ $x \equiv b \pmod{n}$ を満たす x を探す
離散対数: `log(Mod(6,7), Mod(3,7))`
$a \pmod{n}$ の次数 = `Mod(a,n).multiplicative_order()`
$a \pmod{n}$ の平方根 = `Mod(a,n).sqrt()`

ORGINAL TEXT
Euler's $\phi(n)$ function: `euler_phi(n)`
Kronecker symbol $\left(\frac{a}{b}\right)$ = `kronecker_symbol(a,b)`
Quadratic residues: `quadratic_residues(n)`
Quadratic non-residues: `quadratic_residues(n)`
ring $\mathbb{Z}/n\mathbb{Z}$ = `Zmod(n) = IntegerModRing(n)`
a modulo n as element of $\mathbb{Z}/n\mathbb{Z}$: `Mod(a, n)`
primitive root modulo n = `primitive_root(n)`
inverse of $n \pmod{m}$: `n.inverse_mod(m)`
power $a^n \pmod{m}$: `power_mod(a, n, m)`
Chinese remainder theorem: `x = crt(a,b,m,n)`
finds x with $x \equiv a \pmod{m}$ and $x \equiv b \pmod{n}$
discrete log: `log(Mod(6,7), Mod(3,7))`
order of $a \pmod{n}$ = `Mod(a,n).multiplicative_order()`
square root of $a \pmod{n}$ = `Mod(a,n).sqrt()`

特殊函数 Special Functions

```
complex_plot(zeta, (-30,5), (-8,8))
```



$$\zeta(s) = \prod_p \frac{1}{1 - p^{-s}} = \sum \frac{1}{n^s} = \texttt{zeta(s)}$$

$$\mathrm{Li}(x) = \int_2^x \frac{1}{\log(t)} dt = \texttt{Li(x)}$$

$$\Gamma(s) = \int_0^\infty t^{s-1} e^{-t} dt = \texttt{gamma(s)}$$

ORGINAL TEXT

$$\zeta(s) = \prod_p \frac{1}{1 - p^{-s}} = \sum \frac{1}{n^s} = \texttt{zeta(s)}$$

$$\mathrm{Li}(x) = \int_2^x \frac{1}{\log(t)} dt = \texttt{Li(x)}$$

$$\Gamma(s) = \int_0^\infty t^{s-1} e^{-t} dt = \texttt{gamma(s)}$$

連分数 Continued Fractions

```
continued_fraction(pi)
```

$$\pi = 3 + \cfrac{1}{7 + \cfrac{1}{15 + \cfrac{1}{1 + \cfrac{1}{292 + \cdots}}}}$$

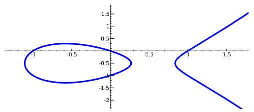
連分数: `c=continued_fraction(x,`*bits*`)`
近似分数 (達): `c.convergents()`
部分分子 $p_n =$ `c.pn(n)`
部分分母 $q_n =$ `c.qn(n)`
値: `c.value()`

. . . . ORGINAL TEXT

continued fraction: `c=continued_fraction(x,`*bits*`)`
convergents: `c.convergents()`
convergent numerator $p_n =$ `c.pn(n)`
convergent denominator $q_n =$ `c.qn(n)`
value: `c.value()`

楕円曲線 Elliptic Curves

```
EllipticCurve([0,0,1,-1,0]).plot(plot_points=300,thickness=3)
```



`E = EllipticCurve([`a_1, a_2, a_3, a_4, a_6`])`

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

E の導手 (conductor) $N =$ `E.conductor()`
E の判別式 $\Delta =$`E.discriminant()`
E の階数 $=$ `E.rank()`
$E(\mathbb{Q})$ の自由生成系 $=$ `E.gens()`
j-invariant $=$ `E.j_invariant()`
$N_p = \#\{$modulo p での E の解 $\} =$ `E.Np(`*prime*`)`
$a_p = p + 1 - N_p =$`E.ap(`*prime*`)`
$L(E,s) = \sum \frac{a_n}{n^s} =$ `E.lseries()`
$\mathrm{ord}_{s=1} L(E,s) =$ `E.analytic_rank()`

. . . . ORGINAL TEXT

`E = EllipticCurve([`a_1, a_2, a_3, a_4, a_6`])`

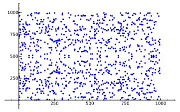
$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

conductor N of $E =$`E.conductor()`
discriminant Δ of $E =$`E.discriminant()`
rank of $E =$ `E.rank()`
free generators for $E(\mathbb{Q}) =$ `E.gens()`
j-invariant $=$ `E.j_invariant()`
$N_p = \#\{$solutions to E modulo $p\} =$ `E.Np(`*prime*`)`
$a_p = p + 1 - N_p =$`E.ap(`*prime*`)`
$L(E,s) = \sum \frac{a_n}{n^s} =$ `E.lseries()`
$\mathrm{ord}_{s=1} L(E,s) =$ `E.analytic_rank()`

---

p で合同な楕円曲線 Elliptic Curves Modulo p

```
EllipticCurve(GF(997), [0,0,1,-1,0]).plot()
```



`E = EllipticCurve(GF(p), [`a_1, a_2, a_3, a_4, a_6`])`
$\#E(\mathbb{F}_p) =$ `E.cardinality()`
$E(\mathbb{F}_p)$ の生成系 $=$ `E.gens()`
$E(\mathbb{F}_p) =$`E.points()`

. . . . ORGINAL TEXT

`E = EllipticCurve(GF(p), [`a_1, a_2, a_3, a_4, a_6`])`
$\#E(\mathbb{F}_p) =$ `E.cardinality()`
generators for $E(\mathbb{F}_p) =$ `E.gens()`
$E(\mathbb{F}_p) =$`E.points()`