九州大学プログラミングコンテスト2018
解説

九州大学ICPCチャレンジ部

結果

- 優勝 (全体)

  ainta : 5400点 (全完) 148:50

- 優勝 (オンサイト)

  WA_TLE : 5400点 (全完) 186:59 [全体2位]

- 参加者数

  283 人

統計

	問題名	正解者数	提出者数	提出数
A	QUPC	275	276	309
B	Tapu & Tapi	199	257	767
C	Ito Campus	96	162	479
D	Novelist	62	85	207
E	Treeone	69	91	153
F	Team Making	30	34	43
G	Tapu & Tapi 2	18	26	51
H	ukuku	7	23	64
I	Buffalo	5	9	26
J	Repeat Strings	9	11	23

First Accept

	問題名	First Accept
A	QUPC	shibh308 (00:21)
B	Tapu & Tapi	square1001 (02:24)
C	Ito Campus	square1001 (10:53)
D	Novelist	WA_TLE (21:36)
E	Treeone	WA_TLE (05:30)
F	Team Making	kirika_comp (20:55)
G	Tapu & Tapi 2	SyntaxSato (team) (46:23)
H	ukuku	ainta (62:35)
I	Buffalo	ainta (29:32)
J	Repeat Strings	square1001 (40:33)

A : QUPC

writer & 解説 : treeone

tester : ei13333, climpet, konjo, blue_jam, nikollson

A : QUPC

問題概要

2014 年から 4 年に 1 度 QUPC が開催されるとき,
N 回目の QUPC が開催されるのは何年になるか？

制約：$3 \leqq N \leqq 1333$

解法

$N \times 4 + 2010$ を出力する

B : Tapu & Tapi

writer & 解説 : treeone

tester : ei13333, climpet, konjo, blue_jam

B : Tapu & Tapi

問題概要

A 枚の 100 円玉と B 枚の 10 円玉と C 枚の 1 円玉を持っている．これらの硬貨を等しい金額になるように 2 人に分けることができるか？

制約：$0 \leq A, B, C \leq 10^9$ (満点)

$0 \leq A, B, C \leq 100$ (部分点)

B : Tapu & Tapi

解法 (部分点)

- 一方に i 枚の 100 円玉と j 枚の 10 円玉と
  k 枚の 1 円玉をあげるとき,
  他方には $A-i$ 枚の 100 円玉と $B-j$ 枚の 10 円玉と
  $C-k$ 枚の 1 円玉をあげることになる.
- i, j, k を全通り試して,
  $100i + 10j + k = 100(A-i) + 10(B-j) + C - k$
  が成り立つことがあるかどうかを調べればよい.
- 計算量は $O(QABC)$

B : Tapu & Tapi

解法 (満点)

- 合計金額を $S = 100A + 10B + C$ 円とすると,
  A 枚の 100 円玉と B 枚の 10 円玉と C 枚の 1 円玉で
  ちょうど $S/2$ 円を払うことができるか？という問題.
- S が奇数なら “No”

B : Tapu & Tapi

解法 (満点)

- S が偶数のとき，$S/2$ 円を
  - 100 円玉で払えるだけ払う
  - 残りを 10 円玉で払えるだけ払う
  - さらに残りを 1 円玉で払えるだけ払う

  最終的に残りを 0 円 にできれば “Yes”
- 計算量は $O(Q)$

C : Ito Campus

writer & 解説 : treeone

tester : ei13333, climpet, konjo

C : Ito Campus

問題概要

縦 H × 横 W の迷路が与えられる．壁以外の隣接するマスに移動できる．いくつかのマスにはイノシシがいる．
X 回以下の移動で行ける全てのマスに
イノシシがいない場合，そのマスは安全なマスである．
安全なマスのみを通ることができるとき，
スタートからゴールまでの最短距離を求めよ．

制約：$4 \leq H, W \leq 1000$ (満点)
$4 \leq H, W \leq 50$ (部分点)

C : Ito Campus

解法

- 最も近いイノシシとの距離が X 以下のマスは通ることができない.
- Step 1 : 各マスから最も近いイノシシまでの距離を求める.
  - 全てのイノシシをスタート地点として幅優先探索で求めることができる.
  - 1 匹のイノシシをスタート地点としてイノシシのマス毎に幅優先探索すると TLE (部分点)

