Contents

Chapter 1

Functions

1.1 factor.methods – factoring methods

It uses methods of **factor.find** module or some heavier methods of related modules to find a factor. Also, classes of **factor.util** module is used to track the factorization process. `options` are normally passed to the underlying function without modification.

This module uses the following type:

factorlist :

`factorlist` is a list which consists of pairs `(base, index)`. Each pair means $base^{index}$. The product of these terms expresses prime factorization.

1.1.1 factor – easiest way to factor

factor(`n`: *integer*, `method`: *string*='default', **`options`)
→ factorlist

Factor the given positive integer `n`.

By default, use several methods internally.

The optional argument `method` can be:

- `'ecm'`: use elliptic curve method.
- `'mpqs'`: use MPQS method.
- `'pmom'`: use $p-1$ method.
- `'rhomethod'`: use Pollard's ρ method.
- `'trialDivision'`: use trial division.

(†In fact, the initial letter of method name suffices to specify.)

1.1.2 ecm – elliptic curve method

ecm(n: *integer*, **options) → factorlist

Factor the given integer n by elliptic curve method.

(See **ecm** of **factor.ecm** module.)

1.1.3 mpqs – multi-polynomial quadratic sieve method

mpqs(n: *integer*, **options) → factorlist

Factor the given integer n by multi-polynomial quadratic sieve method.

(See **mpqsfind** of **factor.mpqs** module.)

1.1.4 pmom – $p-1$ method

pmom(n: *integer*, **options) → factorlist

Factor the given integer n by $p-1$ method.

The method may fail unless n has an appropriate factor for the method.
(See **pmom** of **factor.find** module.)

1.1.5 rhomethod – ρ method

rhomethod(n: *integer*, **options) → factorlist

Factor the given integer n by Pollard's ρ method.

The method is a probabilistic method, possibly fails in factorizations.
(See **rhomethod** of **factor.find** module.)

1.1.6 trialDivision – trial division

trialDivision(n: *integer*, **options) → factorlist

Factor the given integer n by trial division.

options for the trial sequence can be either:

1. `start` and `stop` as range parameters.
2. `iterator` as an iterator of primes.
3. `eratosthenes` as an upper bound to make prime sequence by sieve.

If none of the options above are given, the function divides `n` by primes from 2 to the floor of the square root of `n` until a non-trivial factor is found.
(See **trialDivision** of **factor.find** module.)

Examples

```
>>> factor.methods.factor(10001)
[(73, 1), (137, 1)]
>>> factor.methods.ecm(1000001)
[(101L, 1), (9901L, 1)]
```

Bibliography