JAPLA 研究会資料　2019/5/18

Ｊによる平方剰余(Quadratic Residue)－整数論シリーズ３
－Ｊでの合同式演算、ルジャンドル表記を見やすくする－

西川　利男

整数論の教科書で、しばらく読み進むと、平方剰余という語が出てきて、大いに戸惑う。似たような字ずらだが、平方根と比べてどうだろうか。

これは、ガウスの黄金則(Theorema Aureum, Golden Theorem)という聖なる山への登山道だという。このあたりから整数論は雲の上の数学となってしまう。

整数論のほとんどの教科書で良くないことは、このように黄金則などと崇拝することに終始して、凡人に対しては、それで「何ができる」「何がわかる」とそのメリットを明らかにしてくれない。

唯一、高木貞治の「初等整数論講義」[3] (p.71)では、平方剰余を使って、「どんな数でも４つまでの平方和で表せる」と一つの例を示してくれていた。ただ、数値例はなく私には不満だった。

Ｊの助けを借りて、何とかこの山登りをなしとげたいものだと始めた。
なお、前回、２回にわたる私の整数論関連のレポートを含めて、整数論シリーズとした。

[1] 西川利男「Ｊでピタゴラス数(２つの平方数に分解)を求める
　　Ｊによるフェルマーの無限降下法」JAPLA 研究会資料 2019/1/19
[2] 西川利男「Ｊでピタゴラス数(２つの平方数の和に分解)を求める－その２
　　自然数に対するピタゴラス数」JAPLA 研究会資料　2019/3/9
[3] 高木貞治「初等整数論講義、第２版」共立出版(1977)

まずは、いままでの四則演算とは一見、似ているように見えるが違う合同式演算に慣れることが第一である。これをもっと見やすくすることから始めた。

また、今回もシルバーマンの教科書、第 22 章(p.131-)にそって、話題をすすめる。

[4] ジョセフ H. シルバーマン、鈴木治朗訳「はじめての数論」
　　ピアソンエデュケーション (2001).

１．Ｊの世界の合同式の表記

整数論の数学書では、法7とする合同式でははつぎのように表記される。

$5+15 \equiv 6 \pmod 7$

この意味は、5 + 15 = 20 を 7 で割った余りは6となる。

これを、Ｊの世界で、つぎのように対応させて、表示する。

```
   (5, 15) add_mod 7
6
```

その中身は、Ｊの以下のコードと同じだが、数学の表記と違和感がないようにした。

```
   7 | 5 + 15
6
```

同様にして、数学書の次の掛け算合同式は

$5*15 \equiv 5 \pmod 7$

Ｊでは、つぎのように表示する。

```
   (5, 15) mul_mod 7
5
   7 | 5 * 15
5
```

ここまではあまり代わり映えはしないように見えるが、もっと多くの数の場合でその効果を見てみよう。

```
   (1 2 3 4 5 6 7 8 9 10 11 12) mul_mod 13
12
```

つまり、以下の計算をやっていることになる。

```
   */ 1 2 3 4 5 6 7 8 9 10 11 12
479001600
   13 | */ 1 2 3 4 5 6 7 8 9 10 11 12
12
```

ただし、前回[2]に示したように、いきなり初めに積をとりその大きな数で計算にするのではない。それぞれの数の mod | をとってそれらの小さな数についてその積を計算しているのである。

以下にＪによる add_mod と mul_mod の定義を示す。

```
add_mod =: 3 : 0
:
y | +/ y | x
)
```

```
mul_mod =: 3 : 0
:
y | */ y | x
)
```