C : Ito Campus

解法

- Step 2：スタート地点から幅優先探索でゴール地点までの最短距離を求める．このとき，最も近いイノシシまでの距離が X 以下のマスは壁として扱う．
- 計算量は $O(HW)$

D : Novelist

writer & 解説 : treeone

tester : ei13333, climpet, konjo, blue_jam

D :Novelist

問題概要

王都と都市 $1, 2, \ldots, N$ があり，
王都を A_i 日目に出発し，都市 X_i に $A_i + T_{X_i}$ 日目に到着する依頼が M 件，
都市 Y_i を B_i 日目に出発し，王都に $B_i + T_{Y_i}$ 日目に到着する依頼が L 件ある．
期間が重ならないように依頼を受ける時，最大で幾つの依頼を受けることができるか？

制約：$1 \leq N, M, L \leq 10^5, 0 \leq T_i \leq 10^9, 1 \leq A_i, B_i \leq 10^9$

D :Novelist

解法

- 王都からある都市 i に行って，都市 i から王都に帰ってくるまでを 1 つのタスクとして，
  これらのタスクの中から重ならないようにできるだけ多くのタスクを選ぶという問題を考える.
  (区間スケジューリング)
- 都市 i に行ってから最速で何日目に王都に帰って来れるかは，都市毎に依頼を時間でソートして，
  二分探索や尺取り法によって求めることができる.
- あとは王都に帰って来る日が早い順に貪欲にタスクを選んでいくことで求まる.

