

# J: Matrix

@beet\_aizu

# 概要

長さHの数列Aと長さWの数列Bが与えられる。

H行W列からなる行列Cを以下のように定義する。

$$C_{i,j} = A_i \times B_j (1 \leq i \leq H, 1 \leq j \leq W)$$

以下の四種類のクエリを合計Q回処理せよ。

Aの区間[a,b]にvを一様加算

Bの区間[c,d]にvを一様加算

Cの矩形領域[a,b] × [c,d] の最小値とその個数を出力

Cの矩形領域[a,b] × [c,d] の最大値とその個数を出力

# 考察

クエリからして明らかにセグ木

区間add, 区間min, 区間max, 区間countがあればできそう

## 考察2

Cの最小、最大値はA,Bの最小、最大値で構成されている

→区間countが必要なのは最大、最小値だけ

遅延セグ木（あるいはStarrySky木）で処理できる

# コーナーケース

答えが0のときは0以外を数えて全体から引く

- 区間に含まれる0の数を数える必要があるが、これは最大値か最小値に含まれているはず  
(少し考えるとそれはそう)

最大値が負になることもある

最大値と最小値が一致することもある

## ans=0のケース

整数環 $\mathbb{Z}$ は整域なので非自明な零因子が存在しない  
→  $A, B$ の中の0の数を数えればよい

最小値が0のとき、

$A_i \in [0, \infty)$  かつ  $B_i \in [0, \infty)$

または

$A_i \in (-\infty, 0]$  かつ  $B_i \in (-\infty, 0]$

## ans=0のケース

最大値が0のケースも同様に

$A_i \in [0, \infty)$  かつ  $B_i \in (-\infty, 0]$

または

$A_i \in (-\infty, 0]$  かつ  $B_i \in [0, \infty)$

したがって、いずれのケースでも

0が（どちらかの区間内の）最小値または最大値となる。

→ 最小値と最大値の数を数えるセグ木で求められる。

# 気持ち

セグ木は抽象化しようね！

注意点が無限にあるので落ち着いて実装しましょう

平方分割落としがなかったなあ（小声）

（人の心があるのでやめた）

## 講評

- First Accepted:
  - Onsite: rupc\_latte\_chikoku (3h22m53s)
  - Online: Cppi (1h20m28s)
  
- Success Rate: 12.5% (5/40)

# ジャッジ解

名前	言語	行数
beet	C++	184
uku	C++	174
haji	C++	179
Gacho	C++	281