# AtCoder Regular Contest 153

## B問題 別解

# 自己紹介

しんちゃん @Sophia_maki

兵庫県西脇市出身

明石高専　→　大阪大学工学部

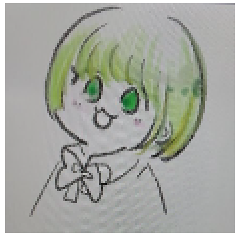
AtCoder 黄色・Codeforces 薄橙

実は明日誕生日です！！

なんでも聞いてね！

**初段**

shinchan ★



[アイコン設定]

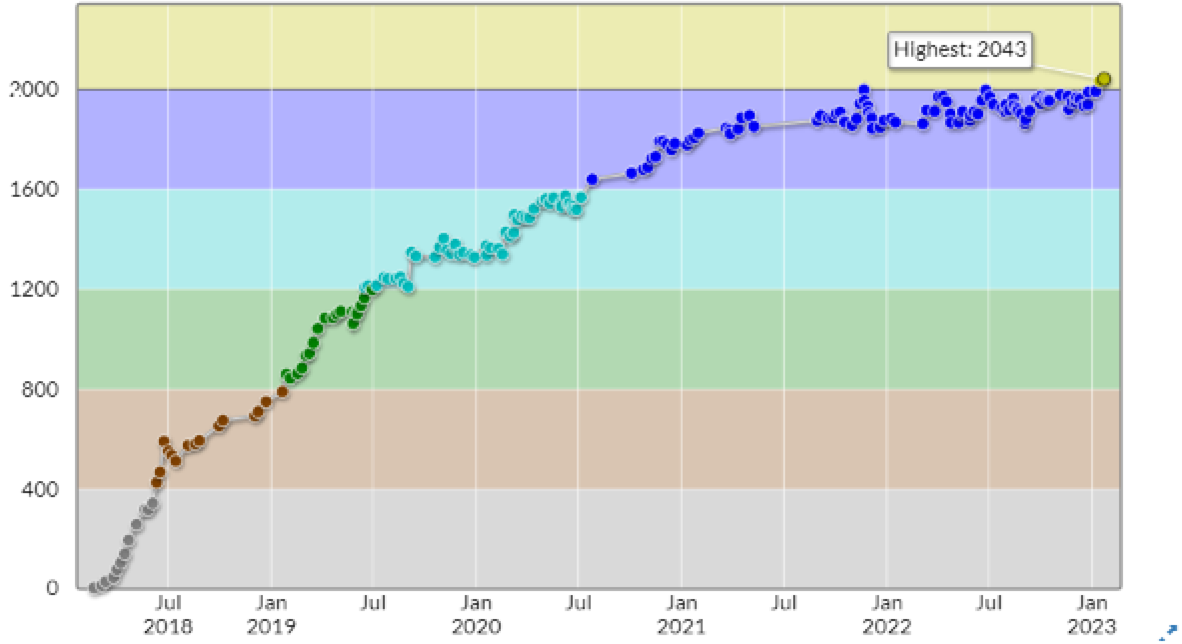
| | |
|---|---|
| 国と地域 | 日本 |
| 誕生年 | 2002 |
| Twitter ID | @Sophia_maki |
| Codeforces ID | shinchankosen |
| 所属 | Osaka University |

# コンテスト実績

Algorithm　Heuristic

| | |
|---|---|
| 順位 | 1161st |
| Rating | 2043 |
| Rating最高値 | 2043 — 初段 (昇格まであと+157) |
| コンテスト参加回数 | 192 |
| 最後に参加した日 | 2023/01/22 |

2043　277th +9　2023/1/22 AtCoder Regular Contest 154



# 競技プログラミングとは

高校数学やアルゴリズムをめっちゃ使うプログラミング

数学オリンピックの知識がすごく使える！
(現に数学オリンピック経験者が多い)

実は高専数学ではそういう数学が少ないのは別の話

1～5時間程度で4～10問程度(1問あたりコード1～300行程度)

AtCoder等で、ほぼ毎週コンテストが開かれ、参加者は1000～20000人程度

近年コーディングテストで重宝

# 競技プログラミングの大会

国際大学対抗プログラミングコンテスト(ICPC)

1チーム3人。高専4年次、アジア地区大会で25位。
(予選通過したら横浜！)

情報オリンピック(JOI)

パソコン甲子園(PCK)

