JAPLA 研究会資料 2008/1/26

J602 の新しい機能 M. について（続き）

数の分割(Partition)とその J プログラム

西川　利男

J602 の新しい機能 M.(Memorize)について、J の Help-Vocabulary の記述を元に先月報告した[1]。その際、J のコーディング例としてあげられていたもう一つの例、分割数について今回ややくわしく検討した。

１．数の分割(Partition)とは

分割なる整数論の分野については、２人の天才数学者オイラーとインドの数学者ラマルジャンの貢献を欠かすことはできない。分割とは何か、およびそれを巡るエピソードを啓蒙書[2]によりのぞいてみよう。

数の分割(Partition)とは、与えられた数をより小さいか等しい数の和としてその数を表すことである。

例えば値４を得るために、自然数を組み合わせたすべての方法を列挙してみる。これが分割である。

```
4 = 1 + 1 + 1 + 1
4 = 2 + 1 + 1
4 = 3 + 1
4 = 2 + 2
4 = 4
```

この分割自身を得るプログラムは付録にあげてある。上の結果から項の順序を考えないならば５通りの方法がある。

分割が何通りあるかという数を分割数といい、p(n)と表す。

```
p(4) = 5
```

分割数 p(n)は n が小さいときは

```
p(1) = 1, p(2) = 2, p(3) = 3, p(4) = 5, p(5) = 7, p(6) = 11, ..
```

となるが、n が大きくなると、p(n)は急速に大きくなる。

```
p(10) = 42, p(20) = 627, p(30) = 560, p(40) = 37338,
p(100) = 190569292, p(200) = 3972999029388
```

これは J に限らずふつうのプログラム言語ではメモリオーバーか大変な時間がかかる。後で示すように J602 の M. を利用したプログラムでは一瞬で結果が出る。

２．２人の天才数学者－オイラーとラマルジャンのアプローチ

この問題に最初に取り組んだのはオイラーであった。彼は、関連した次の問題をあきらかにした。例えば次のような式の展開を考える。

$$f(x) = (1+x)(1+x^2)(1+x^3)(1+x^4)(1+x^5)(1+x^6)\cdots$$

$$= 1 + x + x^2 + 2x^3 + 2x^4 + 3x^5 + 4x^6 + 5x^7 + 6x^8 + 9x^{10} + \cdots$$

この関数は「生成関数」あるいは「母関数」と呼ばれる。

ここで例えば x^6 の係数は

x^6, x^{5+1}, x^{4+2}, x^{3+2+1}

の４通りとして得られるものである。一方これは値４を繰り返しのない(unique)整数の和として表すことになる。

このような手法は現在もっとも脚光をあびている「組合せ数学」の先駆となるものだが、彼自身、分割数 p(n) そのものについてはうまくいかなかったようである。

分割数 p(n) を与える生成関数はつぎのようになる[3]。

$$g(x) = \frac{1}{1-x} \times \frac{1}{1-x^2} \times \frac{1}{1-x^3} \times \cdots$$

$$= (1 + x + x^2 + \cdots)(1 + x^2 + x^4 + \cdots)(1 + x^3 + x^6 \cdots)\cdots$$

ラマルジャンはインドにいたときこの仕事に手をつけ、イギリスに渡ってハーディとの共同研究として分割の問題は再開されたのである。

彼らは分割数 p(n) を漸化式で求める次の式を見出した。

$$p(n) = p(n-1) + p(n-2)$$
$$- p(n-5) - p(n-7)$$
$$+ p(n-12) + p(n-15)$$
$$\cdots\cdots$$

この漸化式は次のようにしてつくられる。

① 各項は２つずつの対となり、その符号は正、正、負、負、… となる。

② 各対の中でインデックスは１ずつ増える。

すなわち最初の対では１増える。（１から２）

２番目の対では２増える。（５から７）

３番目の対では３増える。（１２から１５）

③ 先行する対の後ろのインデックスと次の対の前のインデックスは

最初は３で、次々に２ずつ増えていく。

p(n-2) につづく次のインデックスは 2+3=5 で p(n-5)

p(n-7) につづく次のインデックスは 7+5=12 で p(n-12)

p(n-15) につづく次のインデックスは 15+7=22 で p(n-22) …

これを手計算で行うとしたら大変な忍耐力を要する。途中で一箇所でも間違えたらそれから先はすべて水のあわである。

３．Ｊによる分割数の計算

分割数を求めるには、現在のわれわれにはＪという強力な武器がある。

実は J602 の Help-Vocabulary にある分割数のコーディングはこの漸化式のアルゴリズムにしたがっていることが分かった。

```
NB. Partition - Original Version / Help-Vocabulary
pn =: -/@(+/)@:($:"0)@rec ` (x:@(0&=)) @. (0>:]) M.
pnx=: -/@(+/)@:($:"0)@rec ` (x:@(0&=)) @. (0>:])
rec=: - (-: (*"1) _1 1 +/ 3 * ]) @ (>:@i.@>.@%:@((2%3)&*))

timer=: 6!:2
```