２．平方剰余(Quadratic Residue)とは

四則演算と同様にして、２乗の演算を行うことも出来る。これが、あらためて定義された平方剰余(Quadratic Residue)である。

たとえば、法を7とした、0から6までの数の平方剰余は次のようになる。

```
  (i.7) sq_mod 7
 0 1 4 2 2 4 1
```

つづいて、法11, 13, 17のそれぞれの平方剰余の例を示してみよう。

```
  (i. 11) sq_mod 11
 0 1 4 9 5 3 3 5 9 4 1
  (i. 13) sq_mod 13
 0 1 4 9 3 12 10 10 12 3 9 4 1
  (i.17) sq_mod 17
 0 1 4 9 16 8 2 15 13 13 15 2 8 16 9 4 1
```

Ｊの定義はつぎのようになる。

```
NB. 平方剰余 Quadratic Residue ================
sq_mod =: 3 : 0
:
y | *: y | x
 )
```

３．平方剰余の積法則とルジャンドル記号

次に、例えば i.13

```
0 1 2 3 4 5 6 7 8 9 10 11 12
```

の値について、その中のある数が13を法とした平方剰余のグループに含まれるか、どうかで２つのグループに分かれる。

これは、Ｊのプリミティブe.を用いて、簡単に得られる。

```
  4 e. (i. 13) sq_mod 13 => 1       NB. QR(Quadratic Residue)
```

4は7を法とした平方剰余である。

```
  5 e. (i. 13) sq_mod 13 => 0       NB. NR(Nonquadratic Residue)
```

5は7を法とした平方剰余ではない。

ところが、整数論の教科書では、次のようなルジャンドル記号という表記が好んで使われる。形こそ分数の形をしているが、分母に相当するのは法であり、分子は

対象とする数である。これは、一種の論理フラグであり、その論理演算として、これらの積がつぎのように演算される。

$$\left(\frac{4}{13}\right)=1,\quad \left(\frac{5}{13}\right)=-1,\quad \left(\frac{4}{13}\right)\left(\frac{5}{13}\right)=-1$$

この表記法が、私にとっては思ったより分かりにくかった。

Jでも、これに対応した表示を工夫した。

つぎのようになる。

```
  4 qr 13 => 1      平方剰余 Quadratic Residue QR
  5 qr 13 => _1     非平方剰余 Nonquadratic Residue NR
(4 qr 13) * (5 qr 13)  => _1
```

整数論で、平方剰余の積の法則と呼ばれているものがこれで、分母にあたる法が奇の素数であるときにはいつでも成り立つ。証明は整数論の本[3]にある。

ここで、このような論理のフラグについて、一言コメントする。

Jをはじめとして一般にコンピュータの世界では、論理の真理値(真True Yes vs. 偽False No)に対して(1, 0)としている。しかし、ルジャンドル記号では(1, -1)となっている。どちらが良いかは、一概に言えないが、以下の定義の中で、zzはその変換をするものである。

Jによる上のルジャンドル記号表記の定義は以下に示す。

```
NB. Legendre Notation ==============
zz =: <:`]@.]"(0)
NB.     zz 0 => _1
NB.     zz 1 => 1
NB.     zz 0, 1 => _1 1
qr0 =: 3 : 0
(i. y.) e. (i. y.) sq_mod y.
)
qr =: 3 : 0
:
zz x. { qr0 y.
)
```

シルバーマンの本では、次のような例があげられていた。

75が97を法として、平方剰余の数かどうか（QRかNR）知りたいとする。

$$\left(\frac{75}{97}\right)=\left(\frac{3\cdot 5\cdot 5}{97}\right)=\left(\frac{3}{97}\right)\left(\frac{5}{97}\right)\left(\frac{5}{97}\right)=\left(\frac{3}{97}\right)=1$$

そのまま、計算しないで、75を素因数分解して、それぞれ小さい数にしてから計算する。後ろの２つの項は２度掛けているので、QR, NRによらず１になる。

Ｊでは次のようになる。

```
   q: 75
3 5 5
   3 qr 97
1
   5 qr 97
_1
   (3 qr 97) * (5 qr 97) * (5 qr 97)
1
```

