Project Euler 88. Product-sum Numbers

hiragn

2024 年 12 月 27 日

1. 問題の概要

k ($\geqq 2$) 個の自然数 $a_1, a_2, \cdots, a_k$ の集合の和かつ積として表せる自然数 n を積和数と呼ぶ。

$$n = a_1 + a_2 + \cdots + a_k = a_1 \times a_2 \times \cdots \times a_k$$

たとえば $6 = 1 + 2 + 3 = 1 \times 2 \times 3$ より 6 は積和数である。
ある集合の大きさ k に対してこの性質をもつ最小の n を最小積和数と呼ぼう。
$k = 2, 3, 4, 5, 6$ に対する最小積和数は次のようになる。

- $k = 2 : 4\ (= 2 \times 2 = 2 + 2)$
- $k = 3 : 6\ (= 1 \times 2 \times 3 = 1 + 2 + 3)$
- $k = 4 : 8\ (= 1 \times 1 \times 2 \times 4 = 1 + 1 + 2 + 4)$
- $k = 5 : 8\ (= 1 \times 1 \times 2 \times 2 \times 2 = 1 + 1 + 2 + 2 + 2)$
- $k = 6 : 12\ (= 1 \times 1 \times 1 \times 1 \times 2 \times 6 = 1 + 1 + 1 + 1 + 2 + 6)$

$2 \leqq k \leqq 6$ に対するすべての最小積和数の和は $4 + 6 + 8 + 12 = 30$ である。8 は和に一回だけカウントされる。
$2 \leqq k \leqq 12$ に対する最小積和数の集合は $\{4, 6, 8, 12, 15, 16\}$ であり，その和は 61 である。$2 \leqq k \leqq 12{,}000$ に対する最小積和数の集合の和を求めよ。

https://projecteuler.net/problem=88

2.　n と k の関係

「同じ個数の自然数の積と和が一致」は一見厳しい条件に思えますが，適当に 1 を補うことで実現できます。たとえば $n = pq$（p, q は 2 以上の自然数）は次のように分解できます。

1. 積と和の差だけ $+1$ を補う

$$n = p \times q = \underbrace{p + q + \underbrace{1 + 1 + \cdots + 1}_{pq-p-q\text{ 個}}}_{pq-p-q+2\text{ 個}}$$

2. 積の方に $\times 1$ を必要なだけ補う

$$n = p \times q \times \underbrace{1 \times 1 \times \cdots \times 1}_{pq-p-q\text{ 個}} = p + q + \underbrace{1 + 1 + \cdots + 1}_{pq-p-q\text{ 個}}$$

これを一般化します。2 以上の自然数だけを使って $n = b_1 \times b_2 \times \cdots \times b_l$ $(b_i \geqq 2)$ と書けるとき，k は次のようにして求められます。

$$k = \left(n - \sum_i b_i\right) + l = n - \sum_i^l b_i + l \quad \cdots\cdots①$$

実際に計算してみましょう。$n = 2 \times 6 = 3 \times 4 = 2 \times 2 \times 3$ に対応する k を順に k_1, k_2, k_3 とします。

$$\begin{aligned} k_1 &= 12 - (2+6) + 2 = 6 \\ k_2 &= 12 - (3+4) + 2 = 7 \\ k_3 &= 12 - (2+2+3) + 3 = 8 \end{aligned}$$

3.　n の上限をおさえる

以下，k の最小積和数を $\texttt{mps}(k)$ であらわします。[*1]

これは k を与える n の最小値なので①をみたす (n, k) を 1 つ見つければ，$\texttt{mps}(k)$ を上から評価できます。

①をみたす例は簡単に作ることができて，$n = 2 \times k$ のとき $l = 2$ なので①が成立します。

$$(\text{右辺}) = 2k - (2 + k) + 2 = k = (\text{左辺})$$

$\texttt{mps}(k) \leqq 2k$ が言えました。たとえば $2 \leqq k \leqq 6$ に対する $\texttt{mps}(k)$ を求めるには $n = 12$ まで調べれば十分です。

[*1] mps は minimal product-sum number（最小積和数）の略。

4. コードを書く

これで解答の方針が立ちました。

1. $n = 4$ から $n = 2 \times k_{\max}$ までの数を 2 以上の数の積に分解する
2. 各分解に対応する k を求める
3. k ごとに最小の n を求めて和を計算

積への分解は Ordered and Unordered Factorizations of Integers という pdf が参考になりました。5 ページの `OrderedFactorizations` と `DistinctOrderedFactorizations` をもとにコードを書きました。[*2]

```
In[]:= Clear["Global`*"];
RepeatedTiming[
 kmax = 12*10^3;

 (* n を積に分解したリストを作る *)
 tbl[1] = {{}};
 tbl[n_?PrimeQ] := tbl[n] = {{n}};
 tbl[n_] := tbl[n] = Module[{lst},
    lst = Function[{d}, Append[#, d] & /@ tbl[n/d]] /@ Rest@Divisors@n;
    lst = Flatten[lst, 1];
    DeleteCases[lst, Null];
    DeleteDuplicates[Sort /@ lst]];

 (* n の分解から k を求める *)
 calc[lst_] := Times @@ lst - Total@lst + Length@lst;

 (* 最小積和数 *)
 mps[i_] := mps[i] = 2 i;
 Do[If[i < mps@#, mps@# = i] & /@ (calc /@ tbl@i), {i, 4, 2*kmax}];
 ans = Total@DeleteDuplicates[mps /@ Range[2, kmax]]]

Out[]= {1.07915, 7587457}
```

[*2] https://www.reed.edu/physics/faculty/wheeler/documents/Quantum%20Mechanics/Miscellaneous%20Essays/Partitions%20&%20Separability/Factorizations.pdf