プログラムは各項のインデックスを作る部分recとそのインデックスにより項の値を取り出し合計する部分pnからなっている。またrec = r1@r0から成っている。

しかしながら、このコーディングはtacitを駆使したもので、たやすく分かるものではない。筆者なりにexplicitに書き直したプログラムにより解説しよう。

それぞれに対応した西川版のプログラムをpnn, recn, rn0, rn1 として作った。

```
NB. Partition - Nishikawa's Version / 2007/12/16 =================
rn0 =: 3 : 0
rr =. i. >. %: (2%3) * y
>: rr
)

rn1 =: 3 : 0
rnn =. 2 * (y-1)
fn rnn, (rnn+1)
)

fn =: 3 : 0 ^:(0)
if. 0 = y
  do. 1
    return.
  else.
    if. 0 = 2|y
      do. (>: y) + fn <: y
      else. (-: >: y) + fn <: y
    end.
end.
)

recn =: 3 : 0
y1 =. rn0 y
y2 =. rn1 y1
y3 =. y - y2
(2, (-:#y3)) $ y3
)

pnn =: 3 : 0 M.
if. 0 >: y
  do. 0 = y
  else.
      p0 =: recn y
      p1 =: pnn^:(0) p0
      p2 =: +/ p1
      p3 =: -/ p2
end.
)
```

NB. ==========
まずいろいろな引数に rn0 と rn1 とを連続して実行してみる。

```
   rn1 rn0 1
1 2
```

引数 2 から 6 に対しては

```
   rn1 rn0 2
1 5 2 7
   rn1 rn0 3
1 5 2 7
   rn1 rn0 6
1 5 2 7
```

引数 7 から 13 に対しては

```
   rn1 rn0 7
1 5 12 2 7 15
   rn1 rn0 13
1 5 12 2 7 15
```

引数 14 から 13 に対しては

```
   rn1 rn0 14
1 5 12 22 2 7 15 26
```

これらの値が先の漸化式の各項 p(n-j)における j であることがわかるだろう。
　これを使って recn は引数 n-j を与えるものである。なお、ここで正の値のみを使い、負の値は使わないようにする。

```
   recn 1
 0
_1
   recn 2
1 _3
0 _5
   recn 6
5  1
4 _1
   recn 7
6 2 _5
5 0 _8
   recn 14
13 9  2  _8
12 7 _1 _12
```

　上で得られたインデックスで示される項を選び出し、次の関数 pnn で合計をとる。このとき、関数の再帰呼び出しを効果的に用いている。p1, p2 は途中経過である。

```
   pnn 6
11
   p1
7 1
5 0
   p2
```

```
12 1

   pnn 14
135
   p1
101 30 2 0
 77 15 0 0
   p2
178 45 2 0
```

最後にJ602の新機能M.を用いたものと用いないものとの比較した実行例を示す。

```
   pnn″(0) i.18  NB. With M.
1 1 2 3 5 7 11 15 22 30 42 56 77 101 135 176 231 297
   timer 'pnn″(0) i.18'
0.000175162
```

関数pnnxはM.を使用していない。

```
   pnnx″(0) i.18 NB. Without M.
1 1 2 3 5 7 11 15 22 30 42 56 77 101 135 176 231 297
   timer 'pnnx″(0) i.18'
3.47918
```

次は巨大数になる例である。

```
   pnn 100
190569292
   timer 'x: pnn 100'
0.000111187
   x: pnn 200
3972999029388
   timer 'x: pnn 200'
0.000126273
```

ためしに、MapleVで行ってみた。numbpart(100)では可能だったが、numbpart(200)ではStack Overflow !でダメだった。

４．おわりに

ラマヌジャンは分割数を求める次の近似関数をつくった。

$$p(n) \approx \frac{\exp(\pi\sqrt{\frac{2n}{3}})}{4n\sqrt{3}} + \cdots$$

また、nとp(n)との合同関係など分割に関してさまざまな研究成果を残した。

ところでこの分割がどう実際に役立つのだろう。先の啓蒙書[2]によれば通信ライン、クレジットカード、さらには宇宙の物質とエネルギーの統一理論としての南部洋一郎に始まる超ひも理論にも用いられているという。

文献

[1] 西川利男「J602の新しい機能M.について」JAPLAシンポジウム2007/12/8

[2] C.クロースン著、好田順治訳「数学の不思議」草土社 p. 297
[3] リュ著、伊理正夫、伊理由理訳「組合せ数学Ⅰ」共立全書 p. 43

付録　数の分割と分割数のＪプログラム

```
NB. Partition
NB. Usage => part 6
NB. revised into J by T. Nishikawa, 2001/7/5, 2001/7/8
NB.   from N. Thomson, "APL Program for the Math. Classroom", p. 142
NB. R. Korfhage, "Discrete Computational Structures", Chap. 4, p. 96
wr=: 1!:2&2
NB. partition
part =: 3 : 0
P =. (((<:y.) #2) #: i. 2^<:y.),"(1) 1
i =. 0
j =. 0
while. i < 2^<:y.
  do. Q =. 0, (i{P) # >: i. y.
      R =. 2 -~/¥ Q
      if. -. */ 2 >: /¥ R do. goto_next. end.
      j =. >: j
      wr R
      label_next.
      i =. >: i
  end.
'Number of Partitions = ', ": j
)

   part 6
6
5 1
4 2
4 1 1
3 3
3 2 1
3 1 1 1
2 2 2
2 2 1 1
2 1 1 1 1
1 1 1 1 1 1
Number of Partitions = 11
```