ABC98D Xor Sum 2
解説

https://atcoder.jp/contests/abc098/tasks/arc098_b

目次

1 問題概要

2 ヒント 1

3 ヒント 2

4 ヒント 3

5 解法

問題概要

問題

長さ N の数列 A が与えられる．以下の条件を満たす整数の組 (l, r) $(1 \leq l \leq r \leq N)$ の個数を求めよ．

$$A_l \text{ xor } \cdots \text{ xor } A_r = A_l + \cdots + A_r$$

制約

- $1 \leq N \leq 2 \times 10^5$
- $0 \leq A_i < 2^{20}$

ヒント 1

ヒント

各 l について条件を満たす r の個数を数えましょう.

ヒント1

ヒント

各 l について条件を満たす r の個数を数えましょう.

区間の数え上げでは，l や r で集計することが圧倒的に多いです.
オフライン区間クエリでもこの順で処理することが多いです.

ヒント1

ヒント

各 l について条件を満たす r の個数を数えましょう.

区間の数え上げでは, l や r で集計することが圧倒的に多いです.
オフライン区間クエリでもこの順で処理することが多いです.
区間に関する話題は頻出なので引き出しの数を増やそう！

ヒント 2

ヒント

$a \text{ xor } b \leq a + b$であることを使います．また，等号成立条件も考えてみましょう．

ヒント 2

ヒント

$a \text{ xor } b \leq a + b$ であることを使います．また，等号成立条件も考えてみましょう．

xor の見方

- $\mathbb{F}_2$ ベクトルの足し算（定義）．
- 2 進数表記して位上がりのない足し算.
- 対称差
- $a \text{ xor } b = (a + b) - 2 \times (a \text{ or } b)$

ヒント 2

ヒント

$a \text{ xor } b \le a + b$ であることを使います．また，等号成立条件も考えてみましょう．

xor の見方

- $\mathbb{F}_2$ ベクトルの足し算（定義）．
- 2 進数表記して位上がりのない足し算.
- 対称差
- $a \text{ xor } b = (a + b) - 2 \times (a \text{ or } b)$

今回は 2 個目か一番最後の見方がいいと思います.

xor についての諸知識

演算の性質を書きます.

$$a \text{ xor } b = b \text{ xor } a$$
$$a \text{ xor } (b \text{ xor } c) = (a \text{ xor } b) \text{ xor } c$$
$$a \text{ xor } 0 = 0 \text{ xor } a = a$$
$$a \text{ xor } a = 0$$

以上は通常の足し算と似た性質！(群)
よく，モノイドはセグ木に乗る，というステートメントをみますが，xor はモノイドより強いです（引き算ができる）.

xor についての諸知識

演算の性質を書きます.

$$a \operatorname{xor} b = b \operatorname{xor} a$$
$$a \operatorname{xor} (b \operatorname{xor} c) = (a \operatorname{xor} b) \operatorname{xor} c$$
$$a \operatorname{xor} 0 = 0 \operatorname{xor} a = a$$
$$a \operatorname{xor} a = 0$$

以上は通常の足し算と似た性質！(群)
よく，モノイドはセグ木に乗る，というステートメントをみますが，xor はモノイドより強いです（引き算ができる）.
では，これに対応する掛け算みたいなものはある？

xor についての諸知識

演算の性質を書きます.

$$a \operatorname{xor} b = b \operatorname{xor} a$$

$$a \operatorname{xor} (b \operatorname{xor} c) = (a \operatorname{xor} b) \operatorname{xor} c$$

$$a \operatorname{xor} 0 = 0 \operatorname{xor} a = a$$

$$a \operatorname{xor} a = 0$$

以上は通常の足し算と似た性質！(群)
よく，モノイドはセグ木に乗る，というステートメントをみますが，xor はモノイドより強いです（引き算ができる）.
では，これに対応する掛け算みたいなものはある？
→ cf. Nim product

ヒント 3

ヒント

各 l について条件を満たす r には単調性があります.

単調性

$\Leftrightarrow (l, r)$ が条件を満たすかつ $r' < r \Rightarrow (l, r')$ も条件を満たす.

