

6.3 友愛数・婚約数

完全数には親戚のようなものが存在する。220 の自分自身を除く約数の和は

$$1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$$

である。もちろん完全数ではない。ところが、284 についても同じことをすると

$$1 + 2 + 4 + 71 + 142 = 220$$

である。なんと、互いに完全数の性質を補完し合っているみたいではないか。このようなペアを友愛数と呼んでいる。ここでは友愛数のペアを求めてみよう。

[py script]

```
>>> def msum(n):
...     sum = 0
...     for i in range(1, n//2+1):
...         if n % i == 0:
...             sum += i
...     return sum
...
>>> def fnfind(n):
...     for n in range(1, n):
...         m = msum(n)
...         if (msum(m) == n) and (m > n):
...             print(n, m)
...
>>> fnfind(10000)
220 284
1184 1210
2620 2924
5020 5564
6232 6368
```

約数の和を計算する `msum` 関数は、完全数を求めるときに使ったものと同じである。`fnfind` 関数は、引数 `n` までにある友愛数のペアを求めるスクリプトになっている。実際、10000 未満の友愛数を求めると 5 組あった。このほとんどは、オイラーによって見つけられた。

正直に 1 から一すなわち 1 の友愛数を調べることから一始めているけれど、1 や 2 に友愛数が見つかるわけがないのもっと上の数から始めてもよかった。一旦、`n` の約数の合計を求め、それが `n` に等しいかどうかで友愛数を判定している。`if` 文のもうひとつの条件 `m > n` は、完全数や同じ組の友愛数を表示しないためにある。つまり、完全数の約数の合計はもとの数になるから一それが完全数の定義だから一広い意味で完全数も友愛数である。しかし、わざわざそれを表示することは

ないだろう。また、 n の友愛数 m を見つけたら、同時に m の友愛数は n なので、同じペアが重ねて表示されてしまう。これらを除くために $m > n$ を条件に加えている。

友愛数に似たペアに婚約数がある。完全数も友愛数も約数の合計に1は加えるが自分自身を加えない。そもそも1と自分自身はどんな数でも持っている自明の約数だから、そのうち1は加えて自分自身を加えないのはバランスを欠いているのではないだろうか。そこで、約数の合計に1を加えないとどうなるか、と考えるのは自然な流れだ。ここでは1と自分自身を含まない約数の和を、真の約数の和と表現しておこう。

これを調べるスクリプトは、`msum`関数において`in range(2, n//2+1)`とするだけである。

[py script]

```
>>> fnfind(10000)
48 75
140 195
1050 1925
1575 1648
2024 2295
5775 6128
8892 16587
9504 20735
```

10000未満の婚約数を調べてみると、友愛数とはおもむきの違うペアが現れる。友愛数のペアと婚約数のペアを見比べて気づくことはないだろうか。そう、友愛数はすべて偶数のペアで、婚約数はすべて奇数と偶数のペアになっていることだ。これは非常に不思議なことで、これまでに見つかった友愛数と婚約数の組み合わせは、すべてそうなのだ。友愛数とか婚約数の名称がどうやってついたかは知らない。ただ、古代ギリシアの数字感に、1は神、2(偶数)は女、3(奇数)は男、5(2+3)は結婚、...というものがある。それに照らすと、偶数どうしのペアが友愛数、奇数と偶数のペアが婚約数という名称は的を射ている。

さらにこの考えを進めると、 a の真の約数の和 $\rightarrow b$ 、 b の真の約数の和 $\rightarrow c$ 、 c の真の約数の和 $\rightarrow a$ のように、3数で巡回する組や4数で巡回する組など、いくらでも拡張することができる。有名なところでは12496 \rightarrow 14288 \rightarrow 15472 \rightarrow 14536 \rightarrow 14264 という5数の巡回がある。このような数は社交数と呼ばれる。社交数には5数以外の社交の輪がいくつも見つまっているが、不思議なことに3数の社交の輪は見つからないようだ。三角関係は成り立たないのだろうか。

ただし、社交数の組を見つけるスクリプトを書くのはちょっと苦勞する。というのは、ある数の真の約数を調べるのは簡単なことでも、何組でもとに戻るかが最初に分かってないので、比較を何回すればよいか未定であることだ。つまり、数 a の巡回を20回やって最初の約数の和に一致しな

いからといって、 a が社交の輪にないとは言い切れないのである。実際に、28 数で巡回するものがあるというから一筋縄ではいかない。まあ、100 数以内で見つからなければあきらめる、というように決めていれば、スクリプトは難しくはない。for 構文ひとつでこなせるはずだから、君たち自身でがんばってみよう。