Codeforces

PGbattle

賞金でるよ！！

その他企業コンテスト

TopCoder

# どんな問題？

実行時間制限: 2 sec / メモリ制限: 1024 MB

配点 : 100 点

## 問題文

高橋君の財布の中には 100 円硬貨が 1 枚以上入っており、それ以外には何も入っていません。

高橋君の財布の中の合計金額が $X$ 円である可能性はありますか？

## 制約

- $0 \leq X \leq 1000$
- 入力は全て整数

C++での解答例

```cpp
#include <bits/stdc++.h>
using namespace std;

int main() {
    int x;
    cin >> x;
    cout << (x % 100 == 0 && x > 0 ? "Yes" : "No") << endl;
    return 0;
}
```

幾何

整数論

グラフ理論

確率・組合せ

文字列

パズル(構築)

# 使用するアルゴリズム

深さ優先探索(DFS)

ネットワークフロー(Dinic等)

セグメント木

累積和

幅優先探索(BFS)

最小共通祖先(LCA)

強連結成分分解(SCC)

Union-Find

ローリングハッシュ

行列累乗(一番好きなアルゴリズムです！)

動的計画法(DP)

ダブリング

ダイクストラ(Dijkstra)

# 本題

ARC153でAtCoder 黄色になりました！！

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 190 | shinchan | 1400<br>71:36 | 300<br>5:35 | 500<br>31:03 | 600<br>71:36 | - | - | - | 2348 | 1993→2034 (+41) |

B問題の解説をします！(ネタバレになったらすみません！！)

# 問題概要

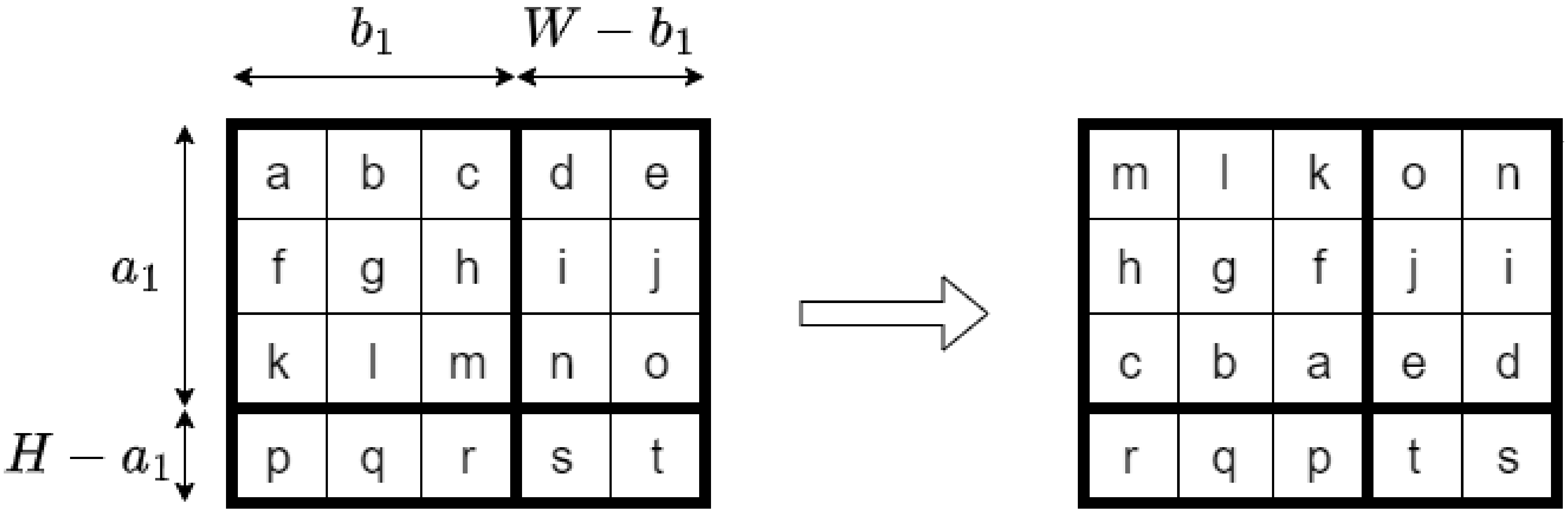
$H \times W$のグリッドが与えられる($HW \leq 5 \times 10^5$)

