Contents

Chapter 1

Functions

1.1 factor.find – find a factor

このモジュールの方法は与えられた整数に対して一つの要素を返す。非自明な要素wさがすことができない場合は 1 を返す。しかし 1 も要素であることお忘れなく。

`verbose` boolean flag can be specified for verbose reports. このメッセージを受け取るため、logger を準備してください。 (logging 参照。)

1.1.1 trialDivision – trial division

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

試割り算によって得る `n` の要素を返す。

`options` は以下のどちらか:

1. `start` と `stop` は範囲パラメータ。 さらには `step` も利用可。
2. `iterator` は素数のイテレータ。

`options` が与えられない場合、この関数は非自明な要素がみつかるまで `n` を素数 2
から `n` の二乗までの数で割っていく。 `verbose` boolean flag can be specified for verbose reports.

1.1.2 pmom – $p-1$ method

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

$p-1$ 法を使い `n` の要素を返す。

この関数は [1] のアルゴリズム 8.8.2 ($p-1$ first stage) を使って　n の非自明な要素を探すよう試みる。
　$n = 2^i$ の場合、 この関数はループにおちいる。自然法によってこの方法は自明な要素しか返さないかもしれない。

verbose Boolean flag can be specified for verbose reports, though it is not so verbose indeed.

1.1.3　rhomethod – ρ method

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

　Pollard の ρ 法より n の要素を返す。

この実装は [2] の説明に言及する。. 自然法によって因数分解は自明な要素しか返さないかもしれない。

verbose Boolean flag can be specified for verbose reports.

Examples

```
>>> factor.find.trialDivision(1001)
7
>>> factor.find.trialDivision(1001, start=10, stop=32)
11
>>> factor.find.pmom(1001)
91
>>> import logging
>>> logging.basicConfig()
>>> factor.find.rhomethod(1001, verbose=True)
INFO:nzmath.factor.find:887 748
13
```

Bibliography

[1] Henri Cohen. *A Course in Computational Algebraic Number Theory.* GTM138. Springer, 1st. edition, 1993.

[2] Richard Crandall and Carl Pomerance. *Prime Numbers.* Springer, 1st. edition, 2001.