EDITORIAL : Code LNM 21.0

A. Let's Play

If the number is prime , Alice would have only one option to play and that would update N to 1 , hence Bob will win , otherwise it is easy to say that Alice would divide N by K such that (N/K) is a prime number and hence Alice would win.

Code : https://ideone.com/RP8OT2

B. XOR

If we take an integer temp representing the XOR of the whole array. And If the count of A in the array is an even number of times then we know that the XOR of the same elements is zero. then the contribution of all A in Whole XOR is 0. and by converting all A to B the parity of B remains the same. so XOR of the whole array remains the same which is *temp*.
If the count of A is an odd number then the contribution of all A in XOR of the whole array is A so XOR after removing all A from the array became:
For XOR:
X=Y^Z => Y=X^Z from this we can say that
temp=A^(remaining) => remaining =A^temp
and by Converting all A to B parity of B gets changed then the XOR of the whole array becomes *A^temp^b*.

Code: https://ideone.com/mmzcWY

C. Goldmines

They can take that gold in 2 ways.
First is when they both take single gold.

Second is when any one of them takes both gold(If possible). So by using BFS we were able to find these possibilities(If possible) and the time taken by them. And the answer is a minimum of all 4 possible paths.

Code : https://ideone.com/jxM7I1

D. Make-it-Tree

The key property to observe in this problem is that for every node we need to find the total number of ways to select some node for each child of that node separately, which satisfy the property of tree structure and simply take the product of all the values that received from the childs of that node and continuously add them to the final answer that should be the output.

This can be easily achieved by the modified dfs.

Code : https://ideone.com/PbW6xs

E. K-th smallest

Let us modify our initial array in such a way that if we for each element in the initial array we will put 1 at that index .
Now suppose we have to find the Kth element then our answer will be that minimum index (x) such that the sum of A[i] upto x is equal to "K" .
Now basically we have to do two tasks
1) find prefix sum upto index "i" efficiently
2) update the array A after the query.

We can use binary search to find index i in log n time and then update the array for each query.

Code : https://ideone.com/vygYWN

F. Paint Matrix

The key property to observe in this problem is that the answer is dynamic in nature. If we increase the size of given array i.e. n then we don’t need to compute from start again instead we should use the value of previous states for further operations.

We have to do the same as a dp approach to count. We can define the state with respect to the size of the array and maximum number of cells we need to paint. Note that we can not paint cells more than 1.5*n+1 in any condition. We required to visit each dp state and find the maximum number of ways to paint the cells till there from 0 to maximum number of paint cells allowed and iterate the same procedure for each size of the array from 1 to n.

Code : https://ideone.com/Up39gb