整数分割　PARTITION　計算の話題

演算結果は果たして正解か？

中野嘉弘　（札幌市南区・８５才）

FAX 専　011-588-3354　　　yoshihiro@river.ocn.ne.jp

順列・組合や分割法など長大計算が正解か否か、どうしたら分かるのか？
確かめて見た。　その例を話題とする。

０．　は　し　が　き

「はしがき」の冒頭は筆者の旧稿「組合せ（Combination）計算の話題」（文献０）のそれと共通である。
JAPLA １月例会の西川論文「J602 の新機能　M. について（続き）」　（文献１）を拝見中に気付いた話題がある。　新機能 M. (Memorize) を用いれば、メモリーオーバーになるかも知れぬ問題が、瞬時に解ける例があるとの事だ。
西川の例は「整数の分割 (Partition)が幾通りあるか？」であった。
例えば、 p(1) = 1, p(2) = 2, p(5) = 7,　p(10) = 42,　p(100) = 190569292,
p(200) ＝ 3972999029388 であった。
この際、計算をカナダ産の有名な数学ソフト　Maple V （学生版）でのトライでは、p(200) 相当の　numbpart(200) で　STACK OVERFLOW エラー　を発するので、
Ｊ言語の方がベター云々の記述が見えた。
しかし、中野のトライでは、格別悲観的でも無い。
多少遡って、例えば numbpart(150) 辺りから、再出発して、再計算すれば、何とか突破出来るのだ。　その繰り返しで、遥かに大数の、なんと　numbpart(1550)　が　５０桁の数　66483129659256568162714006797726637797731　通りと計算出来た。
この件を中野は西川宛に ＦＡＸ した。　（文献 2）
だがしかし、これが「末尾まで正解」であると思う根拠は何処にあろうか？

ところで、同様に大数になるであろう「順列・組合せ」の個数ならば、どうか？
１０ケの順列の個数 ！１０ ＝ 3628800、 ！１１ ＝　39916800　等は、論理が単純だから、計算機を信用するのに不安は無いだけの話であるまいか？
それでも、「順列・組合」のリスト を表示せよとなれば、至難の技である。
有名な数学ソフト Mathematica や Maple でも難儀らしく、解説や例題も乏しい。
しかし、リストの末尾の１0 ケほどをを表示せよ程度ならば、出来てもよかろうに。
前記の有名数学ソフトでさえ、そもそも、そんな機能は避けているらしい。

本題の「分割 Partition 」のリストともなったら、さらに難しかろう。
しかし、今は取りあえず、最も簡単な「幾通りか？」に重点を置こう。

実は、西川は取り上げていなかったが、 J602 beta （文献３）では、　M. 関数を Combination 関数　comb　に適用しようとして、実は何もしていない、奇妙に中途半端な例題があるのだ。　しかし、西川は、「M. 関数の効き目」の例題を、「分割」問題

で示しているので、言及したい。（文献４）

１．　会友と「分割」問題へのタッチ

　暦の順では無くて、判り易さの順に選んで見よう。　（敬称略）

１）鈴木義一郎「数の分割（**Partition**）について」　**JAPLA '08/2/23 pp.4**
　　分割リストの表示を考える。
　　分割数は、**5** ～ **10**　　　（文献５）
２）西川利男「数の分割とフェラーズ　**Ferres** の図」**JAPLA '08/2/23 pp.3**
　　分割リスト関数　　**part n**　例は　**n =. 6**
　　分割個数関数　　　**partn n**　　例は　**n =. 6**、　結果は **11** 通り　（文献６）
３）志村メール「分割法　原始的積み上げ法」**JAPLA discussion '08.3.24 pp.2**
　　再帰 も **M.** も不使用。　**partition n (n = 5)** から開始。
　　n = 20 では、西川先生と結果は合う。　**n = 30** 位で顎が出る（高橋尚子状態）。
　　n = 100 では悲惨な結果も。　とにかく、計算結果をストアして、再利用しながら
　　先へ進んでいます　（文献７、中野註：これが重要、或種の **M.** 法であろうか？）

２．　西川著多数中の有効な論文

　　上記、志村メールで「西川先生と結果は合う」は、具体的には、どの論文を意味するのか？　はっきりしないが、抜群に有効な論文があった事は事実である。
それは、かなり先の１月（**JAPLA** 研究会資料 **2008/1/26**）のもの
「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」だった。　（文献**1,**）

