

# 会津合宿 2018 Day 3

## B: 階層的計算機

原案：鈴木  
問題文：瀧澤  
解答：瀧澤・杉江  
解説：瀧澤

2018/03/28

# 問題概要

## 階層的計算機

$a_1, a_2, \dots, a_N$  が与えられたとき、そのうちのいくつかを選んで掛け合わせることを考えます。この積を最大たらしめるような添字のうちで、個数最小で辞書順最小のものを求めてください。ただし、一つも選ばなかった場合の積は 1 であるとしています。

### 制約

- $1 \leq N \leq 60$
- $-2 \leq a_i \leq +2$  for  $1 \leq i \leq N$

# 解法

- 値を最大にしたい以上、+2 を掛けないことはありえない。
- -2 は一つであれば掛けたくないが二つずつであれば掛けたい。
- -2 が奇数個でも -1 があれば、-2 を全て掛けることができる
  - -1 のうちで最も添字の小さいものを掛ける。

これにより、 $O(N)$  で解くことができました。

当初、この問題には個数最小の制約はありませんでした。すなわち、積を最大化するもののうち辞書順最小のものを求める問題でした。

実装が少々面倒になることが予想されたため設定を変えたのですが、簡単になりすぎたかもしれません。B 問題は難易度調整が難しいですね。

元問題の解説 PDF も書いてしまったので載せておきます。解いてもらえたら嬉しいです。

## 方針・考察（元問題）

一旦辞書順のことは忘れて、積の最大値を考える。

- 積を最大化したいので、 $+2$  は全て掛けたい。
- 単位元なので、 $+1$  は全て掛けることにしておく。
- $0$  は掛けない。
  - ひとつでも  $0$  を掛けると積は  $0$  になるが、これは何も掛けなかった場合の値より小さいため。
- $-1$  や  $-2$  は合わせて偶数個掛けたい。 $-2$  をできるだけ多く掛けたい。

## 想定解法（元問題）

$a_1, a_2, \dots, a_N$  の中に  $+2$  が  $s$  個、 $-1$  が  $t$  個、 $-2$  が  $u$  個あった場合、積の最大値は以下になる。

$$\begin{cases} 2^s \cdot 2^u & \text{if } u \bmod 2 = 0 \text{ or } t \geq 1 \\ 2^s \cdot 2^{u-1} & \text{otherwise} \end{cases}$$

これを実現するように、掛け合わせる要素を考えればよい。

- $+2$  の要素は  $s$  個全て選ぶ。
- $-2$  の要素は、上の条件に応じて先頭から  $u$  個または  $u - 1$  個選ぶ。
- $-1$  および  $+1$  の要素は次ページで述べるように辞書順を気にして選ぶ。

## 想定解法（元問題）

まず  $-1$  の要素について考える。

- $u \bmod 2 = 0$  の場合
  - 先頭から 2 個ずつ選んでいく。このときの添字を  $(i_1, i_2)$  とする。
  - $i_1$  が既に選んだ要素 ( $+2$  か  $-2$ ) の添字の最大値を超えない限り続ける。
- $u \bmod 2 = 1$  かつ  $t \geq 1$  の場合
  - まず先頭の 1 つを選ぶ。
  - その後は  $u \bmod 2 = 0$  の場合同様に 2 個ずつ選んでいく。
- $u \bmod 2 = 1$  かつ  $t = 1$  の場合
  - その 1 つを選ぶ。

## 想定解法（元問題）

次に  $+1$  の要素について考える。

先頭から 1 個ずつ選んでいく操作を、その要素の添字が既に選んだ要素 ( $+2$ 、 $-1$ 、 $-2$ ) の添字の最大値を超えない限り続ける。

以上の操作によって選ばれる添字の集合が辞書順最小となる。積が同じ値であるとき、今までに選ばれた最大の添字より大きい添字は選ぶべきでないことと、今までに選ばれた最大の添字より小さい添字は選ぶべきであるということに基づく。

この方針により、 $O(N)$  で解くことができた。



## 備考（元問題）

自明に  $2^{60} < 2^{63}$  なので、実際に要素を全て掛け合わせたとしても符号つき 64bit 整数に収まる。このことを利用した解法を生やしてもよい。

$60^4 \approx 1.3 \times 10^7$  なので、 $O(N^4)$  程度の DP を生やしてもよい。

前回の会津合宿の B 問題では、入力の上限値を大きくしたところ事故ったので、今回は上限値を小さくして多様な解法を期待する流れとなった。

## 備考（元問題）

前述の想定解法において、絶対値の上限値が2であることは利用していない。絶対値が3以上の要素があった場合でも、与えられた要素たちを「-2以下」「-1」「0」「+1」「+2以上」と分けることでほぼ同様に話を進めることができる。

与えられた要素の中に -1 が存在せず、-2 以下の要素が奇数個あった場合には、そのうちで絶対値が最小（複数ある場合はそのうち添字が最大）のものを除くことにし、それ以外のものは今回の解法と同様に選ぶ。

この方針により  $N \leq 100,000$  かつ  $|a_i| \leq 10^{18}$  でも解くことができる。

# Writer 解（元問題）

## Writer 解

- 瀧澤：54 lines, 1152 bytes in C++
- 杉江：60 lines, 1724 bytes in C++

# Writer 解と統計

## Writer 解

- 瀧澤 : 8 lines, 278 bytes in Python3
- 瀧澤 : 166 lines, 808 bytes in Whitespace
- 杉江 : 35 lines, 855 bytes in C++

## Acceptance / Submission

- 57.95 % (51 / 88)

## First Acceptance

- On-site : rupc\_OKA (00:15:13)
- On-line : tsm (00:08:34)