もちろん、Ｊでは、つぎのように一度で求められる。

```
75 qr 97
1
```

今回はここまでにして、シルバーマン第２２章のp.137-8に練習問題があるのでいままで、定義したＪのツールを使ってやってみよう。

練習問題２２．１

19を法とした平方剰余および非剰余を求めよ。

```
   P =: (i.19) sq_mod 19
   P
0 1 4 9 16 6 17 11 7 5 5 7 11 17 6 16 9 4 1
   ~. P                    NB. ユニークなものだけ
0 1 4 9 16 6 17 11 7 5
   (/:{]) ~. P             NB.  昇順に並び変える
0 1 4 5 6 7 9 11 16 17
   P =: (/:{]) ~. P        NB.  平方剰余
   P
0 1 4 5 6 7 9 11 16 17
   Q =: (i.19) -. P
   Q                       NB.  非平方剰余
2 3 8 10 12 13 14 15 18
```

練習問題２２．２

素数ｐを入力して、

Ａ = 1 ＜ ａ ＜ ｐの範囲でｐを法として平方剰余であるようなａの和

Ｂ = 1 ＜ ａ ＜ ｐの範囲でｐを法として非平方剰余であるようなａの和

たとえば、p = 11 とすると

```
   }. i. 11
1 2 3 4 5 6 7 8 9 10
   (}. i.11) sq_mod 11
1 4 9 5 3 3 5 9 4 1
   ] A =: ~. (}. i.11) sq_mod 11
1 4 9 5 3
   ] B =: (}. i.11) -. A
2 6 7 8 10
   +/A
22
   +/B
33
```

問 (a) ｐ ＜ 100 である素数のすべてについて、Ａ と Ｂ とを書き出せ。

問 (b) Ａ + Ｂ　の値はどうなるか。

問 (c) Ａ mod ｐ と Ｂ mod ｐ とを計算せよ。

問 (d) Ａ = Ｂ となる素数はどれか？

問 (e) Ａ と Ｂ とのどちらが大きくなる傾向があるか？

シルバーマンの本にはないが、前回報告した素数の平方和分解の可能性[2] に関連した 4n+1 型素数、4n+3 型素数の判定のため、p mod 4 の値も付記した。

上の計算を行うため、Ｊで次のようなプログラムを作成した。

```
problem_222 =: 3 : 0"(0)
a =. ~. (}. i. y.) sq_mod y.
b =. (}. i.y.) -. a
A =. +/a
B =. +/b
y. , (A), (B), (A+B), (y.|(A)), (y.|(B)) , (4|y.)
)
results =: 3 : 0
wr '      p      A      B    A+B    p|A    p|B'
wr '  ---------------------------------------'
7j0 "': problem_222 (primes y.)
)
```

```
NB.    指定した数までの素数
  primes =: 3 : 0
P =. p: i. y
(y > P) # P
)
NB.    primes 20 => 2 3 5 7 11 13 17 19
```

実行した結果は次のようになった。

```
 results 100
```

p	A	B	A+B	p|A	p|B	4|p
2	1	0	1	1	0	2
3	1	2	3	1	2	3
5	5	5	10	0	0	1
7	7	14	21	0	0	3
11	22	33	55	0	0	3
13	39	39	78	0	0	1
17	68	68	136	0	0	1
19	76	95	171	0	0	3
23	92	161	253	0	0	3
29	203	203	406	0	0	1
31	186	279	465	0	0	3
37	333	333	666	0	0	1
41	410	410	820	0	0	1
43	430	473	903	0	0	3
47	423	658	1081	0	0	3
53	689	689	1378	0	0	1
59	767	944	1711	0	0	3
61	915	915	1830	0	0	1
67	1072	1139	2211	0	0	3
71	994	1491	2485	0	0	3
73	1314	1314	2628	0	0	1
79	1343	1738	3081	0	0	3
83	1577	1826	3403	0	0	3
89	1958	1958	3916	0	0	1
97	2328	2328	4656	0	0	1