　　本稿の「０．　はしがき」で、中野が大いに論じたのもこれであったが、大変有効な論文であって、その **p. 3** のプログラムによる関数　**pnn** で殆どの答は出て仕舞うのだ。
その　**p.5** には、巨大数 **100** や **200** の分解例　**pnn 100** や　**pnn 200**　があった。
　有名な数学ソフト **Maple V**　との比較の問題について中野は先報（文献０）でコメントしたのであった。
つまり、 **Stack Overflow** エラー　になる数　ｎ より少し小さな値まで遡って、計算を再試行すれば、容易に避けられる。　中野は **n = 1550** まではチェック済みであった。

　　しかし、それでも大きな問題が残る。　それは、計算機が　**Stack Overflow** エラーを起こさず、志村メールに云う「高橋尚子状態」も起こさず、無事に演算を出力したとしても、それが、「数　ｎ」の分割数として、正しい保証は何処にあるか？！
の問題である。

　１）例えば啓蒙書：カルヴィン・Ｃ・クロースン「数学の不思議」（文献８）
　　ｐ．２９８　表１４　**1 - 50** の分割表
　　　　　　　　註：ミスー＞正　数 **25 1,958**、　数 **51 239,943**
　２）**Eric W. Weisstein: CRC Concise Encyclopedia of MATHEMATICS,**
　　　　　　　　　Chapman & Hall/CRC 2002 pp.3242　（文献９）
　　p. 2157　**Sloane** の表より　**n = 10** で **42, n = 50** で　**204226**
　　　　　　　n = 100 で　　**190569292**
　　　　　　　n = 150 で　　**40853235313**
　　　　　　　n = 200 で　**3972999029388 ,**
　　　　　　　n = 250 で　**2.307 x 10^14** 詳しくは　**230793554364681**
　　　　　　　n = 300 で　**9.253 x 10^15** 詳しくは　**9253082936723602**
　３）最も簡単なのは、有名な数学ソフト　Ｍａｐｌｅ Ｖ　の結果と比較する事だ。

４）西川関数　**pnn** の結果との比較でも良い。　（ **Maple V** との一致が良いので）
５）漸化式の利用：　これでは、結局全部を計算するが、要所々々でチェックすれば良かろう。
６）ラマヌジャンの　**mod** 関係：　**P(n) = nr** の時
　ａ）　**n = 5m + 4** ならば　**nr = 0 (mod 5)**　即ち　**5** の倍数
　ｂ）　**n = 7m + 5** ならば　**nr = 0 (mod 7)**　即ち　**7** の倍数
　ｃ）　**n = 11m + 6** ならば　**nr = 0 (mod 11)** 即ち　**11** の倍数
この関係式を利用すれば、検算は簡単である。　（文献　８、１０－ａ）

３．　漸化式　の　利用

啓蒙書：クロースン著「数学の不思議」（文献８）　**p.300** に、漸化式を利用した分割数の計算法がある。　西川論文（文献 **1**）にもインドの天才・ラマヌジャンの仕事として紹介されている。　志村メールでは、敢えてこの手法を避けている。その理由は次の如し。　西川曰く「この計算は大変な忍耐力を要する。　途中で一ケ所でも間違えたら、それから先は、すべて水の泡である。」と。
しかし逆に、要所々々でチェックすれば、それ以前の計算は、まず、全て正しいとも言えるので、検算としては、一挙・簡単である。　中野はこの方法を採用する。逆に、計算ミスが見付かれば、そこで修正して、更に前進すれば良い。

漸化式の原理は、すでに多くの教科書類に紹介されているので省く。

Euler 大先生の母関数

Ｐ（ｎ）＝　**Σ (-1)^(k+1)［P(n - k(3k - 1)/2) + P(n-k(3k+1)/2)］**
　ただし　**Σ** は　**k = 1** から　**n**　まで
による中野のＪプログラムのみを掲げる　（Ｊ６０２版）。

```
   PN=: 3 : 0
   n=. y
 x:  pnn=. 0
       nk0s=.0
         k=.1
   while. k < 20 do.
NB. wr  'k = ', '': k
     k30=. k*((3*k)-1)%2
       nk=. n - k30
 if. nk < 0 do. nk0s=.nk0s+1   NB.  pnk1 =. 0
                              goto_e. end.
  x: pnk0=. nk { pns
     k31=. k*((3*k)+1)%2
     nk=. n - k31
   if. nk < 0 do.   pnk1 =. 0
                    nk0s=.nk0s+1
                    goto_0. end.
   x: pnk1 =. nk { pns
   label_0.
       x:   pnn=. pnn +  ((_1)^k)* (pnk0+pnk1)
NB. if. nk  < 0 do.   goto_e. end.
     if. nk0s > 2 do. goto_e. end.
     k=. k+1
```

```
    end.
label_e.
  x: -pnn
   )
```