E : Treeone

writer & 解説 : ei13333

tester : treeone, climpet, konjo

E： Treeone

問題概要

長さ N の数列 A_i が与えられる

1 つの要素を好きな値に変更できるとき
総和が 0 となる連続部分列の個数の最小値を求めよ

制約：$1 \leq N \leq 10^5, |A_i| \leq 10^9$

E : Treeone

解法

累積和をとってみる

1	-1	-1	2	-2	0

0	1	0	-1	1	-1	-1

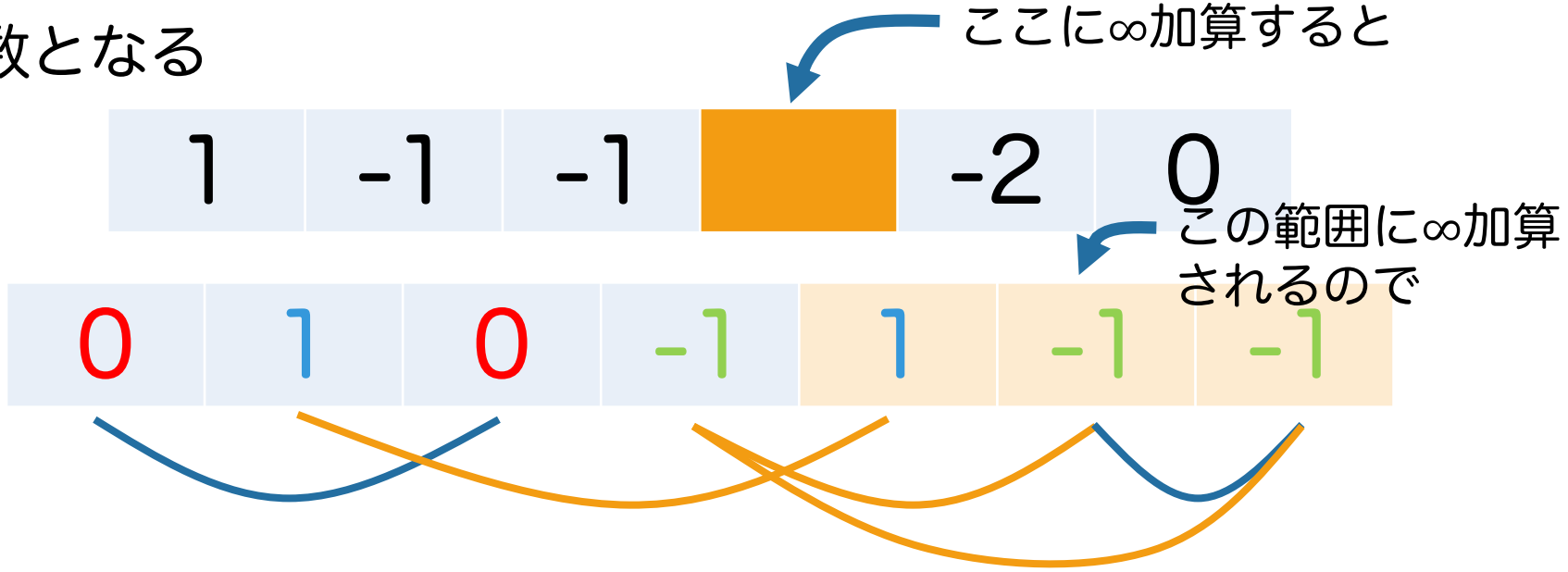
同じ値同士を結んだ区間の和は 0 になっていることがわかる

それぞれの値についてその値の個数が分かれば
数列全体の総和が 0 の連続部分列の個数を求められる

E : Treeone

解法

総和が 0 の連続部分列の個数は、ある要素を 1 つ変更するとその要素を含まない総和が 0 の連続部分列の個数となる



E : Treeone

解法（部分点）

要素の和が 0 の連続部分列は左端を決め打ちして
右端を左から右にずらすと $O(N^2)$ で全列挙可能

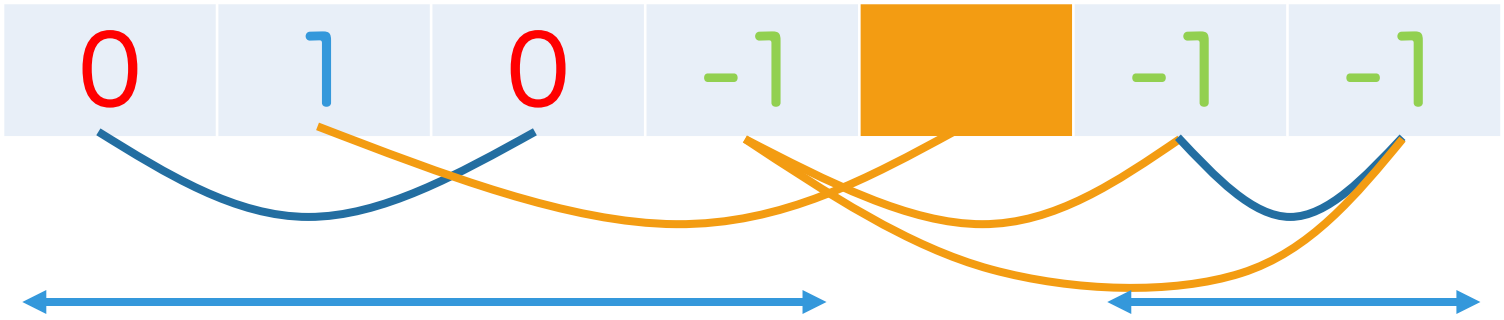
ある要素を含まない総和が 0 となる連続部分列の個数も
がんばると求めることができる

E : Treeone

解法（満点）

要素 i を含まない連続部分列の個数

= 要素 i より左で終わる 0 区間の個数 +
　要素 i より右で始まる 0 区間の個数



この場合　2 個存在する

E : Treeone

解法（満点）

結局 要素 i で終わる(始まる)連続部分列の個数を求めて
累積和をとればよい

要素 i で終わる連続部分列の個数は
要素 i より左でその値が出現した回数なので
配列を左から右に走査することで求めることが可能

全体の計算量は $O(N \log N)$

F : Team Making

writer : treeone

解説 : ei13333

tester : ei13333, climpet, konjo

F： Team Making

問題概要

N 人いて i 番目の人のレーティングは A_i である.

