

**A+B**

原案:生田、寺尾

解説:生田

テスター:秋葉、林、生田

# 問題概要

- 二進数A,Bがある、以下のクエリを処理せよ
  - $A \leq x < A+B$  となるxで2進数表示した時の1の個数の最大値を出力
    - $A=10000, B=00101$  の場合は  $x=10011$  の3個が最大
  - Aの指定ビットを反転
    - $A=10000$  の(下位から) 0ビット目を反転
    - $\rightarrow A=10001$
  - Bの指定ビットを反転
    - $B=00101$  の(下位から) 1ビット目を反転
    - $\rightarrow B=00111$
- クエリ個数、A、Bの長さは  $3 \cdot 10^5$  以下

# 解法1

- AとBを陽に管理する
- Qクエリを処理することを考える
- Bの最も大きい立っているビットを見つける。

○ A .....1.....

○ B 0...01.....

○                   ↑ココ

# 解法1続き

- A .....1.....
- B 0...01.....
- ↑ココの位置をiとする
- A[i]が1の時
  - A .....1.....
  - B .....01.....
  - というAとBを考えると $x = (\text{Aの上位})11111$ という形が答えの候補となる。
  - 繰り上がりしてもせいぜい上の $x$ と同じビット数になる
- Bの1が立っている位置をsetで持ち、Aを01の数列と考えた累積和のBITを用いることで答えられる。

# 解法1続き

- A .....0.....
- B 0...01.....
- ↑ココの位置を*i*とする
- A[*i*]が0の時

- A .....0.....

- B 0...01.....

というAとBを考えると $x = (\text{Aの上位})01111$ という形が  
答えの候補となる。

- 問題は上の $x$ の*i*ビット目を1に出来るかどうか
- これは*i*より下のビットから繰り上がりがあるかどうかで判定する。
- AとBで同時に0になっている場所、同時に1になっている場所をsetで持つことでキャリーが到達して繰り上がるか

## 解法1続き

- 変更のクエリはsetをいじるだけで可能
- 質問のクエリはlower\_bound等をうまく使う

## 解法2

- segtreeでも可能
  - $A + B$ と $A \text{ xor } B$ を管理し、1の個数とmsb,lsbの位置を管理。
  - 木のジャッジ解
  - $O(n \log n)$ で書けます。

# ジャッジ解

- 生田、C++ 101行
- 林、C++ 150行
- 秋葉、C++ 158行



# 統計

- First AC (184:56)
  - (  $\omega$  )
- AC/Try/Total
  - 2/4/11