

問題 5 毒蛇の脱走 解説

DEGWER

問題概要



へび〜〜〜
〜!!!



問題概要

咬まれた痛い



申し訳ねえ



問題概要



問題概要



毒性の合計はどれくらいかな？

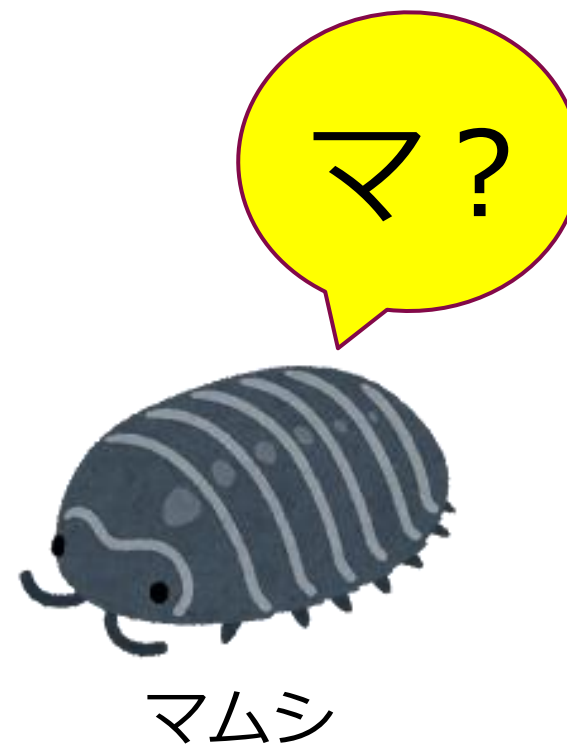
へびがへびー

問題概要

- ▶ 2^L 匹の毒蛇があり、0 から $2^L - 1$ までの番号がついていて、各々には毒性が定まっている
- ▶ 次のクエリ Q 個に答えよ
 - ▶ 0,1,? からなる文字列が与えられるので、番号の二進表記がそのパターンにマッチする毒蛇の毒性の合計を求めよ
- ▶ $L \leq 20, Q \leq 10^6$
- ▶ メモリ制限が小さい (64MB)

小課題1

- ▶ $L \leq 10, Q \leq 10^3$
- ▶ やる
- ▶ $O(2^L Q)$ となり、5点が得られる



小課題2

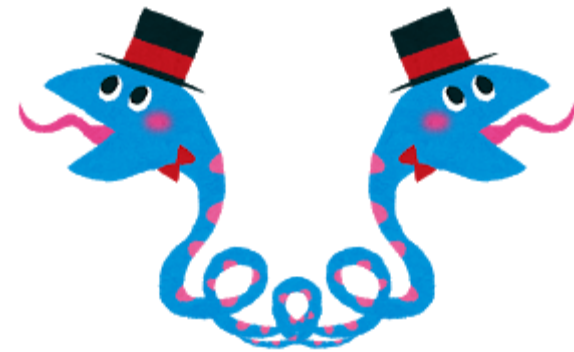
- ▶ $L \leq 10, Q \leq 10^6$
- ▶ あらかじめ答えを全部前計算しておけばよい
- ▶ $O(4^L L + Q)$ となり、計 12 点が得られる
- ▶ 小課題 1 の解法も定数倍によってはこれが取れるようです

小課題3

- ▶ $L \leq 13, Q \leq 10^6$
- ▶ 前計算をビット演算とかで頑張って $O(4^L + Q)$ とかにすると取れるのかもかもしれないが、メモリとかいろいろきつい
- ▶ そろそろ無をやめたい気持ちになってきた

小課題3 の前に.....

- ▶ クエリで与えられる文字列が 0,? のみからなる場合、
どう解けばいいか？
- ▶ アルゴリズムパート開始です



アオダイショウ

小課題3 の前に.....

- ▶ 突然ですがここで有名アルゴリズムの紹介をします
- ▶ その名も高速ゼータ変換
 - ▶ なんか名前がかっこいい
 - ▶ 「名前は知らなかった」という人も若干はいるはず
 - ▶ 「名前も知らなかった」という人はもっといるはず

高速ゼータ変換

- ▶ クエリで与えられる文字列が $0, ?$ のみからなる場合、どう解けばいいか？
- ▶ そのような文字列すべてに対して答えを前計算しておけばよさそう
 - ▶ $O(2^L)$ 通りくらいしかないので全部持っておける
- ▶ ただし、愚直にやると $O(3^L)$ くらいかかる
- ▶ これを $O(2^L L)$ でやるのが高速ゼータ変換