平均レートが K 以下となる 3 人以下のチームを組む

ただし 1 人チームの場合はレートは問わない

このときチームの組み方は何通りあるか求めよ

制約：$1 \leq N \leq 18, 0 \leq K, A_i \leq 4208$

F： Team Making

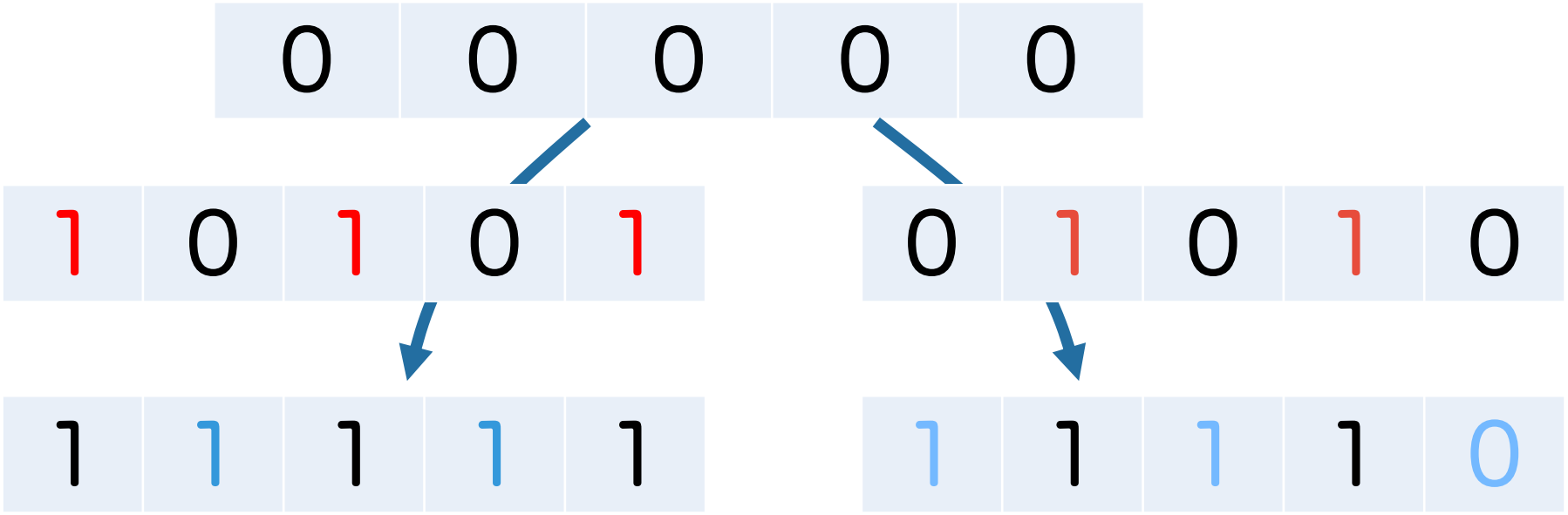
制約をよくみると bitdp ということがわかる

- ある人が既にチームを組んだかどうかを
  i 桁目の bit が立っているかどうかで管理する

dp[bit] := bit が立っている人たち同士でチームを組む
　　組み合わせの個数　とする

F： Team Making

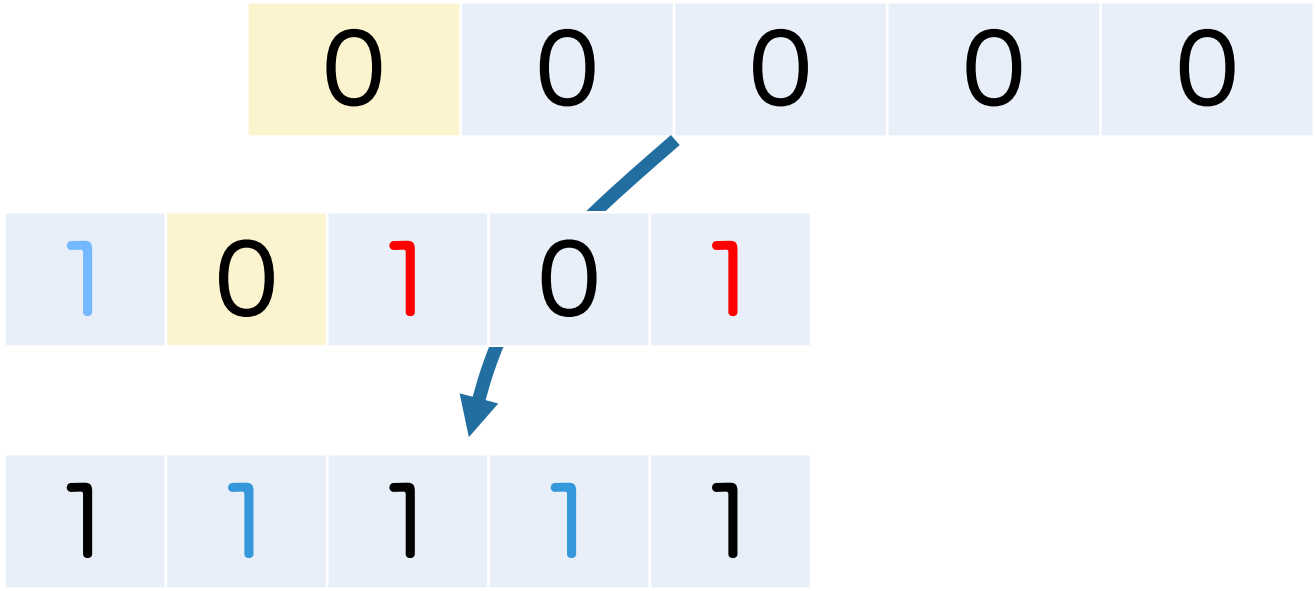
ある状態 bit から bit が立っていない人から
適当にチームを決めると重複して数え上げられてしまう



