# CODE FESTIVAL 2016 Tournament Round2 A 問題解説

chokudai

## 部分点 1

部分点 1 において、$p = 100$ であることから、高橋君は必ず $-1$ 移動することが解る。高橋君の初期位置を $x$ とすると、

- $x$ が偶数の時、青木君は常に $+1$ 移動し続けることで、 $\frac{x}{2}$ ターンで高橋君と合流出来る。
- $x$ が奇数の時、青木君は、最初の 1 ターンだけ動かず、次のターンから $+1$ 移動し続けることで、$\frac{x+1}{2}$ ターンで高橋君と合流できる。

よって、どちらの場合も、$\frac{x+1}{2}$ を求め、余りを切り捨てることで、求めるターン数を求めることが出来る。

## 部分点 2

部分点 2 においては、高橋君がどちらに移動するか解らないため、高橋君とすれ違ってしまう可能性がある。だが、高橋君と絶対にすれ違わないように、青木君が移動する方法が存在する。

高橋君は、必ず $-1$ か $+1$ のどちらかに移動する。すると、高橋君の初期位置を $x$ とすると、

- $x$ が偶数の時、奇数ターン目には偶数座標に、偶数ターン目には奇数座標にいる。
- $x$ が奇数の時、奇数ターン目には奇数座標に、偶数ターン目には偶数座標にいる。

ということが解るため、高橋君の座標の偶奇は常に把握できる。

ここで、すれ違いについて考える。高橋君と青木君がすれ違ってしまうパターンを考えた時、高橋君と青木君の偶奇は必ず一致していない。つまり、高橋君と青木君の偶奇が一致することで、すれ違いを回避することが可能である。

これを踏まえて、青木君の戦略として、以下のような戦略が考えられる。

- $x$ が偶数の時、青木君は常に $+1$ 移動し続ける。
- $x$ が奇数の時、青木君は、最初の 1 ターンだけ動かず、 次のターンから $+1$ 移動し続ける。

このような戦略を取った時、青木君は高橋君を必ず捕まえることが出来る。また、この戦略を取ることで、ターン数を最小に出来ることは、容易に証明出来る。よって、このような戦略を取った時の期待値を計算すれば良い。

この期待値は、高橋君と青木君の距離を $x$ と置くと、以下のような漸化式が立てられる。

$$dp_x = dp_{x-2} * \frac{p}{100} + dp_x * \frac{100-p}{100}$$

これを解くと、以下の式を立てることが出来る。

$$dp_x = dp_{x-2} + \frac{100}{100-p} \tag{1}$$

この漸化式を動的計画法を用いて解くことで、$x$ が偶数の時の期待値を求めることが出来る。$x$ が奇数の時は、1 ターン待機することで、$x+1$ と $x-1$ の 2 通りに分かれるため、これらについてそれぞれ解くことで、偶数の場合と同様に求めることが出来る。

## 満点

(1) の式を考えると、$x$ が偶数の時、以下のような数式に変換することが出来る。

$$dp_x = \frac{100}{100-p} * \frac{x}{2}$$

この式は、$x$ が奇数の時についても計算すると、

$$dp_x = \frac{100}{100-p} * \frac{x-1}{2} * \frac{p}{100} + \frac{100}{100-p} * \frac{x+1}{2} * \frac{100-p}{100} = \frac{100}{100-p} * \frac{x}{2}$$

となり、$x$ が偶数の時と一致する。つまり、この式を解くだけで答えを求めることが出来る。

# CODE FESTIVAL 2016 Elimination Tournament Round 2 B 問題 解説

yutaka1999

$M = 1$ のとき、ある $i$ で $A_{(i,1)} \geq A_{(i+1,1)}$ となるならば目標を達成できず、それ以外の時は初期状態で目標を達成しているため、明らか。以下、$M \geq 2$ の時を考える。

数列 $a$ の $i$ 項目を $a_i$ で表し、数列 $a$ に対して魔法をかけて得られる数列を $f(a)$ で表すことにする。このとき、次の補題が成立する。

**補題．** $a < b \Leftrightarrow f(a) < f(b)$

まず、$a < b$ のとき、$a_j = b_j (1 \leq j < i), a_i < b_i$ なる $i$ が存在し、このとき、$f(a)_j = f(b)_j (1 \leq j < i), f(a)_i < f(b)_i$ であるから成立。逆も同様である。よって、$A_j$ に魔法を $t_j$ 回かけるとき、最終的に $A_i < A_{i+1}$ となることと、

