Editorial:

We can construct two arrays lbest and rbest for the left hand side of the expression and for the right hand side of the expression.
To calculate lbest[i] we will iterate m from i to i such that (arr[m] ^ arr[m+1] ^ arr[m+2] …. arr[i]) is maximised. Then lbest[i] = Max(lbest[i-1], val). Similar can be done for rbest. Now we will calculate val. Let C[i] = (arr[1] ^ arr[2] ^ arr[3] …. arr[i]). For some j <= i, C[j-1] ^ C[i] = (arr[j] ^ arr[j+1] ^ arr[j+2] … arr[i]). We can now say that lbest[i] = max(lbest[i−1],val) where val = maximum of (C[j−1]C[i]) for j = 1 to i. Now using Tries we can easily calculate ‘val’ in (log arr(max)). We can store the bits of each number in the nodes of trie and iterate through the trie such that we can get the maximum xor possible.