$Q$個のクエリが与えられる($Q \leq 2 \times 10^5$)。

クエリの形式 : 整数$a, b$ $(1 \leq a \leq H-1, 1 \leq b \leq W-1)$

クエリの詳細は次ページに記載

すべてのクエリ終了後のグリッドの状態を出力




愚直にやったら$O(HWQ)$ → 間に合わない！

画像引用 : https://atcoder.jp/contests/arc153/tasks/arc153_b

# 公式解説・その他の解法

行と列でわけて考えることができる

つまり1次元で考えることができる

でもそこからが難しい

https://atcoder.jp/contests/arc153/editorial/5484

Implicit Treap が使えるらしい。(区間反転が$O(\log N)$でできたりするらしい)

↑ 邪道

# 行列を用いた解法

| | | | | |
|---|---|---|---|---|
| a | | c | d | e |
| | | | | |
| k | | m | n | o |
| p | | r | s | t |

$H = 4, W = 5, a = 3, b = 3$

| | | | | |
|---|---|---|---|---|
| m | | k | o | n |
| | | | | |
| c | | a | e | d |
| r | | p | t | s |

求めたいもの

# 行列を用いた解法

| | | | | |
|---|---|---|---|---|
| a | | c | d | e |
| | | | | |
| k | | m | n | o |
| p | | r | s | t |

$H = 4, W = 5, a = 3, b = 3$

| | | | | |
|---|---|---|---|---|
| t | s | r | | p |
| o | n | m | | k |
| | | | | |
| e | d | c | | a |

$a, b$を気にせず反転してみよう

# 行列を用いた解法

| | | | | |
|---|---|---|---|---|
| t | s | r | | p |
| o | n | m | | k |
| | | | | |
| e | d | c | | a |

$a, b$気にせず反転したもの

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | t | s | r | | p |
| | | | o | n | m | | k |
| | | | | | | | |
| | | | e | d | c | | a |

$(a, b)$だけ平行移動してみよう

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | t | s | r | | p |
| | | | o | n | m | | k |
| | | | | | | | |
| | | | e | d | c | | a |

| | | | | |
|---|---|---|---|---|
| m | | k | o | n |
| | | | | |
| c | | a | e | d |
| r | | p | t | s |

行、列をそれぞれ、$mod\ H, W$してみよう

求めたいものと一致！！

# 座標$(i, j)$の動きを行列にしてみよう

$i$ 行目 $j$ 列目が、$a, b$によって$i'$行目 $j'$列目に動くとする。($i, j$ は0-indexed)

$$\begin{pmatrix} i' \\ j' \end{pmatrix} = \begin{pmatrix} H-1-i \\ W-1-j \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix} \pmod{H, W}$$

$$= \begin{pmatrix} a-1 \\ b-1 \end{pmatrix} - \begin{pmatrix} i \\ j \end{pmatrix} (\mathrm{mod}\text{なので } H, W \text{省略})$$

$$\begin{pmatrix} i' \\ j' \\ 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & a-1 \\ 0 & -1 & b-1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}$$

# 座標$(i,j)$の動きを行列にしてみよう

$$\begin{pmatrix} i' \\ j' \\ 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & a-1 \\ 0 & -1 & b-1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}$$

$k$番目のクエリについて$A_k = \begin{pmatrix} -1 & 0 & a_k - 1 \\ 0 & -1 & b_k - 1 \\ 0 & 0 & 1 \end{pmatrix}$とすると、操作後の座標は、

$$\begin{pmatrix} i' \\ j' \\ 1 \end{pmatrix} = A_Q A_{Q-1} \dots A_1 \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}$$

$A_Q A_{Q-1} \dots A_1$を前計算しておくことで、

あとから各座標を$O(1)$で計算できる！

全体$O(Q + HW)$！！

# 実装方針

1行目と2行目で別のmodを使うため、modintは使えない。

でもlong long で大丈夫！($i$ の絶対値は一回のクエリで高々$a$しか変化しない)

long long で行列演算後、modすればいい。

このままでもできるが、、

$$\begin{pmatrix} i' \\ 1 \end{pmatrix} = \begin{pmatrix} -1 & a-1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ 1 \end{pmatrix}, \begin{pmatrix} j' \\ 1 \end{pmatrix} = \begin{pmatrix} -1 & b-1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} j \\ 1 \end{pmatrix}$$

こうすればmodintも使えて実装楽だと思う

# 行列でできること(競プロ関係)

- 行列累乗　(一番好きなアルゴリズムです！！)
- 幾何における変換

（今回の問題もこれ。アフィン変換とか。行列ではなく複素数を使う場合も。）

- 連立方程式を解く　(競プロではあまり使わないかも？)

# 結論

行列を使うと幸せになれます

※宗教勧誘ではありません