高速ゼータ変換

- ▶ アイデアは至ってシンプルな DP
 - ▶ ただし、直感的にはやや受け入れ難いかも
- ▶ $DP[t][m]$: 下位 t ビットがパターン m に包含され、その他のビットがパターン m と一致するようなビット列に対する値の合計
 - ▶ ビット列 x が y に包含されるとは、 x で立っているビットは y でも立っているという意味

高速ゼータ変換

▶ 以下に例を示す

	000	001	010	011	100	101	110	111
$t = 0$	3	1	4	1	5	9	2	6
$t = 1$								
$t = 2$								
$t = 3$								

高速ゼータ変換

▶ 最下位ビットについて操作

	000	001	010	011	100	101	110	111
$t = 0$	3	1	4	1	5	9	2	6
$t = 1$	3	4	4	5	5	14	2	8
$t = 2$								
$t = 3$								

高速ゼータ変換

▶ 真ん中のビットについて操作

	000	001	010	011	100	101	110	111
$t = 0$	3	1	4	1	5	9	2	6
$t = 1$	3	4	4	5	5	14	2	8
$t = 2$	3	4	7	9	5	14	7	22
$t = 3$								

高速ゼータ変換

▶ 最上位ビットについて操作

	000	001	010	011	100	101	110	111
$t = 0$	3	1	4	1	5	9	2	6
$t = 1$	3	4	4	5	5	14	2	8
$t = 2$	3	4	7	9	5	14	7	22
$t = 3$	3	4	7	9	8	18	11	31

高速ゼータ変換

- ▶ 一番下の行を見ると、求めたいものが求まっている

	000	001	010	011	100	101	110	111
$t = 0$	3	1	4	1	5	9	2	6
$t = 1$	3	4	4	5	5	14	2	8
$t = 2$	3	4	7	9	5	14	7	22
$t = 3$	3	4	7	9	8	18	11	31

小課題3

- ▶ これで、クエリで与えられる文字列が 0,? からなる場合は解けた
- ▶ では、1 もくる場合は？

小課題3

- ▶ 実はこの応用でできる
- ▶ 各ビットを見るときに $0 \rightarrow ?$, $1 \rightarrow ?$ の遷移を行うことにすれば、 $O(3^L L)$ 時間で答えがすべて前計算可能
 - ▶ ただし、オリジナルのアルゴリズムでビット演算を用いた箇所の実装を工夫する必要がある
- ▶ $O(3^L L + Q)$ となり、22 点が得られる

小課題4

- ▶ さて、小課題4 の配点は大きい
 - ▶ 53 点とか書いてあってびっくりする
- ▶ この小課題が、この問題の本質
 - ▶ このページには情報を出し惜しみして期待感を煽る以上の目的はありません
 - ▶ オッ……？

小課題4

- ▶ $L \leq 20, Q \leq 50000$
- ▶ ? が少ないクエリに対しては、愚直に計算する
- ▶ ? が多いクエリに対しては？

小課題4

- ▶ とりあえず高速ゼータ変換をしておく
- ▶ 高速ゼータ変換で得られた情報から、うまく答えを計算できないか？
- ▶ か？

小課題4

- ▶ 高速ゼータ変換で得られた情報から、うまく答えを計算できる
- ▶ メインアイデアは包除原理

小課題4

- ▶ たとえば、列 $011????$ に対する答えは？
- ▶ 列 s に対する答えを $f(s)$ とおけば、これは
 $f(0??????)-f(00?????) - f(0?0?????) + f(000?????)$
として計算できる
 - ▶ 集合の L 個あるベン図みたいなイメージ

小課題4

- ▶ このように、包除原理を用いて答えを求められる
- ▶ 気になる計算量は？

$$H(x) = \begin{cases} 1 & (x > 0) \\ 0 & (x < 0) \end{cases}$$

ヘビサイド関数

小課題4

- ▶ 計算量は？
- ▶ 前処理は高速ゼータ変換のみで、 $O(2^L L)$
- ▶ ? の個数が半分以下のとき、クエリあたり $O(2^{L/2})$
- ▶ ? の個数が半分以上のとき、見るべき $f(s)$ たちの個数は 2 の 1 の個数乗個
- ▶ すなわち、これも $O(2^{L/2})$

小課題4

- ▶ よって、 $O(2^L L + 2^{L/2} Q)$ 時間で解けた
- ▶ 75 点 that 得られる



小課題5

▶ $L \leq 20, Q \leq 10^6$

▶ あとはウイニングラン

小課題5

- ▶ 小課題 4 では 0,? のみからなる文字列に対する答えだけ前計算したが、1,? のみからなる文字列に対しても前計算をしておくことにする
 - ▶ 前計算の計算量はかわらず $O(2^L L)$

