

友愛数の計算法へのコメント (Beiler 本の寸見から)

中野嘉弘 (86 歳、札幌市南区)
Yoshihiro NAKANO (Sapporo, JAPAN)

2009 年 2 月 25 日

目次

1	どの位速いか?	1
2	Beiler 本の第一問	2
3	与数 60 では?	2
4	関数の繰り返し	3
5	友愛数の演算実例	4
6	まとめ	5
7	文献	6
8	中野関数	6

*1

はしがき

山下会員 と 西川会長の FAX 通信 を 拝見して。

我が JAPLA の活発な山下紀幸会員と敬愛する西川利男会長から、「友愛数 Amicable Numbers」について、新年来、多量の計算情報を頂いた (文献 1)。

関連する主要なる参考書として有名な A.H.Beiler 著 : Recreations in the Theory of Numbers (1966, 文献 2) を下積み、紛失中のところ、最近やっと発見出来たので、我が老友間で、話が通い始めた。

その後の数日間に気付いた若干をメモとして、お伝えして置こう。云わずもがなかも知れぬが？

*1 FAX:011-588-3354

Yoshihiro@river.ocn.ne.jp

1 どの位速いか?

山下 FAX 中に、Tacit な関数の迅速さについて、しきりに感嘆があった。それは、どの位のものか? 自分でチェックしてみた。

中野マシンは

Sony ノート PC VAIO (VGN-BX6AAPS 2008.6) XP Professional 正規版

CPU インテル(R)Core(TM)2 Duo プロセッサ T7100 1.80GHz

メモリー 4GB(2GBx2) デュアルチャンネル転送

GPU インテルグラフィクスメディアアクセラレーター X3100

HDD 160 GB

J 言語 J602 版

山下例 (文献 1-e)

500 MHz 機で、旧法で 14316 の結果が 5 分、

50 MHz 機で、Tacit な新法 amicable 14316 の結果が 40 秒、 中野例 上記機種で、時間(sec)

と 使用メモリ (Byte?) が、印刷を含め、ts 'wr amicable 14316x' から、
19116

19116 0.0151391 (sec), 5056 (Byte?)

印刷を除いた計量では、ts 'amicable 14316x' から

0.000196673 (sec), 4864 (Byte?)

使用関数 "amicable" は、西川 FAX(文献 1-a) よりのももの。

この比較値は、皆様が、納得出来る数値でしょうか?

以下、Beiler の本を再発見してからの、数日間の、慌ただししい知見を述べる。

2 Beiler 本の第一問

冒頭、p.1, Chapter 1 の第 1 問を、兎に角、解いてみよう。

1. Find the divisors, if any, of 16000001. である。

(解) 与数の因数は q: 16000001 より、109, 229, 641 の 3 ケ。 計算は、連乗積の外積法 (中野)

/: ~ ~. , (/: ~ ~. , (1 109) */ (1 229)) */ (1 641) から
1 109 229 641 24961 69869 146789 16000001 の 8 ケ。

時間・スペースは

ts '/: ~ ~. , (/: ~ ~. , (1 109) */ (1 229)) */ (1 641)' から
4.21841e_5 (sec), 3520 (byte)

(比較) J Phrase の関数 divisors (文献 1-a, 西川 FAX) で 数値は同じ、時間・スペースは

ts 'divisors 16000001x'

0.000125714(sec), 4352 (byte)

ほぼ、同等の成績であるが、「中野法」が多少は、ベターかも？これを、プログラム化することは、簡単である。

実は、この連乗積法は、Beiler 本の次章 (Chapter 2, p. 7) で登場する。当然の手法である。

3 与数 60 では?

Beiler 本の p.8 に登場する例だ。 q: 60 は $2^2 * 3 * 5$ の如く、複数乗因子がある場合である。対策としては、前節の連乗積の外積法を (単数乗で) 因子の重複数だけ、繰り返せばよからう。

```
]qs=./: ~ ~. ,(/: ~ ~. ,(1 2) */ (1 3)) */ (1 5)
1 2 3 5 6 10 15 30
```

```
]qd=./: ~ ~. ,(/: ~ ~. ,(qs) */ (1 2))
1 2 3 4 5 6 10 12 15 20 30 60
# qd
12
```

この場合にも、プログラム化は簡単である。

```
(比較) divisors 60
1 2 3 4 5 6 10 12 15 20 30 60 (12 ケ)
```

4 関数の繰り返し

上の結果を追跡して見る。最初は、因子 3 と 5 から

```
]qs=./: ~ ~. ,(/: ~ ~. ,(1 3) */ (1 5))
1 3 5 15
```

次の因子 2 で

```
]qs21=./: ~ ~. ,(/: ~ ~. ,(qs) */ (1 2))
1 2 3 5 6 10 15 30
```

さらに、因子 2 で

```
]qs22=./: ~ ~. ,(/: ~ ~. ,(qs21) */ (1 2))
1 2 3 4 5 6 10 12 15 20 30 60 (12 ケ)
```

(比較)

```

divisors 60
1 2 3 4 5 6 10 12 15 20 30 60

```

この divisors 関数の定義は

```

odometer=: #: i.@(*/)
divisors=: /:~@(~. */ .^"1 odometer@:>:@(#/.~))@q:

