

① 予備知識

自然数 N に対して、その全ての真の約数の和が N に一致するとき、 N を完全数と呼びます。例えば 6 は完全数です。実際、真の約数 $\{1, 2, 3\}$ の和は 6 になります。

② 完全数を探す

Maxima を使って、完全数を探してみたいと思います。

Maxima には、与えられた整数の全ての正の約数を求める関数 `divisors` が用意されています。

```
(%i1) divisors(6);
(%o1) {1, 2, 3, 6}
```

出力結果が「リスト」であれば関数 `apply` や `lreduce` などを用いて総和を求めることができますが、関数 `divisors` の出力結果は「集合」のため、関数 `tree_reduce` を用いて総和を求めることにします^{*1}。

```
(%i2) tree_reduce("+", %);
(%o2) 12
```

完全数か否かを判定する関数 `perfectp` を作ってみます。

```
(%i3) perfectp(n) := block(
  if tree_reduce("+", divisors(n)) = 2*n
  then true
  else false
);
(%o3) perfectp(n) := block(if tree_reduce("+", divisors(n)) = 2 n
                           then true)
```

作成した関数 `perfectp` が期待通りの動作するか確かめてみます。

```
(%i4) perfectp(6);
(%o4) true
(%i5) perfectp(7);
(%o5) false
```

6 は完全数ですが、7 は完全数ではありません ($1 + 7 \neq 2 \times 7$) ので、うまく機能しているようです。

^{*1} 関数 `args` や `listify` を用いれば、「集合」を「リスト」に変換できます。

この関数を利用して、10000 までの完全数を求めてみましょう。

```
(%i6) for i: 1 thru 10000 do if perfectp(i) then print(i);  
6  
28  
496  
8128  
(%o6) done
```

③ 未解決問題

前節の計算で見つけた完全数 6、28、496、8128 はいずれも偶数ですが、実は、奇数の完全数が存在するか否か、まだ未解決です。一方、偶数の完全数については、現在 44 個見つかっていますが、無数に存在するのかそれとも有限個しか存在しないのかは未解決です。なお、偶数の完全数 N は、 $2^p - 1$ の形をした素数 (Mersenne 素数) を用いて $N = 2^{p-1} (2^p - 1)$ と表されることが知られています。例えば、最小の完全数 6 は $2^{2-1} (2^2 - 1)$ と表せます。