これは2通りではなく1通りとして数えたい

F： Team Making

チームのうち 1 人は bit が立っていない最左の人を
使うことを決めておけばいい



これで1通りとして数え上げられる

F： Team Making

遷移は 1〜3 人すべての組み方を試せば良い

- ただし最初の1人は使ってない人のうち最左に固定
- 3 人決める時には $O(N^2)$ かかる

状態数は 2^N なので$O(2^N \times N^2)$

G : Tapu & Tapi 2

writer & 解説 : ei13333

tester : treeone, climpet, konjo

G : Tapu & Tapi 2

問題概要

N 頂点の重み付き木が与えられる

いくつかの頂点にはたぴちゃんとたぷちゃんがいる

たぴちゃんとたぷちゃんを同じ連結成分に所属させないために取り除く辺の最小重みを求めよ

制約：$2 \leq N \leq 5 \times 10^5$

G : Tapu & Tapi 2

解法（部分点）

最小カット

たぴ同士、たぷ同士を結ぶ辺を取り除いておく

するとたぴがいる全ての頂点からたぷがいる全ての頂点への最小カットがこの問題の答えであることがわかる

- 例えば超頂点S, Tを用意し、
  Sからたぴがいる頂点、たぷがいる頂点からTへ、
  容量∞の辺をはって S から T への最大流を流す

Dinicで $O(V^3)$

G : Tapu & Tapi 2

解法（満点）

赤色をたぴ、青色をたぷとする

色が塗られていない頂点にいずれかの色を塗ったあと
違う色同士を結ぶ辺の重みを最小化する問題

木DPをすればよい

dp[idx][color] := 頂点idxを根とする部分木で、頂点idx
をcolorで塗ったときの辺の重みの最小値 とする

G : Tapu & Tapi 2

解法（満点）

idx の子についての計算は終わっているとする

dp[idx][赤] = dp[idx][青] = 0;
ただし既にidxの色が決まっている時はもう一方の色を∞

idx のすべて子 child について

dp[idx][赤] += min(dp[child][青] + 辺の重み, dp[child][赤]);
dp[idx][青] += min(dp[child][赤] + 辺の重み, dp[child][青]);

全体の計算量は $O(N)$

H : ukuku

writer & 解説 : treeone

tester : ei13333, climpet, konjo

H : ukuku

問題概要

長さ N の数列 A_i が与えられる

i 番目の文字を中心とする最長の回文の長さが A_i であるような，長さ N のアルファベット小文字からなる文字列をどれか 1 つ構成せよ.

制約：$1 \leq N \leq 2 \times 10^5$

H : ukuku

解法

先頭から貪欲に決めていくことができる.

i 番目を中心とする最長の回文半径 $r_i := (a[i] + 1) / 2$

- 手順 1 : i 番目の文字が決まっていなければ，その位置で使えない文字以外を 1 つ選ぶ
- 手順 2 : $i - r_i$ 番目の文字と $i + r_i$ 番目の文字は異なるので，$i + r_i$ 番目の位置に $i - r_i$ 番目と同じ文字は使えないということを記録しておく．($i - r_i > 0$ のとき)
- 手順 3 : $i + r_i - 1$ 番目までで文字が決まっていない位置 j があれば，$s[j] := s[2i - j]$ とする.

H : ukuku

解法

- 英小文字だけで文字の種類は足りるのか？

  ある位置を末尾とする回文をその周期で
  グループ分けすると，$O(\log N)$ 種類に分けられる.
  ある位置で使えない文字の種類は最大 $\log N$ 個程度.
  今回の問題では 18 種類以下の文字で構成可能.

- 計算量は $O(N)$.

