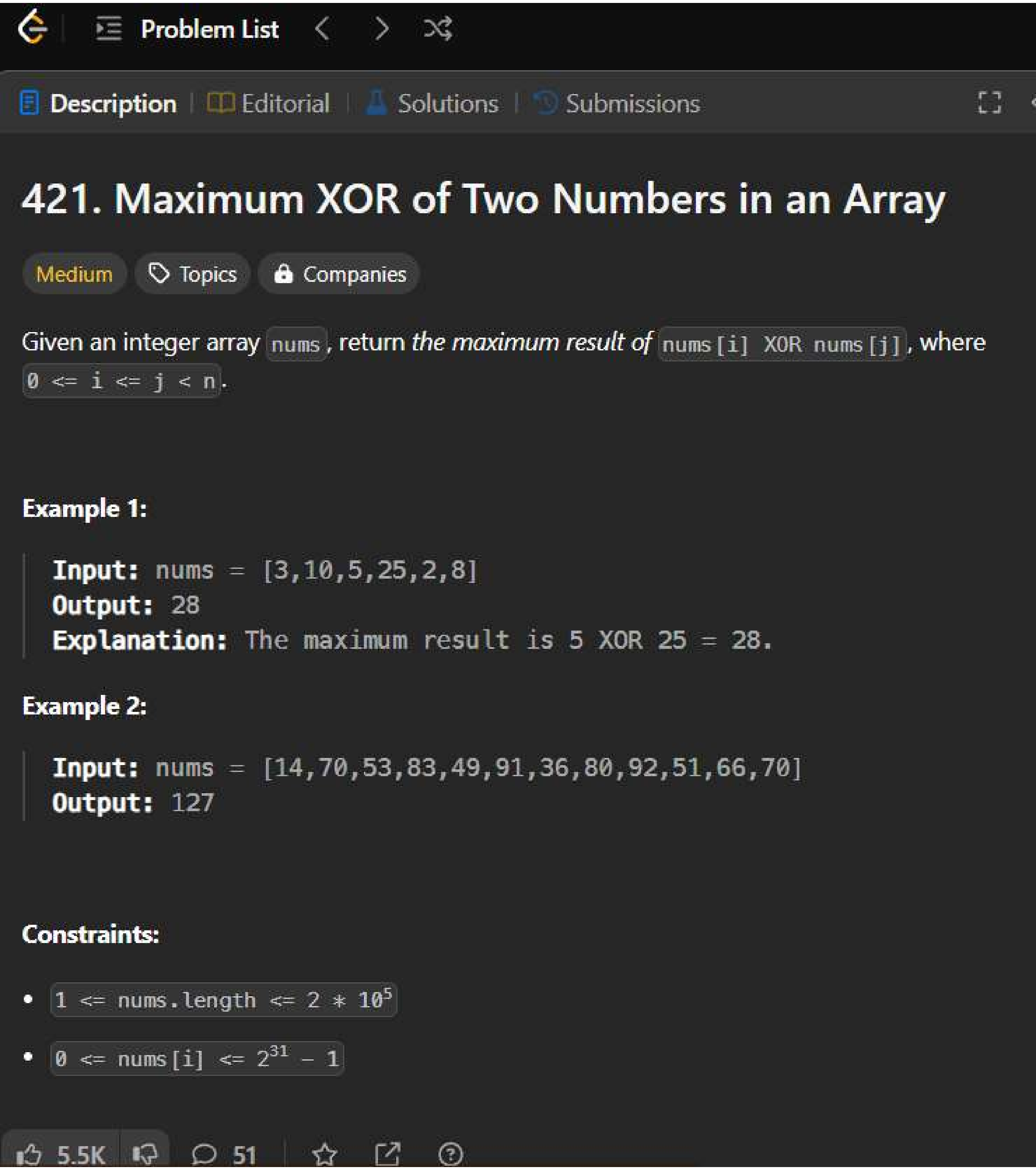

421. Maximum XOR of Two Numbers in an Array

Given an integer array `nums`, return *the maximum result of* `nums[i] XOR nums[j]`, where `0 <= i <= j < n`.

Example 1:

```
Input: nums = [3,10,5,25,2,8]
Output: 28
Explanation: The maximum result is 5 XOR 25 = 28.
```

Example 2:

```
Input: nums = [14,70,53,83,49,91,36,80,92,51,66,70]
Output: 127
```

Constraints:

- `1 <= nums.length <=` $2 * 10^5$
- `0 <= nums[i] <=` $2^{31} - 1$

```
// LEETCODE 421 SOLUTION
#include<bits/stdc++.h>
using namespace std;
class Node{
    private:
    Node* links[2];
    public:
    bool have(int bit){
        return links[bit] != NULL;
    }

    void put(int bit,Node* node){
        links[bit]=node;
    }

    Node* get(int bit){
        return links[bit];
    }
};
class Trie{
   Node* root;
   public:
   Trie(){
      root=new Node();
   }

   void insert(int num){
       Node* curr=root;
       for(int i=31; i>=0; i--){
            int bit=(num>>i)&1;
            if(!curr->have(bit)){
                curr->put(bit,new Node());
            }
            curr=curr->get(bit);
       }
   }

    int getmax(int num){
         Node* curr=root;
         int ans=0; // 000...000
         for(int i=31; i>=0; i--){
            int bit=(num>>i)&1;
            if(curr->have(!bit)){
                ans=ans|(1<<i);
               curr=curr->get(!bit);
            }else{
                curr=curr->get(bit);
            }
         }
         return ans;
    }
};
class Solution {
public:
    int findMaximumXOR(vector<int>& nums) {
          Trie trie;
          for(auto i:nums){
            trie.insert(i);
          }
          int maxi=0;
          for(auto i:nums){
             maxi=max(maxi,trie.getmax(i));
          }
          return maxi;
    }
};
```