

① 予備知識

自然数 n を用いて $2^n - 1$ の形に表される数を Mersenne 数と呼びます。Mersenne 数 $2^n - 1$ が素数のとき、自然数 $N = 2^{n-1}(2^n - 1)$ は完全数になります。実際、自然数 N の約数は、素数 $2^n - 1$ を含むもの

$$2^i (2^n - 1), \quad i = 0, 1, 2, \dots, n - 1 \quad [1]$$

と、含まないもの

$$2^i, \quad i = 0, 1, 2, \dots, n - 1 \quad [2]$$

に分類でき、これらの総和は、

$$\sum_{i=0}^{n-1} 2^i (2^n - 1) + \sum_{i=0}^{n-1} 2^i = 2^n \sum_{i=0}^{n-1} 2^i = 2^n \times \frac{1 - 2^n}{1 - 2} = 2^n (2^n - 1) = 2N \quad [3]$$

となります。逆に、偶数の完全数はいずれも $2^{n-1}(2^n - 1)$ の形で表されることが知られています。

一般に巨大な自然数を与えたとき、それが素数か合成数かを厳密に判定することはきわめて困難ですが、Mersenne 数に対しては、「Lucas テスト」と呼ばれる簡単な判定法が知られています。

そこで、完全数を見つけようとする場合などに Mersenne 数が利用されます。

② Lucas テスト

数列 $\{L_n\}$ を、漸化式

$$L_1 = 4, \quad L_{n+1} = L_n^2 - 2 \quad (n \geq 1) \quad [4]$$

で定義すると、3 以上の自然数 n に対して、次の (1) と (2) が同値であることが知られています。

- (1) Mersenne 数 $M_n = 2^n - 1$ は素数である。
- (2) 自然数 L_{n-1} は Mersenne 数 $M_n = 2^n - 1$ で割り切れる。

つまり、Mersenne 数 M_n が素数か否かを判定するには、 L_{n-1} が M_n で割り切れるかどうかを知らねばよいこととなります。この判定法が Lucas テストです。

実際に、Maxima を使って、Lucas テストを実行してみましょう。まず、数列 $\{L_n\}$ を定義します。

```
(%i1) luc(n) := block([L: 4], for i: 1 thru n - 1 do L: L^2 - 2, L);
                                2
(%o1)      luc(n) := block([L : 4], for i thru n - 1 do L : L - 2, L)
```

これを用いて、いくつかの Mersenne 数を判定してみると、次のようになります。

```
(%i4) remainder(luc(2), 2^3 - 1);
(%o4)                                0
(%i5) remainder(luc(3), 2^4 - 1);
(%o5)                                14
(%i6) remainder(luc(4), 2^5 - 1);
(%o6)                                0
(%i7) remainder(luc(5), 2^6 - 1);
(%o7)                                23
```

この結果から、 $2^3 - 1 = 7$ と $2^5 - 1 = 31$ は素数であり、 $2^4 - 1 = 15$ と $2^6 - 1 = 63$ は合成数と分かりました。

③ 改良

前節の数列 $\{L_n\}$ の一般項は次のようになります。

$$L_n = (2 + \sqrt{3})^{2^{n-1}} + (2 - \sqrt{3})^{2^{n-1}} \quad [5]$$

指数 2^{n-1} が指数関数であることから、数列 $\{L_n\}$ は急激に大きくなることが分かります。従って、前節のように「 L_{n-1} を求めて、 M_n で割る」という手順では効率が悪すぎます。そこで、「 $L^2 - 1$ を L と置く」代わりに、「 $L^2 - 1$ を M_n で割った余りを L と置く」方法に変更し、Mersenne 数が素数か否かを判定する関数 `merpp` を作成します。

```
(%i8) merpp(n) := block([m: 2^n - 1, x: 4],
  for i: 1 thru n - 2 do x: remainder(x^2 - 2, m),
  if x = 0
    then return("prime")
    else return("composite")
);

(%o8) merpp(n) := block([m: 2^n - 1, x: 4],
  for i thru n - 2 do x: remainder(x^2 - 2, m),
  if x = 0 then return("prime") else return("composite"))
```

試しに、1957 年に Hans Ivar Riesel が発見した 18 番目の Mersenne 素数（素数である Mersenne 数） $2^{3217} - 1$ が素数であることを確かめてみましょう。

```
(%i9) merpp(3217);
(%o9)                                prime
```

作成日：平成 20 年 8 月 14 日～8 月 17 日
ソフトウェア：Maxima 5.15.0cvs & CMU Common Lisp Snapshot 2008-08 (19E)