- $t_i < t_{i+1}$ のとき $f^{t_{i+1}-t_i}(A_{i+1}) > A_i$
- $t_i \geq t_{i+1}$ のとき $f^{t_i-t_{i+1}}(A_i) < A_{i+1}$

は同値である。正の数からなる数列 $a$ では $a < f(a)$ であることを考慮すると、以下の問題を解けばよいことになる。

**問題．** 二つの正の数からなる数列 $a, b$ に対して、$b < f^t(a)$ となる最小の $t$ を求めよ。

まず、$b < a$ ならば、明らかなので、$a \leq b$ とする。ここで、第一項目を考えると、$f^t(a)_1 = a_1$ より、$b_1 > a_1$ ならば、条件を満たす $t$ は存在しない。よって、$a_1 = b_1$ の時のみを考えればよい。

ここで、次に第二項目を考えると、$f^t(a)_2 = a_2 + t \cdot a_1$ であるので、$b_2 \leq a_2 + t \cdot a_1$ を満たす必要がある。このような最小の $t$ を考えたとき、$b_2 < a_2 + t \cdot a_1$ なら $t$ を条件を満たすのでこれが最小値。よって、$b_2 = a_2 + t \cdot a_1$ のときを考えればよい。このとき、$b_2 < a_2 + (t+1)a_1$ であるから、$b < f^{t+1}(a)$ となり、最小値は $t$ または $t+1$。よって、$b < f^t(a)$ かどうかの判定をできればよい。$T = \max A_{(i,j)}$ とすると、

- $t \leq \log T$ のとき この場合は $t$ 回 $f$ の操作を行う
- $t > \log T$ のとき $f^t(a)_i$ に $a_j (j \leq i)$ が足される回数は ${}_{i-j+t}C_t$ 回であるから、$i - j > \log T$ の場合は $T$ より大きな数が足されるので、inf として処理し、そうでない場合は実際にその回数分加算する

とすることで、$t$ 回操作した後の各項の値を求めることができる（ただし、$T$ より大きいときは inf として処理する）ので、$b < f^t(a)$ かどうかの判定が可能。

以上より、$t_i - t_{i+1} \geq g(i)$ であることが必要十分というような $g(i)$ をとることができる。これに従い、$\sum t_i$ を最小化するためには、$t_0 = 0$ として、$t_i$ を $i$ の小さい方から貪欲に定めていけばよい。$g(i)$ は各 $i$ に対して、$O(M \log T)$ で求められるので、全体として、$O(NM \log T)$ で解くことができる。

# CODE FESTIVAL 2016 Tournament Round2 Problem A Editorial

chokudai

2016 年 11 月 26 日

## Partial Score 1

Since $p = 100$ in the additional constraints for the first partial score, Takahashi will always move by $-1$. Let the initial position of Takahashi be $x$, then:

- If $x$ is even, Aoki can meet Takahashi in $\frac{x}{2}$ turns by always moving by $+1$.
- If $x$ is odd, Aoki can meet Takahashi in $\frac{x+1}{2}$ turns by halting in the first turn and then always moving by $+1$.

Thus, in either case, $\frac{x+1}{2}$ rounded down to the nearest integer is the answer.

## Partial Score 2

In the constraints for the second partial score, Takahashi can move to both direction, and it is possible that the two pass each other. However, it is possible for Aoki to move so that they will never pass each other.

Takahashi will either move by $-1$ or $+1$. Let the initial position of Takahashi be $x$, then:

- If $x$ is even, Takahashi will be at an even-numbered coordinate at the beginning of the odd-number-th turns, and at an odd-numbered coordinate at the beginning of the even-number-th turns.
- If $x$ is odd, Takahashi will be at an odd-numbered coordinate at the beginning of

the odd-number-th turns, and at an even-numbered coordinate at the beginning of the even-number-th turns.

That is, the parity of the coordinate of Takahashi is always known. Now, we will consider how to avoid passing each other. When the two pass each other, the parity of their coordinate is different. Thus, it can be avoided by having the same parity.

From above, the optimal strategy for Aoki is as follows:

- If $x$ is even, always move by $+1$.
- If $x$ is odd, holt int he first turn and then always move by $+1$.

Following this strategy, Aoki will always catch Takahashi eventually. Also, it can be easily proven that the expected number of turns taken is minimized by following this strategy. Thus, we will compute the expected number of turns taken when Aoki follows this strategy.

