Xor Thingy <July Circuits '19> - Hackerearth

Tuesday, May 19, 2020   7:50 PM

There are n players playing a game. Every player has a skill value denoted by Si. Before the game starts, each player can change their skill value to any non-negative integer smaller than their current skill value or leave it untouched. The coolness of the game is then defined as the bitwise-xor of the players skill values. We'd like to know for every number X from 0 to m the number of ways that the coolness of the game equals X. Since these numbers may be extremely large, output them modulo 10^9+7.

Input
The first line of input contains two integers n and m, the number of the players and the parameter from the problem statement.
The second line of input contains n space-separated integers, describing the skill values of the players.
1≤n,m,Si≤500

Output
Output m+1 integers, the i-th of which is the number of ways that the coolness of the game equals i−1 modulo 109+7.
Sample 2 Input
5 10
1 6 12 4 8
Sample 2 output
606 606 606 606 602 602 601 601 420 420 420

SAMPLE INPUT

3 5
1 3 2
SAMPLE OUTPUT

6 6 6 6 0 0

The Code:

```
#include <stdio.h>
#define MOD 1000000007
int main(){
    int N, M, element, xor, max, curr, prev, i, j, k;
    long int** counts = (long int**)malloc(2 * sizeof(long int*));
    counts[0] = (long int*)calloc(512, sizeof(long int));
    counts[1] = (long int*)calloc(512, sizeof(long int));
    scanf("%d %d", &N, &M);
    counts[0][0] = 1;
    for(k=0; k<N; k++) {
        scanf("%d", &element);
        curr = 1;
        prev = 0;
        for(i=0; i<=element; i++) {
            for(j=0; j<512; j++) {

                xor = i ^ j;
                if(counts[prev][j] > 0) { // to check if that sum exists, only then we consider it's xor. It's not really necessary, cause we'll just be adding zero, but it's good for sense and readability.
                    counts[curr][xor] += counts[prev][j];
                }
                if(counts[curr][xor] >= MOD) {
                    counts[curr][xor] -= MOD;
                }
            }
        }
        for(i=0; i<512; i++) {
            counts[prev][i] = counts[curr][i];
            counts[curr][i] = 0;
        }
    }
    for(i=0; i<=M; i++) {
        printf("%d ", counts[0][i]);
    }
}
```

The Stats:

Score
30.0

Time (sec)
5.0933

Memory (KiB)
64

Language
C