I : Buffalo

writer : treeone

解説 : ei13333

tester : ei13333, climpet

I : Buffalo

問題概要

N 個の容器があり、i 番目の容器の容量は A_i [L]である

以下の操作の繰り返しで K [L] の水を汲み出せるような容器のペア (X, Y) は何通りあるか求めよ

操作1: 一方の容器を満杯にする
操作2: X から Y に、Y が満杯か X が空になるまで移す
操作3: 一方の容器内の水を全て捨てる

- 制約：$2 \leq N \leq 3\times10^5, 1 \leq K \leq 2\times10^6, 1 \leq A_i \leq 10^6$

I : Buffalo

解法

操作1: 一方の容器を満杯にする
操作2: X から Y に、Y が満杯か X が空になるまで移す
操作3: 一方の容器内の水を全て捨てる

上の操作の繰り返しで K [L] を汲み出せる

必要十分条件　がんばるとわかる
実験してもわかる

$\gcd(A_X, A_Y)$ が K の約数 かつ $A_X + A_Y \geq K$

I : Buffalo

解法

それぞれの $i(1 \leq i \leq K)$ について
$\gcd(A_X, A_Y) = i$ かつ $A_X + A_Y \geq K$ となるような
ペア (X, Y) の個数を求めたい

包除原理をしたいなあという気持ちになる

- 約数系包除 [検索]