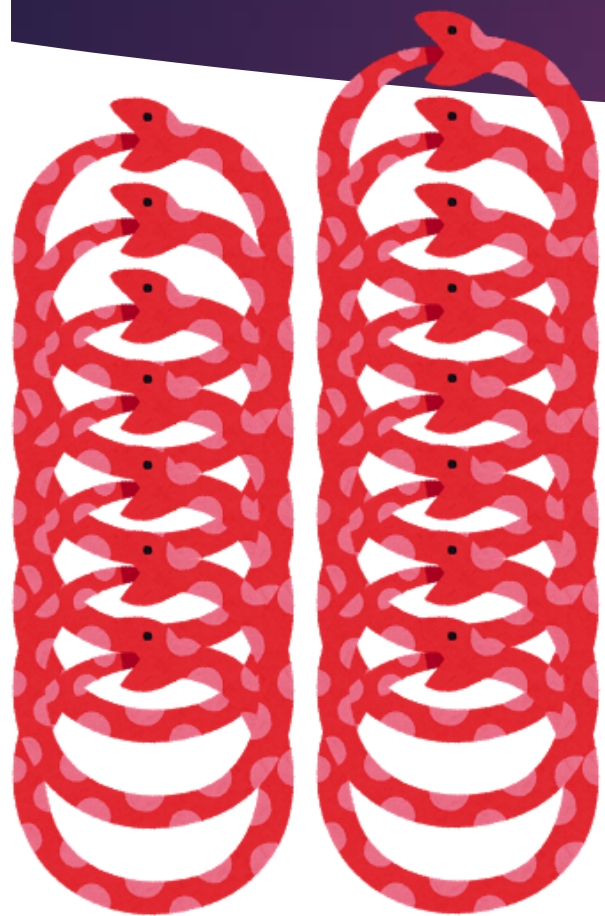
小課題5

- ▶ クエリには以下のように答える
- ▶ $0, 1, ?$ のうち一番少ないものが
- ▶ $?$ の場合、定義通りに計算
- ▶ 0 の場合、 $1, ?$ のみからなる列の答えを用いて包除原理
- ▶ 1 の場合、 $0, ?$ のみからなる列の答えを用いて包除原理

小課題5

- ▶ どの場合も、 $O(2^{L/3})$ 時間でクエリにこたえられる
- ▶ 全体で計算量は $O(2^L L + 2^{L/3} Q)$ となり、満点が得られる

得点分布



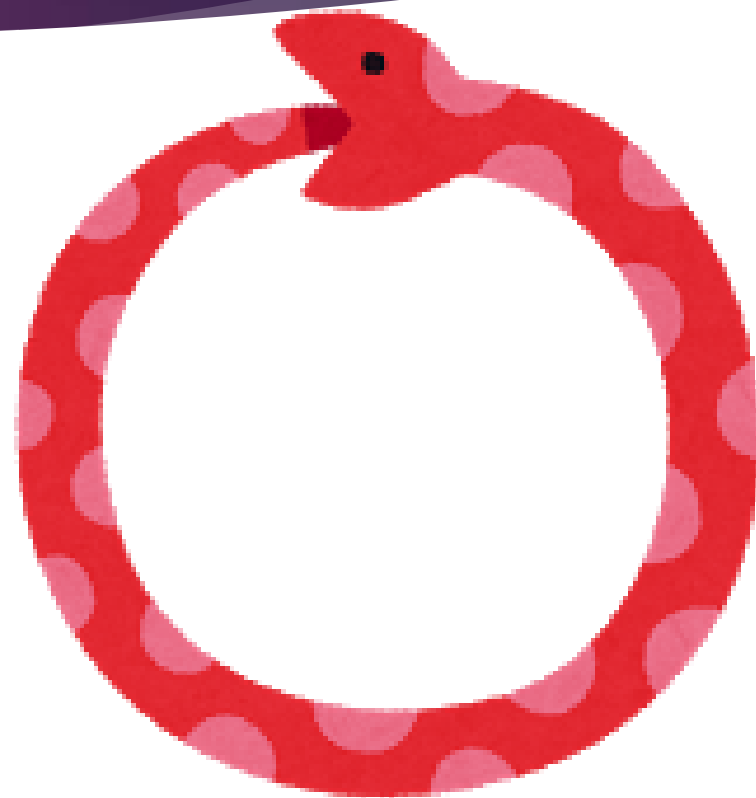
5点



12点



22点



75点