

映画「博士の愛した数式」より

Masato Shimura

2006年1月30日

目次

1	約数	2
2	完全数	3
3	友愛数	4
4	Script	6

小川洋子著の「博士が愛した数式」が映画になり、寺尾聡 深尾絵里 浅丘ルリ子 吉岡秀隆等のキャストで封切られた。1976年頃の交通事故による記憶障害で、事故後は記憶が80分しか持たない数学者と家政婦、 $\sqrt{\quad}$ と呼ばれる10才の家政婦の男の子が織りなすドラマで、愛したオイラーの数式の方は映画を見ていただくとして、この中に、友愛数、完全数が出てきた。

「28は完全数、まだ30個ほどしか発見されていない。最小は6だ。そして偉大な江夏の背番号。」

⇒ 阪神ファンにはたまらない。

「君の誕生日は」

「2月20日です。」

「僕の学長賞の時計は284番。のんきに皿洗いをしている場合ではない。良いかねこれは友愛数だ。約数を足し合わせると・・・」

1 約数

Jには素数や因数分解を求めたり n 進数を求める強力なプリミティブ (p: q: #:) 等の関数があるがすぐに約数を求める方法が見いだせなかった。Jは何通りものアルゴリズム記述の方法があり、柔軟な思考が出来る。ここでは、多少力業に頼った。

初めての人にも分かるように丁寧に約数を求める結果を追ってみよう。メモリ節約のための機能も織り込んである。

整数論は数学では綺麗だが、もの凄いマシンパワーが必要な領域である。何時間、何日、コンテストものでは数カ月以上大量のメモリを搭載したハイパワーマシンがうなりを上げてポロツと答えを出してくれる。普通のコンピュータは倍精度 (2 進数 16 桁) で構成されているが、限界を超える場合には工夫が必要となる。

```
yaku0 28
14 7 4 2 1
```

```
yaku0 12 NB. スペースの関係で 12 を例示する
6 4 3 2 1
```

```
i.12 NB. i. は{0 1..12}の順序数を生成する
0 1 2 3 4 5 6 7 8 9 10 11
>:i.12 NB. >: は 1 を加える。 C の 1++
1 2 3 4 5 6 7 8 9 10 11 12
```

```
}.>:i.12 NB. 約数で自身は不要なため除く
2 3 4 5 6 7 8 9 10 11 12
```

```
12 % }.>:i.12 NB. 12 を上記で割る。% は割り算でパーセントではない
6 4 3 2.4 2 1.71429 1.5 1.33333 1.2 1.09091 1
```

ここから整数=約数を取り出す。プリミティブ x : *Extended precision* を用いる。(|: はスペース節約のため横に寝かせたもの)

```
(1 x: 12%}.>:i.12) NB. 1 x: 分数表記
6 4 3 12r5 2 12r7 3r2 4r3 6r5 12r11 1
```

```
|: (2 x: 12%}.>:i.12) NB. 2 x: 分数の分かち書き 整数は 1 が入る
6 4 3 12 2 12 3 4 6 12 1
1 1 1 5 1 7 2 3 5 11 1
```

ここから下段 (縦表記で右の列) が 1 のものを取り出せばよい。まず、1 の判定には = や e. を用いる。ここでは e. members of を用いた。1 のところがベクトル 1 が立つ。I. でベクトルの立ったところの番号を求め{ from で取り出す方法とベクトルを指標として#で圧縮取り出しを行う方法がある。

```
({: ("1) 2 x: 12%}.>:i.12) e. 1 NB. e.1 で 最後の行の 1 をスキャン
1 1 1 0 1 0 0 0 0 0 1
```

```
((I. ({: ("1) tmp)e. 1){ tmp=. 2 x: 12%}.>:i.12)
NB. I. で no を求め
0 1 2 4 10
```

```
{}. "1 ((I. ({: ("1) tmp)e. 1){ tmp=. 2 x: 12%}.>:i.12)
NB. { で取り出し
6 1
4 1
3 1
2 1
1 1
```

```
{}. "1 ((I. ({: ("1) tmp)e. 1){ tmp=. 2 x: 12%}.>:i.12)
6 4 3 2 1 NB. 左の行だけとする
```

2 完全数

自分自身を除く約数の和が自分自身と等しくなる数。

```
yaku0 12
```

6 4 3 2 1

yaku0 28

14 7 4 2 1

yaku0 496

248 124 62 31 16 8 4 2 1

+/ yaku0 496

496

3 友愛数

自分自身を除く約数の和が相互にパートナーとなる数。550 組発見されているようだ。発見された数は奇数同士か偶数同士であるがその先は不明である。

1. 2 組のカード (表が白、裏が青のカードと、表が黄色、裏が赤など 4 色の色の付いたカード) を用意する。
2. 1000 枚 × 2 組 用意したなら、2 組とも表に 1 から 1000 までの数を書き込み、裏にその数の約数を書き込む
3. 一組 (黄色が表、裏が赤) のカードを裏返す (全て白と赤になっている)
4. 色に関係なく裏表が同じ数字のカードを捜す。

