

← Notes

Gaussian Elimination for XOR Maximization

21

Gaussian

xor-maximization

Originally posted on my blog.

Gaussian Elimination is a technique traditionally used to solve linear quations, finding determinant, rank of matrix, inverse of matrix. If you don't remember it, have look at this video.

Today we will be applying the same technique to solve a problem. Wasting no time, lets proceed.

The problem is to find the subset of no.s from given set so as to get maximum value if all no.s in subset are EX-ORed. This problem is known with id XMAX on SPOJ. The variant of this problem was there on CodeChef with name XOR with Subset.

Here, I will try to give you a step by step method to solve the problem:

- 1. The idea here is to choose a no. which has MSB with maximum value. With a little thinking, we will get the max(array) will have this feature. Say, maximum of array is a k-bit no. and that no. is M. Now, there can be multiple no.s which are k-bit no.s and have the same highest-value bit as '1'. So in that case we will EX-OR each of them with the M and put them again input array (ideally, we can choose any one of those k-bit no.s as M and put others in array after ex-oring with it). At the same time we will keep the no. M in some other array say x[].
- 2. We will keep doing step 1 until we get 0 as maximum in the array i.e. this loop will run for iterations = no. of bit the maximum of array is.
- 3. Now will have a x[] array which will be of size = no. of bit the maximum of array is
- 4. Now we will initialize the answer variable to given 0 and loop for each value in array x[], if value of answer variable is going to increase with the inclusion of ith element, we will update the answer variable with the new value as answer ^ x[i], else we will keep it as is.
- 5. Finally returning the value of answer solves our problem.

This solves our problem and we can get it AC on SPOJ.

Now the change in CodeChef problem is the compulsion of including the given no. and then finding the subset which gives the maximum value.

Nothing to worry about.

Simply we need to initialize answer variable to the given no. instead of a 0 (as we do in step 4) and, solution gets AC on CodeChef too.

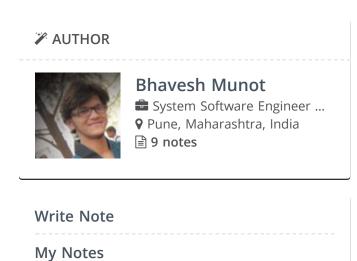
The main logic here is to try making every bit of answer to be '1'. Our x[] will have elements having different set bits position so when we ex-or them with answer variable, answer variable will have more no. of set bits.

I hope I am successful in giving you the procedure. Have look at the code for more clarification.

```
// Solution to problem on Codechef and SPOJ
#include <iostream>
using namespace std;
#include <algorithm>
#include <vector>
#define ull unsigned long long
#define testcases() int var; cin>>var; while(var--)
int length(ull n)
{
    int cnt = 0;
    while(n)
    {
         cnt++;
         n>>=1;
    return cnt;
}
int main()
{
    testcases()
    {
         int n, k;
         cin>>n>>k;
         // input array --> a[]
         ull a[n];
         for(int i=0; i<n; i++)</pre>
```

```
cin>>a[i];
// store how many bit is the ith number (bit-length) -->
int lengths[n];
for(int i=0; i<n; i++)
    lengths[i] = length(a[i]);
// put all numbers with same bit-length in one bucket -->
vector<ull> buckets[65];
for(int i=0; i<n; i++)
    buckets[lengths[i]].push_back(a[i]);
// this array will have a number from each bucket --> mod
ull modified array[100], m index = 0;
for(int i=64; i>0; i--) //since long long is 8 byte = 64
{
    if(buckets[i].size()) //if there exist at least one \epsilon
    {
         // put first value from bucket in modified array
         modified array[m index++]=buckets[i][0];
         // and put remaining elements from same bucket a
         // buckets after ex-oring
         for(int j=1; j<buckets[i].size(); j++)</pre>
         {
              ull temp = buckets[i][0] ^ buckets[i][j];
              int len = length(temp);
              buckets[len].push back(temp);
         }
    }
}
// Step 4 according to the explained procedure
// make ans = k for codechef and ans = 0 for spoj :)
ull ans = k;
for(int i=0; i<m index; i++)</pre>
    if(ans < (ans ^ modified_array[i]))</pre>
         ans = (ans ^ modified array[i]);
// Step 5 according to the explained procedure
cout<<ans<<endl;</pre>
```

```
}
return 0;
}
Like { 0  Tweet { 0  G+1  0 }
```



Drafts

TRENDING NOTES

Strings And String Functions
written by Vinay Singh

Segment Tree and Lazy Propagation
written by Akash Sharma

Number Theory - II
written by Tanmay Chaudhari

Matrix exponentiation
written by Mike Koltsov

Graph Theory - Part II
written by Pawel Kacprzak

more ...

ABOUT US HACKEREARTH DEVELOPERS

Blog API AMA

Gaussian Elimination for XOR Maximization | HackerEarth

Engineering Blog

Updates & Releases

Team

Careers

In the Press

Chrome Extension

CodeTable

HackerEarth Academy

Developer Profile

Resume

Campus Ambassadors

Get Me Hired

Privacy

Terms of Service

Code Monk

Judge Environment

Solution Guide

Problem Setter Guide

Practice Problems

HackerEarth Challenges

College Challenges

RECRUIT

Developer Sourcing

Lateral Hiring

Campus Hiring

FAQs

Customers

Annual Report

REACH US



Illrd Floor, Salarpuria Business Center, 4th B Cross Road, 5th A Block, Koramangala Industrial Layout, Bangalore, Karnataka 560095, India.

contact@hackerearth.com

+91-80-4155-4695

+1-650-461-4192









© 2015 HackerEarth