

歴史の研究 (Historical Research)

解説

秀 郁未 (tozangezan)

概要

- 下図のように日付と出来事の対応が与えられる。

日付	1	2	3	4	5	6	7	8	
出来事	9	9	19	9	9	15	9	19	

- さらにクエリが与えられる。
- 各クエリは「(出来事の種類)*(範囲における個数)」という重要度の最大値を求める。
- $N \leq 100000$, $Q \leq 100000$, 種類の整数 $\leq 10^9$

例

- クエリの範囲が1日目～4日目

日付	1	2	3	4	5	6	7	8		
出来事	9	9	19	9	9	15	9	19		27

- $9 * 3 = 27$
- $15 * 0 = 0$
- $19 * 1 = 19$
- 27を出力

例

- クエリの範囲が3日目～6日目

日付	1	2	3	4	5	6	7	8		
出来事	9	9	19	9	9	15	9	19		19

- $9 * 2 = 18$
- $15 * 1 = 15$
- $19 * 1 = 19$
- 19を出力

例

- クエリの範囲が2日目～8日目

日付	1	2	3	4	5	6	7	8		
出来事	9	9	19	9	9	15	9	19		38

- $9 * 4 = 36$
- $15 * 1 = 15$
- $19 * 2 = 38$
- 38を出力

例

- クエリの範囲が1日目～8日目

日付	1	2	3	4	5	6	7	8		
出来事	9	9	19	9	9	15	9	19	45	

- $9 * 5 = 45$
- $15 * 1 = 15$
- $19 * 2 = 38$
- 45を出力

ありふれた解法(小課題1)

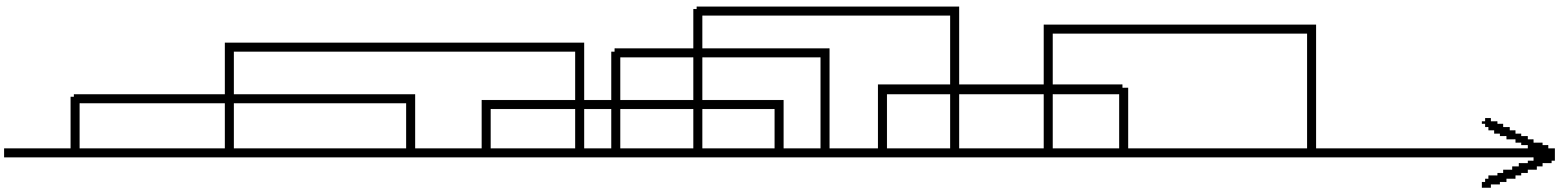
- 範囲の中にない種類の重要度は全部0
 - 範囲の中にある種類すべてに対して個数を数える
 - 最大値を求める
 - **オーバーフローには注意**
-
- **$O(QN^2)$** $Q \leq 100, N \leq 100$ は通る。**5点**
 - **おそらく今日の問題の中で最も簡単な部分点です**
 - **ちゃんと取りましたか??????**

さっきの改善(小課題2)

- 結構さまざまな解法があると思います
- 累積和
 - 各数字について累積和を取る
 - クエリは各数字について範囲の和を求めてmaxをとる
 - $O(N(Q+N))$
- 二分探索
 - 各数字について出現場所をvectorとかに入れる
 - クエリごとにupper_boundとlower_boundで数える
 - $O(QN \log N)$
- **10点**

特殊なケース(小課題3)

- クエリが互いに他を含まない。
- つまりこんな感じ



特殊なケース(小課題3)

- 矢印が書いてありますね…
- 左端から処理していこう。
- 尺取的に左端(開始日)と右端(終了日)を動かすとうまくいくだろう。
- 各段階で最大値を高速に取得できれば良い
- →ん？

特殊なケース(小課題3)

- Segment Tree
 - 詳しくは <http://www.slideshare.net/iwiwi/ss-3578491>
 - 今回は更新と最大値を求めるだけの単純なやつ
- 「出来事の種類」は高々N種類しかない
- 座標圧縮をしよう
- まずクエリを開始or終了が早い順にソート
- 座標圧縮→尺取(ここでSegment Treeを使う)でそれぞれのクエリに対する最大値を求める+更新
- $O((N+Q) \log (N+Q))$ **25点**