４． 演 算 時 間 の 例

数値は中野のマシンにて：

１） 志村プログラム（文献７）で：

partition 20 -> 627 time 0.035 sec
partition 30 -> 5604 time 0.83 sec
partition 40 -> 37338 time 12.79 sec
partition 45 -> 89134 time 38.14 sec かなり遅いな

１－ａ） 志村第２プログラム（文献７－ａ）

tc は **timer** 関数と同じで：

tc 'a45 =. partition_nr 45' -> 0.14 sec 迅速化された
配列 **$ a45 -> 45 2** 先の **45** は **n** 値 **1** ～ **45**
後の **2** は 対応する 分割数値
1 ～ **89134**
tc 'a100 =. partition_nr 100' -> 1 sec
tc 'a200 =. partition_nr 200' -> 8.7 sec
tc 'a294 =. partition_nr 294' -> 27 sec
配列の最後 **x: {: a294 -> 294 6039763882095516**
比較 **Maple V numbpart(294) -> " 515**

tc 'a300 =. partition_nr 300' -> 28.7 sec
配列の最後 **x: {: a300 -> 300 9253082936723604**
比較 **Maple V numbpart(300) -> " 602**

２） 西川第２プログラム （文献１） で： **pnn** は西川関数 として

timer ' pnn 45 ' -> 8.35e_5 sec かなり迅速！
timer 'wr pns294 =. pnn 294' -> 2.375 sec
x: pns294 -> 6039763882095514

３） 中野プログラム **PN** （本稿 ３．節）で： **PN** は中野関数 として

timer ' PN 45 ' -> 0.0004 sec 即ち **4e_4 sec** 適当に迅速
timer 'wr pn45=. PN 45' -> 時間 **0.0085 sec** と 結果 **89134**

timer 'wr pna294 =. x: PN 294' -> 0.01 sec 6039763882095516

４） 中野： 演算結果を配列 **pns** に格納して置き、それから呼び出す方法では

timer ' 45 { pns ' -> 3.7e_5 sec 圧倒的に迅速
timer ' wr x: pna294 =. 294 { pns ' -> 0.004 sec 6039763882095516

５）　上記の計算は、互いに末尾が異なる（４ と ６）。　皆、誤解らしい？

６）　山下の最近のＦＡＸ：　同様な「漸化式法」の例と意見が来信した（文献１０、１０－ａ）。　しかし、上記の例に比べて、演算に、かなり長時間を要しているのは不思議である。

５．　分割表　ｐｎｓ　の　性質

１）　結果が　５　で整除されるもの（末尾が ０ か ５）を示す。
前節の　５）の項の話題である。　この表の最後の行の情報から、我らの計算は、
Ｐ（ｎ）、　ｎ ＜ ２９４　が、残念ながら、限度だと思われる。
これをもって、「結び」の言とする事も出来よう。

```
load 'c:\documents and settings\yoshihiro\j602-user\temp\partyn.ijs'
$ pns  -> 314

n =. 4 + 5 * k =. i. 60     の場合：

pns60=. (4+ 5*k=.i.60){ pns

x: 15 4 $ pns60
```

5	30	135	490
1575	4565	12310	31185
75175	173525	386155	831820
1741630	3554345	7089500	13848650
26543660	49995925	92669720	169229875
304801365	541946240	952050665	1653668665
2841940500	4835271870	8149040695	13610949895
22540654445	37027355200	60356673280	97662728555
156919475295	250438925115	397125074750	625846753120
980462880430	1527273599625	2366022741845	3646072432125
5590088317495	8528581302375	12950095925895	19573856161145
29454549941750	44132934884255	65851585970275	97862933703585
144867692496445	213636919820625	313891991306665	459545750448675
670448123060170	974834369944625	1412749565173450	2040825852575075
2938929793929555	4219388528587095	6039763882095516	8620496275465026

第１行は　**k = 0 1 2 3** 即ち　**n = 4 9 14 19**　に対応する　分割数
終行は　**k = 56 57 58 59** 即ち **n = 284 289 294 299** に対応する　分割数
剰余　　**5 | p(n)　->　　　0　0　1　1**
最後の２例は　不都合！　この辺が限界！

2）　計算結果が　7　で整除される例：

```
   pn7=.(5+7*i.45){ pns
   $ pn7
45

  x: 11 4 $ pn7
               7              77             490            2436
           10143           37338          124754          386155
         1121505         3087735         8118264        20506255
        49995925       118114304       271248950       607163746
      1327710076      2841940500      5964539504     12292341831
     24908858009     49686288421     97662728555    189334822579
    362326859895    684957390936   1280011042268   2366022741845
   4328363658647   7840656226137  14070545699287  25025873760111
  44132934884255  77195892663512 133978259344888 230793554364681
 394723676655357 670448123060170 1131238503938606 1896564103591584
3160137867148997 5234371069753672 8620496275465026 14118662665280004
   7 | x: 11 4 $ pn7
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 1 6

  x: (299 306) { pns
8620496275465026 14118662665280004
  7 | x: (299 306) { pns
1 6
```