ヒント 3

ヒント

各 l について条件を満たす r には単調性があります.

単調性

$\Leftrightarrow (l, r)$ が条件を満たすかつ $r' < r \Rightarrow (l, r')$ も条件を満たす.

単調性といえばアイツしかない.

単調性というワードには反応できると思うので，自分で気づけるようになろう！

単調性の証明

本番ではエスパーでもいいですが，ちゃんと根拠も言えるようになりましょう.
今回は考察方針と典型方針があります.

単調性の証明

本番ではエスパーでもいいですが，ちゃんと根拠も言えるようになりましょう.
今回は考察方針と典型方針があります.
・考察方針
　数の 2 進数表記を考えて，立ってる bit に重複があったら xor の方が小さくなる.

単調性の証明

本番ではエスパーでもいいですが，ちゃんと根拠も言えるようになりましょう.
今回は考察方針と典型方針があります.
・考察方針
　数の 2 進数表記を考えて，立ってる bit に重複があったら xor の方が小さくなる.
考察は難しいので典型で処理できるようになろう.

典型方針

・典型方針

(l, r) $(l < r)$ が条件が満たすとき，$(l, r-1)$ も条件をみたすことを示せばよい（この考え方も典型）．

先ほどの式 $a \text{ xor } b \leq a + b$ を考える．

典型方針

- 典型方針

(l, r) $(l < r)$ が条件が満たすとき，$(l, r-1)$ も条件をみたすことを示せばよい（この考え方も典型）.
先ほどの式 $a \text{ xor } b \leq a + b$ を考える.
これを繰り返し用いることで,

$$A_l \text{ xor } \cdots \text{ xor } A_{r-1} \leq A_l + \cdots + A_{r-1}$$

を得る.

典型方針

・典型方針

(l, r) $(l < r)$ が条件が満たすとき，$(l, r-1)$ も条件をみたすことを示せばよい（この考え方も典型）.

先ほどの式 $a \text{ xor } b \leq a + b$ を考える.

これを繰り返し用いることで,

$$A_l \text{ xor } \cdots \text{ xor } A_{r-1} \leq A_l + \cdots + A_{r-1}$$

を得る．もし，等号でなかったとすると,

$$\begin{aligned}(A_l \text{ xor } \cdots \text{ xor } A_{r-1}) \text{ xor } A_r &\leq (A_l \text{ xor } \cdots \text{ xor } A_{r-1}) + A_r \\ &< A_l + \cdots + A_{r-1} + A_r\end{aligned}$$

となるが，これは (左辺) $= A_l + \cdots + A_r$ に矛盾.

典型方針の考え方

この方針は xor の性質のうち，$a \text{ xor } b \leq a + b$ のみを用いている．
実際の演算の振る舞い（bit がどうなるとか）を考える必要が全くない！

典型方針の考え方

この方針は xor の性質のうち，$a \text{ xor } b \le a + b$ のみを用いている．
実際の演算の振る舞い（bit がどうなるとか）を考える必要が全くない！
このような考え方を知っておくと，

- 一般化しやすい
- 頭がバグらない（超重要）

典型方針の考え方

この方針は xor の性質のうち，$a \text{ xor } b \leq a + b$ のみを用いている．
実際の演算の振る舞い（bit がどうなるとか）を考える必要が全くない！
このような考え方を知っておくと，

- 一般化しやすい
- 頭がバグらない（超重要）

普段の練習から抽象化できる部分を意識的に見つけるようにすると，練習の効率が大幅に上がると思います．

解法

各 l について条件を満たす r の個数を求める.

解法

各 l について条件を満たす r の個数を求める．
l を固定して二分探索すると全体で $O(N \log N)$ 時間．区間 xor や区間和は，前スライドで説明したようにどちらも演算が群（引き算ができる）なので累積和などでよい．

解法

各 l について条件を満たす r の個数を求める．
l を固定して二分探索すると全体で $O(N \log N)$ 時間．区間 xor や区間和は，前スライドで説明したようにどちらも演算が群（引き算ができる）なので累積和などでよい．
単調性から尺取り法で l の昇順に条件を満たす r の最大値を求めることもできる．この場合全体で $O(N)$ 時間．