満点解法

- このままSegment Treeだけ使っていけるだろうか
… → いけなさそう…
- 何か変な制約があるだろうか … → なさそう…
- じゃあ、データ構造で(半ば強引に)改善しよう
- 何を使おうかな??
 - なんとら木 → 絶対やばい
 - 複雑なSegment Tree → 無理そうだとさっき言った

満点解法

- このままSegment Treeだけ使っていけるだろうか
… → いけなさそう…
- 何か変な制約があるだろうか … → なさそう…
- じゃあ、データ構造で(半ば強引に)改善しよう
- 何を使おうかな??
 - なんとら木 → 絶対やばい
 - 複雑なSegment Tree → 無理そうだとさっき言った
- **バケツト法**

満点解法

- とりあえずバケットのサイズは B として後で決めることにしよう。
- バケット法による解法はいくつかあると思います。
- そのうち1つを紹介します。Segtree使いません。
- 一応、時間だけでなく空間計算量も考えることにしましょう。
- (バケット法の問題だとメモリ量ちゃんと考える必要があることも)

前処理①

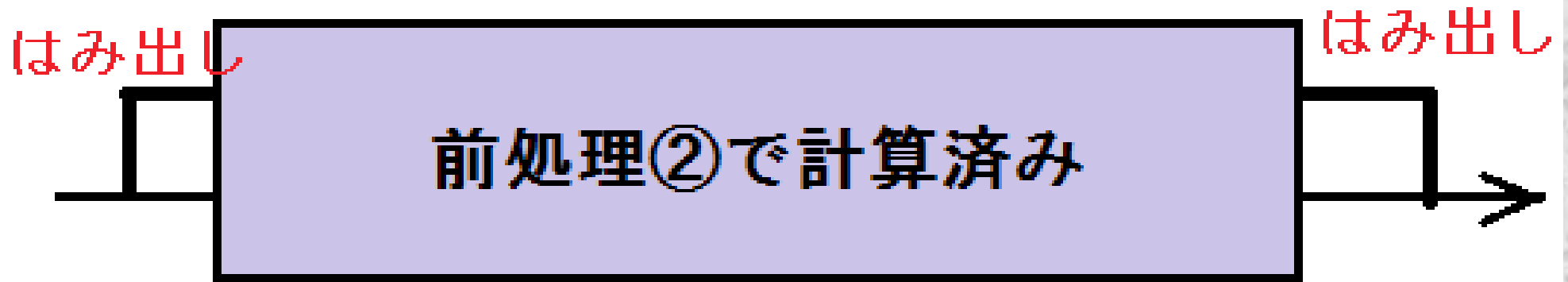
- 各数字について出現位置を昇順でvectorとかにつめておく。
- (座標圧縮が必要)
- この処理は 時間 $O(N \log N)$, 空間 $O(N)$

前処理②

- $(i \cdot B + 1)$ 日目から $j \cdot B$ 日目 ($0 \leq i < j \leq N/B$)における「重要度の最大値」を求める
- この処理は、まずそれぞれの i に対してここを最初の日としたときの種類ごとの和を配列にもって、さらに最大値も持っておけばそれぞれの i に対し合計で時間 $O(N)$ ($j = i+1, \dots, N/B$ を一遍に計算)
- i の選び方は $O(N/B)$
- 時間 $O(N^2/B)$ 空間 $O(N^2/B^2)$

本処理

- クエリの範囲は、下図のように分けられる。



- 前処理②で計算済みの最大値を仮の答えとする
- それを越える答えがあるとしたら、その値に関する「種類」のものがはみ出した部分にある

本処理

- **はみ出した**部分のそれぞれの種類に対して、前処理①で作った出現位置のものを利用して upper_bound と lower_bound で個数を数える
- 前ページのものともあわせて最大値が答え。
- 時間 $O(QB \log N)$

計算量

- 前処理①

時間 $O(N \log N)$ 空間 $O(N)$

- 前処理②

時間 $O(N^2/B)$ 空間 $O(N^2/B^2)$

- 本処理

時間 $O(QB \log N)$

- 合計

時間 $O((N+QB) \log N + N^2/B)$ 空間 $O(N + N^2/B^2)$

- $B=100$ 程度で上手く間に合うものができる

ほかの解法

- 平方分割+Segment Tree(1箇所updateする,区間のmaxを求める)でも解けます
- $O(N * (N \log N)^{1.5})$ 程度?
- 何にせよこういう問題でTLEするときはバケットサイズをいろいろ試して送ることが大事

