1. store the numbers in a Trie. for each number i, query $\max_j a[j]\ xor\ a[i]$ takes $O(w)$, by walking down the Trie. total time $O(nw)$.
2. determine the result bit by bit. We can verify whether we can get an xor result with prefix t in $O(n)$ by hashing. total time $O(nw)$.

3. w-ary Trie with depth $O(\frac{w}{\log w})$, use word operations to walk down. $O(\frac{nw}{\log w})$.
The space complexity can be reduced to $O(\frac{nw}{\log w})$ without using hashing (the naïve implementation needs $O(\frac{nw^2}{\log w})$ space), by 2-level indexing technique. So this algorithm is deterministic.

Could you do this in $O(n)$ runtime?
No I can't.

References