

Statistics

Problem Statement

Problem Statement for BagAndCards

Problem Statement

Fox Ciel has n bags, numbered 0 through $n-1$. Each bag contains lots of cards, each with a single number written on it. For each i and j , bag i contains exactly $\text{count}[i][j]$ cards with number j on them.

You are given the ints m , x , a , b , and c . The values $\text{count}[i][j]$ will be generated from these variables using the pseudocode shown below. (Watch out for integer overflow.)

```
for i = 0 .. n-1:
    for j = 0 .. m-1:
        count[i][j] = x;
        x = ((x * a + b) xor c) modulo 1,000,000,007.
```

All other values $\text{count}[i][j]$ not explicitly initialized in the pseudocode are zero. Hence, each card in each bag contains a number between 0 and $m-1$, inclusive.

According to Fox Ciel, some numbers are good numbers. You are given this information encoded as the String **isGood** with $2*m-1$ characters. For each k , the number k is good if and only if **isGood**[k] is 'Y'.

Let i and j ($i < j$) be any two bags. The value $\text{ans}[i][j]$ is defined as the number of ways in which Ciel can select one card from bag i and one card from bag j in such a way that their sum is a good number.

Compute all values $\text{ans}[i][j]$. Then, in order to keep the return value small, compute and return the hash of these values. More precisely, compute and return the bitwise xor of all values $(\text{ans}[i][j] \text{ modulo } 1,000,000,007)$.

Definition

Class: BagAndCards
 Method: getHash
 Parameters: int, int, int, int, int, int, String
 Returns: int
 Method signature: int getHash(int n, int m, int x, int a, int b, int c, String isGood)
 (be sure your method is public)

Notes

- Pay attention to the unusual time limit.

Constraints

- n will be between 2 and 500, inclusive.
- m will be between 2 and 500, inclusive.
- x will be between 0 and 1,000,000,000, inclusive.
- a will be between 0 and 1,000,000,000, inclusive.
- b will be between 0 and 1,000,000,000, inclusive.
- c will be between 0 and 1,000,000,000, inclusive.
- **isGood** will contain exactly $(2*m-1)$ elements.
- Each element in **isGood** will be 'Y' or 'N'.

Examples

```
0)
 2
 4
 1
 1
 0
```

```

0
"NNYYNYN"
Returns: 9
We have two bags. Each of these bags contains the following cards: {0,1,2,3}. That is, for
each i between 0 and 1 inclusive, and for each j between 0 and 3 inclusive, we have count[i]
[j] = 1.

The good numbers are 2, 3, and 5. There are 9 ways to get a good number:

ways to get sum 2: 0+2, 1+1, 2+0
ways to get sum 3: 0+3, 1+2, 2+1, 3+0
ways to get sum 5: 2+3 ,3+2
1)
3
5
1
1
1
2
"NNYYNYNYN"
Returns: 1532
count will be:
{{1, 0, 3, 6, 5},
 {4, 7, 10, 9, 8},
 {11, 14, 13, 12, 15}}
And the answers will be:
ans[0][1] = 291
ans[0][2] = 500
ans[1][2] = 1323
So you should return 291^500^1323 = 1532
2)
10
20
111
222
333
444
"NNNNNNYYNNNNYYYYYYNNYYYYNNNNNNYYNNNNYYN"
Returns: 450750683
3)
2
2
1
1
0
0
"NNY"
Returns: 1

```

This problem statement is the exclusive and proprietary property of TopCoder, Inc. Any unauthorized use or reproduction of this information without the prior written consent of TopCoder, Inc. is strictly prohibited. (c)2010, TopCoder, Inc. All rights reserved.