omake

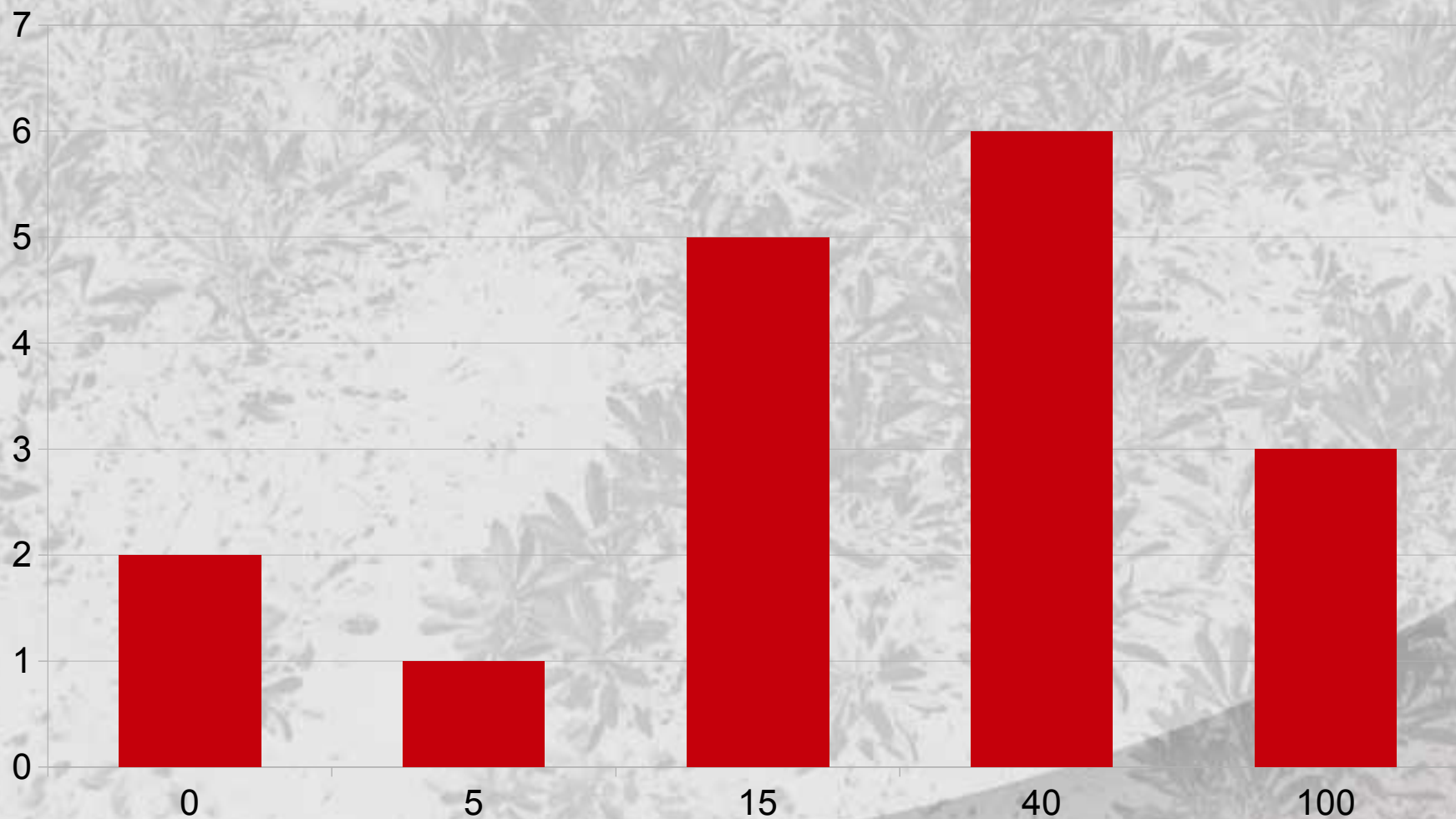
- なんか定数倍がきついらしい？
- バケット法とかの問題は基本的に定数倍が厳しく見えがちです。
- 定数倍改善力や謎エスパー力や嘘解法力がほしいあなたのために最適の練習環境をお教えします。(JOIで使える保障は無いです)

omake

- なんか定数倍がきついらしい？
- バケット法とかの問題は基本的に定数倍が厳しく見えがちです。
- 定数倍改善力や謎エスパー力や嘘解法力がほしいあなたのために最適の練習環境をお教えします。(JOIで使える保障は無いです)

poj.org

得点分布



得点分布