I : Buffalo

$\gcd(A_X, A_Y)$ が i の倍数かつ $A_X + A_Y \geq K$ となるペアの個数 $f(i)$ は比較的容易に求まる

- A_X, A_Y がともに i の倍数かつ $A_X + A_Y \geq K$
- i の倍数である容器を昇順で全列挙したあとに尺取り法などで計算できる

ここでの計算量は調和級数の和を考えると $O(K \log K)$

I : Buffalo

次に i を大きい順から $\gcd(A_X, A_Y)$ がちょうど i であるペアの個数 $g(i)$ を求める

$g(i)$ は $f(i)$ から $\sum g(i$より大きい i の倍数$)$ をひいたもの

- i より大きい値 j については $g(j)$ は計算済みなので求めることができる

ここでの計算量も $O(K \log K)$

J : Repeat Strings

writer & 解説: ei13333

tester : treeone, climpet

J：Repeat Strings

問題概要

ab列 S と Q 個のab列 T_i が与えられる

長さが $|S|$ になるまで T_i を繰り返した列を T_i' とする

S に対して区間を選んで反転する操作を行えるとき、
S を T_i' にするための最小操作回数をそれぞれ求めよ

制約：$1 \leq |S|, Q, \sum|T_i| \leq 2\times10^5$

J：Repeat Strings

解法

とりあえずビット列としてもよい

以下の問題について考える

長さが等しいビット列 S と T が与えられる

S に対して区間を選んで反転する操作を行えるとき、
S を T にするための最小操作回数を求めよ

J：Repeat Strings

解法

とりあえず隣接要素とのxorをとりたい気持ちになる

以後これをxor列と呼ぶことにする

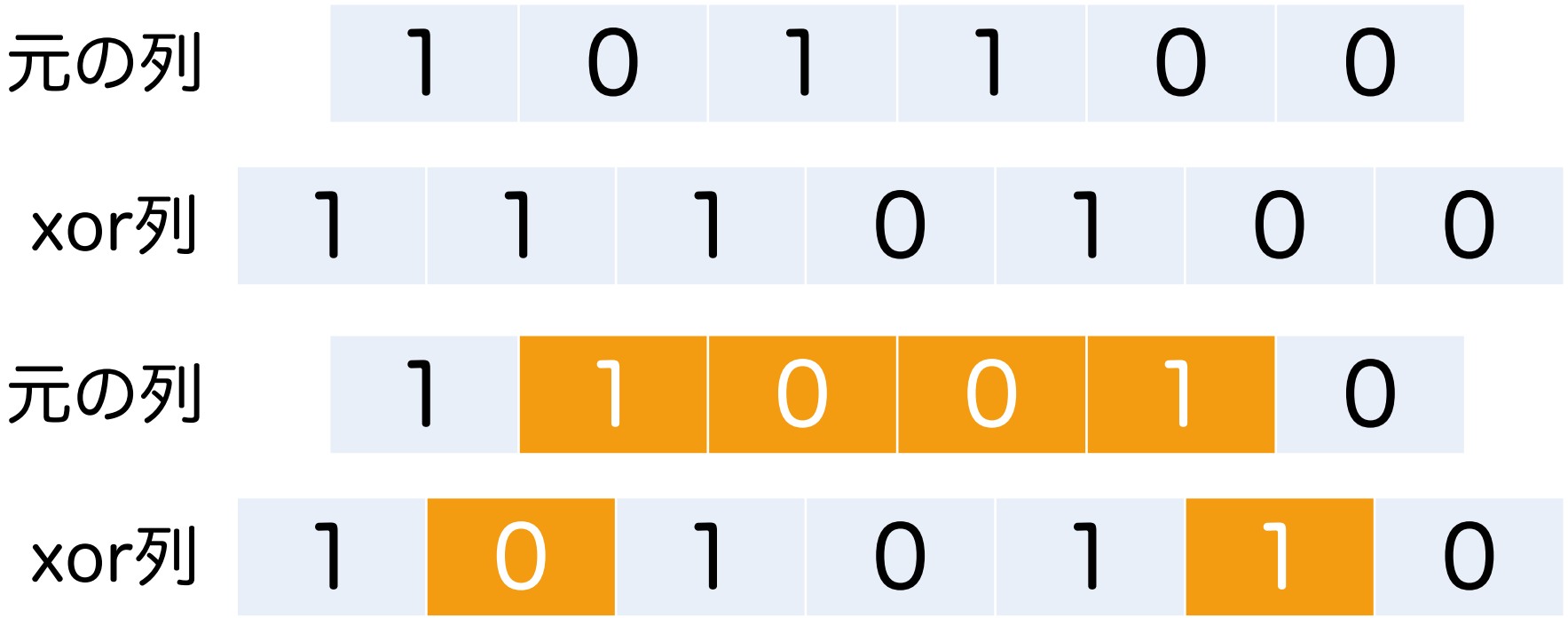
元の列	1	0	1	1	0	0

xor列	1	1	1	0	1	0	0

J：Repeat Strings

解法

すると区間を選んで反転する操作は
2 つの要素を選んで反転する操作に言い換えられる



J：Repeat Strings

解法

つまり S を T にするための最小操作回数は

S のxor列と T の xor列のハミング距離に一致する

したがって $O(|S| + |T|)$ で解くことができた

もともとの問題にもどってみる

愚直にハミング距離を求めていては $O(Q|S|)$ かかって
間に合わないのでこれを高速化することを考える

J：Repeat Strings

解法

$\sum|T_i| \leq 2\times10^5$ の制約より $|T_i|$ の種類数は
$2\sqrt{2\times10^5}$ 以下になることが示せる

- $|T_i|$ が $\sqrt{2\times10^5}$ 以下の文字列の種類数は $\sqrt{2\times10^5}$ 以下
  （そもそも$\sqrt{2\times10^5}$ 種類しかないので）
- $|T_i|$ が $\sqrt{2\times10^5}$ 以上の個数(=種類数)は $\sqrt{2\times10^5}$ 以下
  （$\sqrt{2\times10^5}$ 個より多く存在すると $\sum|T_i| \leq 2\times10^5$ を満たせない）

J : Repeat Strings

解法

$|T_i|$ の長さが同じグループごとにまとめて計算する

いま長さ M のグループを処理しているとする

- T_i の繰り返しの 1周目と最終周目は愚直に処理する
  （両端の処理が面倒なので）
- 2周目から(最終周目-1)周目は…？
  S のxor列の2周目から(最終周目-1)周目の区間の要素を
  $\bmod M$ で累積和をとっておけば
  それぞれ $O(|T_i|)$ で処理できる

J : Repeat Strings

解法

全体の計算量は $O\left(|S|\sqrt{\sum|T_i|}\right)$

J : Repeat Strings

別解

愚直にハミング距離を調べると $O(Q|S|)$ で TLE

例えば64bitごとにビット並列でハミング距離を求めると

$O\left(\frac{Q|S|}{64}\right)$ になるがちょっときびしい

ここで同じ文字列のクエリをメモ化しておくと
$Q \leq 16624$ が保証されるので間に合う

かんたん

ご参加ありがとうございました

QUPC2018 スタッフ

【Writer】

treeone, ei13333

【Tester】

climpet, konjo, blue_jam, nikollson