```

であって、簡潔であるが、内容は判り難い。目下、西川会長が分析中であるが.....。

多少、判り易く書き直して見よう。 中野関数 pd と qn 等を定義する (文献 3 にも示す)。

```

pd=: 3 : 0
:
X =. x
Y =. y
ps=.:~@~.@(1, X) */ (1, Y)
)

```

それを用いて、次の如く演算する。

```

3 pd 5
1 3 5 15
2 pd (3 pd 5)
1 2 3 5 6 10 15 30
2 pd (2 pd (3 pd 5))
1 2 3 4 5 6 10 12 15 20 30 60

```

この最後の 2 段の演算は、繰り返し記号を用い

```

2 pd ^:(2) (3 pd 5)
1 2 3 4 5 6 10 12 15 20 30 60

```

でも良い。

あるいは、次の如く、一挙に演算可能である。

```

2 pd ^:(i.3) (3 pd 5)
1 3 5 15 0 0 0 0 0 0 0 0
1 2 3 5 6 10 15 30 0 0 0 0
1 2 3 4 5 6 10 12 15 20 30 60

```

かくて、J Phrase 辺りに見られた関数は、この流儀で理解 出来た! ?

5 友愛数の演算実例

あと、問題として残るは、Beiler 本内の例の演算例を示すだけかも知れない。研究会の期日まで、時間の許す限りトライして見よう。

Beiler p.27, Table2, Amicable Number Pairs 中の最後の例 :

与数 $ya = . 111448537712x$, $yb = . 118853793424x$

演算

先ず、与数の素因子分解 : 関数 `qn ya` より

べき数	4	1	1	1
素因子	2	17	137	2990783

```
z1 =. 17 pd 137
z2 =. 2990783 pd z1
z3 =. 2 pd ^:(4) z2
```

`+/ }`: `z3` の結果は $118853793424 = yb$ (確かに!)

```
ts ' wr 2 pd ^:(4) z2 '
```

```
118853793423
0.0137643 (sec), 17728 (byte)
```

逆 `yb -> ya` の問題 :

	<code>qn</code>	<code>yb</code>	
4	1	1	
2	10103	735264	

```
w1 =. 10103 pd 735264x
w3 =. 2 pd ^:(4) w1
```

`+/ }`: `w3` の結果は $111448537712 = ya$ (確かに!)

```
ts ' wr 2 pd ^:(4) w1 '
```

```
111448537712
0.0166273 (sec), 11200 (byte)
```

```
比較 ts ' amicable y1'
118853793423
```

0.0176064 (sec), 9792 (byte)

逆 ts 'amicable y2'
111448537712

0.0714437(sec), 19776 (byte)

結果は、中野流と J-Phrase 流で、ほぼ同じである。

6 まとめ

友愛数の計算法の理解が深まったのは嬉しい。会友の今後の御研究に多少なりと、役立てば良いかなと思う。ただし、やったのは、結果の判っているものの検算であって、新たに友愛数を見出した訳では無いので、今後、奮闘しなければと思う。

7 文献

- 1) 山下 FAX (2009.2.10. 13:49) : 「Beiler の本の pp.28-29、
28chains の計算を JPC 版で計算。 関数 yakuze と amica,
-a) 西川 FAX('09.2.10 pm.5.29) :
「J Phrase 中から Tacit 定義関数 divisors 等々を発見、普通の関数 q: よりベターか? 友愛数の計算には有効と思う。」
-b) 西川 FAX ('09.2.11 am.9.31) :
「divisors(odometer)、組み合わせの関数例など」
-c) 山下 FAX(2009.2.11. 15:55) :
「28 chains、Basic 流、q: と divisors」
-d) 山下 FAX(2009.2.11. 18:31) :
「西川情報の Fdivisors etc. Tacit 関数は迅速、0.39sec、旧法の 1 万分の 1 に短縮出来た。」
-e) 山下 FAX(2009.2.14. 9:31) :
「Basic 風関数 amicay、amicayf、40 秒」
-f) 山下 FAX(2009.2.14. 17:26) : 「関数 choosex 追加情報」
-g) 山下 FAX(2009.2.19. 8:22) :
「Beiler p.27 12 桁データ、多倍長で、関数 amicas, soin」
-h) 山下 FAX(2009.2.21. 8:23)
: 「Beiler p.27 12 桁データ、指数計算で、関数 amicay、最終版」
-b) 西川 FAX ('09.2.21 am.9.27) :
「Divisors from J Phrases p.62-63」, 「JAPLA 2 月分の予稿: divisors をトレースする」
- 2) Albert H. Beiler, "Recreations in the Number Theory - The Queens of Mathematical Entertains", Chapter 4, "Just Between Friends", pp.26-30, 2nd ed. Dover Pub. Inc. New York (1966).

8 中野関数

NB. Nakano JAPLA Feb/2009

```
odometer=: #: i.@(*/)
```

```
divisors=: /:~@(~. */ . ^"1 odometer@:>:@(#/.~))@q:
```

```
qs=: ~.@q:
```

```
qn=: 3 : 0
```

```
nys=: # YS=. ~. qc=. q: y
```

```
Y1=. YS=/qc
```

```
qn1=. +/"1 Y1
```

```
(2,nys)$ qn1,YS
```

```
)
```

```
pd=: 4 : 0
```

```
X =. x
```

```
Y =. y
```

```
ps=: /:~@~.@, (1,X)*/(1,Y)
```

```
)
```