

# EDITORIAL

## Codeforces:-MEX Queries(817F)

[Problem Link](#)

**Prerequisite** - Segment tree with lazy propagation, Binary search, STL

Firstly, let's notice that the queries are offline. So we can compress the numbers by taking  $L$  and  $R + 1$  of each query. **MEX will be either one of these numbers or 1** (this is the only thing in problem) . So now we have numbers up to  $2 \cdot 10^5$  and pretty basic task on segment tree.

The first two types of queries are translated to "assign value 1 or 0 on a segment" (set the number on some position is either present or not). The third is "for each  $i$  in segment  $[l, r]$  assign  $x_i$  to  $x_i \oplus 1$ " (this will inverse the segment as described in statement).

**Segment tree should keep sum of the segment in its nodes.** XOR/FLIP on segment will turn  $val$  into  $len - val$ ,  $len$  is the length of the segment being covered by the node.

The leftmost zero cell is MEX.

While standing in some node  $v$ , check if its left son is full (has 1 in every cell of the segment, like  $t[node * 2] = mid - left$  if you use 1-indexed tree and intervals for it). If it is full then go down to the right son, otherwise there

exists some zero cell in a segment of the left child and you should go down to it.

You should use lazy propagation to guarantee  $\log(n)$  per query.

Overall complexity:  $O(q \cdot \log(n))$

Although ,implementation of problem is quite complicated but use of simple functions for each task and using two different arrays for lazy and flipping ,it would be easy to handle .

For better clarification ,go through the code.

**Author's Code** :- [coded by enigma27](#)