

# 会津合宿2017 Day3 E問題

## たい焼きマスターと食べ盛り

- 原案: 杉江
- 問題文: 瀧澤
- 解答: 鈴木、杉江、瀧澤
- 解説: 瀧澤

# 問題概要

- $H \times W$  の長方形のたい焼きプレートに対して以下のクエリを  $Q$  個処理する
  - 時刻  $t$  にマス  $(h, w)$  にたい焼きを置く
    - 置いたたい焼きは  $T$  分後に焼き上がる
  - 時刻  $t$  にマス  $(h, w)$  にあるたい焼きを食べる
    - 焼き上がっていないならば食べない
  - 時刻  $t$  の時点で領域  $[(h_1, w_1), (h_2, w_2)]$  にあるたい焼きを数える
    - 焼き上がっているものとそうでないものを分けて数える

# 制約

- $1 \leq H, W \leq 2 \cdot 10^3$
- $1 \leq Q \leq 10^5$
- $1 \leq T \leq 10^9$
- $1 \leq t_i \leq 10^9$

# 解説(1/4)

- 以下を高速に処理できるデータ構造が欲しい
  - 点  $(h, w)$  に対して加減算を行う
  - 領域  $[(h_1, w_1), (h_2, w_2)]$  の総和を求める
    - それぞれ  $O(\log H \cdot \log W)$  程度でやりたい
- これは二次元 BIT で行える
  - ググるとわかりやすいスライドが出てくる
  - [http://hos.ac/slides/20140319\\_bit.pdf](http://hos.ac/slides/20140319_bit.pdf)

# 解説(2/4)

- 二次元 BIT を二つ用意する
  - 片方は焼き上がったたい焼きを数える
  - もう片方はそうでないたい焼きを数える
    - 便宜上、それぞれ順に A, B と呼ぶことにする
- キューを一つ用意する
  - たい焼きの情報を詰める
  - {置いたマス, 置いた時間} を持っておく

# 解説(3/4)

- 各クエリの処理前に毎回以下を行う
  - キューの先頭を見て、焼き上がったか確認する
  - 焼き上がっていれば BIT を更新
    - A の該当箇所に +1
    - B の該当箇所に -1
- クエリは一つずつ個別に処理する
  - 各クエリの処理の方法は次ページの通り
  - 時間計算量は  $O(Q \cdot \log H \cdot \log W)$
  - 空間計算量は  $O(Q + HW)$

# 解説(4/4)

- クエリ1(たい焼きを置く)
  - たい焼きをキューに詰める
  - B の該当箇所  $+1$
- クエリ2(たい焼きを食べる)(食エリ)
  - A の該当箇所が 1 であれば  $-1$
- クエリ3(たい焼きを数える)
  - 各 BIT について該当領域のたい焼きを数える

# Writer 解

- C++
  - 鈴木 : 60行
  - 杉江 : 84行、103行
- Java
  - 瀧澤 : 108行、169行