これが完全数と友愛数を見付けるアルゴリズムの一つである。他にも様々な方法があるがどれが高速か試してほしい。

先の yaku0 を用いて 30 までの約数を求めてみよう。

{@>は一個づつ box に入れてそれぞれ計算するテクニックでループが不要となる。+/
は Σ で約数を合計している。

ここで下段が 1 となっているのは素数なので外す。

```
|: a=. ( (>:i.30) ,. ; +/ L:0 yaku0 L:0 {@> >:i.30)
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30  
0 1 1 3 1 6 1 7 4 8 1 16 1 10 9 15 1 21 1 22 11 14 1 36 6 16 13 28 1 42
```

```

({:"1 a) e. 1
0 1 1 0 1 0 1 0 0 0 1 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0

```

```

(-. ({:"1 a) e. 1) # a
1 0 0 1 0 1 0 1 1 1 0 1 0 1 1 1 0 1 0 1 1 1 0 1 1 1 1 1 0 1
(-. ({:"1 a) e. 1) # a

```

```

a1=. |: (-. ({:"1 a) e. 1) # a
1 4 6 8 9 10 12 14 15 16 18 20 21 22 24 25 26 27 28 30
0 3 6 7 4 8 16 10 9 15 21 22 11 14 36 6 16 13 28 42

```

一組をひっくり返して連結し、ソートする。

```

|: a2=. /:~ a1,|."1 a1
0 1 3 4 4 6 6 6 7 8 8 9 9 10 10 11 12 13 14 14 15 15 16 16 16 18 20 21 21
1 0 4 3 9 6 6 25 8 7 10 4 15 8 14 21 16 27 10 22 9 16 12 15 26 21 22 11 18

```

同じ組み合わせを捜す。上段を指標として同じく見合わせに区切る。プリミティブ <;.1 cut の機能を用いる。そして1段のみのものを落とす

```

a3=. ((~: {"1 a2)<;.1 a2)
(-. (; # L:0 a3) e. 1) # a3
+---+---+---+---+---+---+---+---+---+---+---+---+
|4 3|6 6|8 7|9 4|10 8|14 10|15 9|16 12|21 11|22 14|28 28| | |
|4 9|6 6|8 10|9 15|10 14|14 22|15 16|16 15|21 18|22 20|28 28|
| | |6 25| | | | | | |16 26| | | |
+---+---+---+---+---+---+---+---+---+---+---+---+

```

~: nub siebe は同じものを探し指標 0 を打つ。この指標を反転させて圧縮して取り出す。

```

~: L:0 a3
+---+---+---+---+---+---+---+---+---+---+---+---+
|1 1|1 0 1|1 1|1 1|1 1|1 1|1 1|1 1|1 1|1 1|1 1|1 1|1 1|1 1|1 0|

```

```
+---+-----+---+---+---+---+---+---+---+---+---+---
```

```
(-. L:0 ~: L:0 a3) # L:0 a3
```

```
+---+---+---+---+---+---+---+---+---+---+---+---
```

```
| |6 6| | | | | | | | |28 28|
```

```
+---+---+---+---+---+---+---+---+---+---+---+---
```

```
; (-. L:0 ~: L:0 a3) # L:0 a3 NB. ; raze is open box
```

```
6 6
```

```
28 28
```

完全数と友愛数が得られる。

```
yaku3 500
```

```
6 6
```

```
28 28
```

```
220 284
```

```
284 220
```

```
496 496
```

4 Script

```
NB. Masato Shimura 29/JAN/2006
```

```
NB. friendly number
```

```
NB. Usage: yaku0 28
```

```
NB. usage: yaku3 1000;1000
```

```
yaku0=: 3 : '({."1 tmp) { ~ I. 1 = {:(("1) tmp=. 2 x: y. % }.>:i.y.'
```

```
NB. Usage: yaku0 28
```

```

yaku_sub0=: 3 : 0
select. # y.
case. 1 do. TMP0=. {@> i. ; y.
case. 2 do. TMP0=. ({. y.) + L:0 {@> i. ;{: y.
end.
TMP0
)

```

```

yaku1=: 3 : 0
NB. Usage: yaku1 100;100
(; TMP1),. ;+/ L:0 yaku0 L:0 TMP1=. yaku_sub0 y.
)

```

```

yaku2=: 3 : 0
NB. Usage: yaku2 100;100
TMP2=(; TMP1),. ;+/ L:0 yaku0 L:0 TMP1=: yaku_sub0 y.
TMP3=. /:~ ( TMP2,.0), (|. "1 TMP2),.1
TMP4=. (~: {. "1 TMP3) <;.1 TMP3
)

```

```

yaku3=: 3 : 0
NB. Usage: 1000;1000 //a + i.b
TMP2=(; TMP1),. ;+/ L:0 yaku0 L:0 TMP1=. yaku_sub0 y.
TMP20=. (I. -. (1{"1 TMP2) e. 0 1) { TMP2
TMP3=. /:~ TMP20, |."1 TMP20
TMP4=. (~: {. "1 TMP3) <;.1 TMP3
TMP6=. ~: (L:0) 2{"1 L:0 TMP4
; (0 = L:0 TMP6) # L:0 TMP4
)

```