Max XOR in Queries

Problem Statement Suggest Edit

You are given an array/list 'ARR' consisting of 'N' non-negative integers. You are also given a list 'QUERIES' consisting of 'M' queries, where the 'i-th' query is a list/array of two non-negative integers 'Xi', 'Ai', i.e 'QUERIES[i]' = ['Xi', 'Ai'].
The answer to the ith query, i.e 'QUERIES[i]' is the maximum bitwise xor value of 'Xi' with any integer less than or equal to 'Ai' in 'ARR'.
You should return an array/list consisting of 'N' integers where the 'i-th' integer is the answer of 'QUERIES[i]'.

arr[] : { 0, 2, 1, 3, 4 }

queries[] : Xi Ai
{ 1, 3 }
{ 5, 6 }

Xi XOR arr[1 ... n]

Ai <= arr[1 ... n]

queries = { { 1, 3 }

{ 5, 6 }

x_i	A_i
1	3
5	6

findmax (x)

arr[.. .] XOR X

{ 0, 2, 1, 3, 4 }

A_i - 3 x_i

0 ^ 1

2 ^ 1

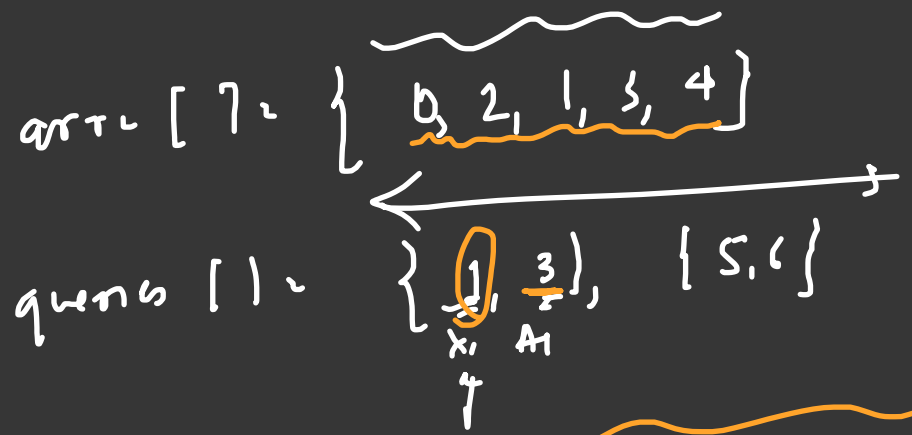
1 * 1

3 ^ 1

} max

queries = [? , ?]

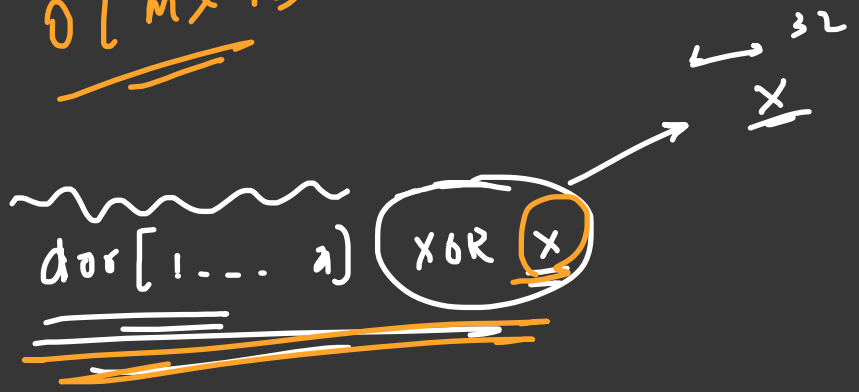
Brute force:




```
for ( i=0; i<queries.size(); i++)
    xi = query[i][0],
    ai = qur[i][1],
    for (j=0; j<n; j++)
        if (arr[j] < ai)
            maxXor = max (maxXor, xi ^ arr[j]),
return maxXor;
```

T.C. = O(M x N)

Tries:



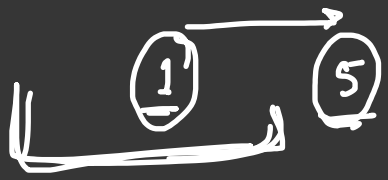
O(N x 32) ≈ O(N)

The insert arr → find max

We insert only that element in the trie that are less than ai

arr[]: { 1, 3, 2, 5, 4 } → sort(): { 1, 2, 3, 4, 5 }

Trie



xi ai

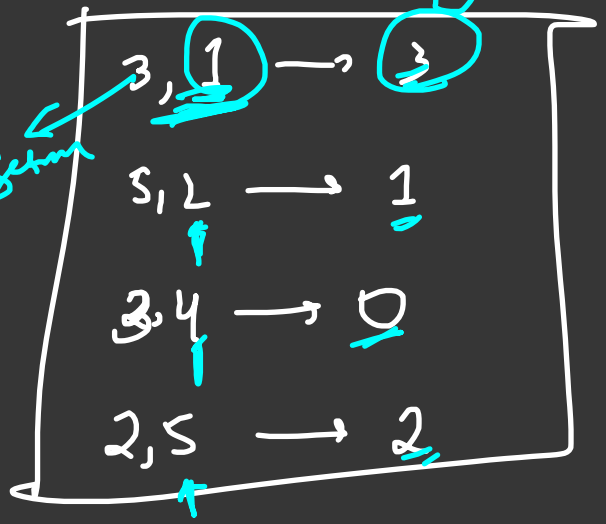
queries[]: { 3, 4 }

getmex!

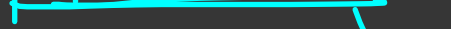
{ 5, 2 } 1

{ 2, 5 } 2

{ 3, 1 } 3

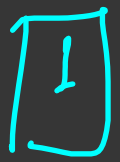



```
if (arr[i] < qi)
Trie.insert( )
```




```
findmax(3)
```

for



- Insert only element which are less than ai in Trie to do binary search
- sort the query array according ai and sort arr.

Offline query

So, we have given query

{1, 5}
{5, 6}
{5, 4}

→ query

We create a similar query we some extra nodes

{1, 5, 0}
{5, 6, 0}
{3, 4, 2}

→ Offline query.