There is a following recurring relation for the expected number of turns $dp_x$, where $x$ is the distance between the two:

$$dp_x = dp_{x-2} * \frac{p}{100} + dp_x * \frac{100-p}{100}$$

from which we obtain:

$$dp_x = dp_{x-2} + \frac{100}{100-p} \tag{1}$$

From this relation, we can perform dynamic programming to find the expected number of turns when $x$ is even. If $x$ is odd, the distance will become either $x-1$ or $x+1$ after halting for 1 turn, and then each of these cases can be solved in the same manner.

## Full Credit

When $x$ is even, the formula (1) can be transformed into:

$$dp_x = \frac{100}{100-p} * \frac{x}{2}$$

Calculating for the case where $x$ is odd, we obtain:

$$dp_x = \frac{100}{100-p} * \frac{x-1}{2} * \frac{p}{100} + \frac{100}{100-p} * \frac{x+1}{2} * \frac{100-p}{100} = \frac{100}{100-p} * \frac{x}{2}$$

which is equal to the result where $x$ is even. That is, the answer can be found directly from this formula.

# CODE FESTIVAL 2016 Elimination Tournament Round 2 Problem B Editorial

yutaka1999

When $M = 1$, if for some $i$ $A_{(i,1)} \geq A_{(i+1,1)}$ holds, the objective cannot be achieved, otherwise the objective is already achieved with the initial state. From now on, we will consider the case where $M \geq 2$.

Let $a_i$ denote the $i$-th term in $a$, and $f(a)$ denote the sequence obtained by casting the spell on $a$. Then, the following lemma holds:

**Lemma .** $a < b \Leftrightarrow f(a) < f(b)$

First, when $a < b$, there exists $i$ such that $a_j = b_j (1 \leq j < i), a_i < b_i$, and then $f(a)_j = f(b)_j (1 \leq j < i), f(a)_i < f(b)_i$ holds, thus the $a < b \Rightarrow f(a) < f(b)$ is shown. $a < b \Leftarrow f(a) < f(b)$ is also shown similarly.

Thus, $A_i < A_{i+1}$ after casting the spell $t_j$ times on $A_j$, is equivalent to the following:

- $f^{t_{i+1}-t_i}(A_{i+1}) > A_i$ when $t_i < t_{i+1}$
- $f^{t_i-t_{i+1}}(A_i) < A_{i+1}$ when $t_i \geq t_{i+1}$

Considering that $a < f(a)$ holds for a sequence $a$ consisting of positive numbers, what we have to solve is the following:

**Problem .** *For two sequences $a, b$ each consisting of positive numbers, find the minimum $t$ such that $b < f^t(a)$.*

When $b < a$, the answer is 0. From now on, we suppose that $a \leq b$. Focusing on the first terms, there exists no $t$ satisfying the condition if $b_1 > a_1$, since $f^t(a)_1 = a_1$. Thus, we only need to consider the case where $a_1 = b_1$. Focusing on the second terms, $b_2 \leq a_2 + t \cdot a_1$ must hold because $f^t(a)_2 = a_2 + t \cdot a_1$. If the minimum value of such $t$ satisfies $b_2 < a_2 + t \cdot a_1$, that value of $t$ is the answer. Now, we only have to deal with the case where $b_2 = a_2 + t \cdot a_1$. Here, since $b_2 < a_2 + (t+1)a_1$, $b < f^{t+1}(a)$ holds and the answer is either $t$ or $t+1$. Thus, we only have to determine whether $b < f^t(a)$ holds. Let $T = \max A_{(i,j)}$, then:

- When $t \leq \log T$, we actually perform $t$ iterations of the operation of $f$.
- When $t > \log T$, since the number of times $a_j (j \leq i)$ will be added to $f^t(a)_i$ is ${}_{i-j+t}C_t$, if $i - j > \log T$, value totaling over $T$ will be added, thus we treat it as inf, otherwise we actually add the value for that number of times

This way, we can find each term after $t$ casts of the spell (or detect that it is above $T$ and treat it as inf), and thus determine whether $b < f^t(a)$ holds.

Therefore, we can find $g(i)$ such that the objective is equivalent to $t_i - t_{i+1} \geq g(i)$. Accordingly, in order to minimize $\sum t_i$, let $t_0 = 0$, and greedily set $t_i$ to the minimum possible value in ascending order of $i$. Since $g(i)$ can be found in $\mathrm{O}(M \log T)$ for each $i$, the whole problem can be solved in $\mathrm{O}(NM \log T)$ time.