最後の２例はまずい。

３）　計算結果が　１１　で整除される場合：

```
    pn11=.(6+11*i.29) { pns
    $ pn11
29
  x:  pn114=.7 4 $ pn11
               11            297           3718           31185
           204226        1121505        5392783        23338469
         92669720      342325709     1188908248      3913864295
      12292341831    37027355200   107438159466    301384802048
     819876908323  2168627105469  5590088317495  14070545699287
   34643126322519  83561103925871 197726516681672  459545750448675
1050197489931117 2362219145337711 5234371069753672 11435542077822104
     11 | x: pn114
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0

   _5 {. (6+ 11* i.29)
270 281 292 303 314
   270 281 292 303 314 { pns
1.0502e15 2.36222e15 5.23437e15 1.14355e16 2.46501e16
  x:  270 281 292 303  { pns
1050197489931117 2362219145337711 5234371069753672 11435542077822104
    11 | x:  270 281 292 303  { pns
0 0 0 0
```

此処までは良かったが、 **n -= 314** では、整除されなかった。

```
    x: 314 { pns
24650106150830544
    11 | x: 314 { pns
10
```

この辺が、我らの計算の限界であろうか？

６．　む　す　び

整数の分割数の計算を、いろいろ調べた。
J　言語利用では、西川の第２プログラムが秀逸であった。

数学ソフト　**Maple V** 並に信頼出来る。
漸化式法と併用すれば有効であろう。
　しかし、 **J**言語による我らの目下の方法では、数　３００　の
分解近くでは、信頼が薄れる。　限界であろう？
　実は、世の中には、もっと優秀な方法もあるらしいので、頑張らなくちゃ！
（文献　１１）

文　　献

０）中野嘉弘：「組合せ（**Combination**）計算の話題」 **JAPLA 2008/2/23 pp.9**
１）西川利男：「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」**JAPLA 2008/1/26 pp.7**
－ａ）「Ｊ６０２の新しい機能　**M.**　について」**JAPLA**シンポジウム資料 **2007/12/8 pp.1**
２）中野ＦＡＸ：「**Maple V numbpart(1550)** までのテスト」**pp.5 2008/1/30**
３）**J602-beta** タスクバー　**Help Vocabulary M. Memo**関数　**timer fib pn**
関数　**comb** と　例.　**3 comb 5** のリスト
４）西川ＦＡＸ：「分割数、 **Maple V numbpart(100)**、 **M.** の効果？
５）鈴木義一郎：「数の分割（**Partition**）について」 **JAPLA 2008/2/23 pp.4**
６）西川利男：「数の分割とフェラーズ　**Ferres** の図」**JAPLA '08/2/23 pp.3**
７）志村正人：メール「分割法　原始的積み上げ法」**JAPLA discussion '08.3.24**
－ａ）志村：メール「**partition_nr** 数だけ数える法　**500** で　**90 sec**、
200 まで　西川、**N.Thomson** と同じ解」 **JAPLA discussion '08.3.28**
８）**Calvin C. Clawson** 原著・好田順治　訳：「数学の不思議」　青土社　**1998.3 pp.412**
９）**Eric W. Weisstein: CRC Concise Encyclopedia of MATHEMATICS, Chapman & Hall/CRC 2002 pp.3242**

１０）山下紀幸　ＦＡＸ　**'08.4.18.10:10**
「恵羅・土屋『組合せ論』漸化式では、所要時間：**400 MHz** 機で、
p(25) 4 min、 **p(50) 162 min**、 **p(100) 64.3 hours**　＝　手計算　**2 hours**」
（中野註：　何故､長時間を要したか？　不思議である。）　－
ａ）山下　ＦＡＸ　**'08.4.19.12:00**
「**p(294)** の値は、ラマジュアンの条件　**p(5m+4) = 0 mod 5**　により、末尾は
0 または **5** の筈である。　それを満たさないので、いずれの答も　アウト！
と考える。」　なお、**p(294)** の値の手計算は無理だと思うので辞退する。
ｂ）山下　ＦＡＸ　**'08.4.21.13:00**　私の手計算の限界
p(114) = 952050665　９桁　（中野　註：正解　**=5*193*986581**）
今後は、指数公式（近似）にて！

11)　例えば　**Partition Number (n = 0 to 1000) with Factorization results**

http://www.asahi-net.or.jp/~KC2H-MSM/mathland/matha1/matha145.htm