

Oppo Codes

Mighty Battle

0 / 12 Completed

Question 11

01:58:51

Max. score: 50.00

Mighty Battle <Recruit RFU>

You are given two Arrays $A[]$ and $B[]$ of length N . Following operations are performed on the arrays:

1. For i^{th} iteration , take two numbers X and Y .
2. X equals $A[i]$ or **xor** of $A[i]$ with any of its previously lost elements.
3. Y equals $B[i]$ or **xor** of $B[i]$ with any of its previously lost elements.
4. If $X > Y$, array B loses its i^{th} element.
5. If $X < Y$, array A loses its i^{th} element.
6. If $X = Y$, both elements live.

Print the count of lost elements of array B after N operations.

```
def max_xor(a,arr):  
    if len(arr)==0:  
        return a  
    else:  
        m=arr[0]  
        for i in range(len(arr)):  
            c=a^arr[i]  
            if m<=c:  
                m=c  
        return m  
def solver(N,A,B):  
    A_dead=[]  
    B_dead=[]  
    for i in range(N):  
        X=max_xor(A[i],A_dead)  
        Y=max_xor(B[i],B_dead)
```

```

if X>Y:
    # print(B)
    b=B[i]
    B_dead.append(b)
elif X<Y:
    # print(A)
    a=A[i]
    A_dead.append(a)
return len(B)-len(B_dead)

```

Nine's Complement

The screenshot shows a programming competition interface with the following details:

- Header:** 0 / 12 Completed, Question 12, 01:57:22, Max. score: 100.00.
- Title:** Nine's Complement
- Description:** Given an array of **N** numbers **nums** and **Q** queries to perform, queries are one of two types :

 - 1 | r s e means performing nine's complement for the $[e_{th}, \dots, s+1_{th}, s_{th}]$ digits of $[num_{s_1}, num_{s_1+1}, \dots, num_{s_e}]$.
 - 2 | r s e means printing the sum of the $[e_{th}, \dots, s+1_{th}, s_{th}]$ digits of $[num_{s_1}, num_{s_1+1}, \dots, num_{s_e}]$.

- Example:** Consider $N = 5$, $Q=1$, $A = [1, 2, 3, 4, 5]$, $k = 2$, $\text{Queries} = [[2, 0, 2, 0, 1]]$. we get the sum of digit of positions 0 and 1 for the first three elements, which is 6. so, answer would be [6].
- Function description:** Complete the `solve` function provided in the editor. This function takes the following 4 parameters and returns array of integers representing the answer of the queries.
 - N:** Represents the size of given array.
 - Q:** Represents the number of queries.
 - nums:** Represents the given array.
 - ...:** Represents each query.

```

def nine_complement(N,s,e,number):
    if len(number)<N:
        while True:
            number='0'+number
            if len(number)==N:
                break
    for i in range(N-1-e,N-s):

```

```

if(number[i] != '.'):
    a = 9 - int(number[i])
    number = (number[:i] +
              str(a) + number[i+1:])
return str(int(number))
def adder(N,s,e,number):
    if len(number)<N:
        while True:
            number='0'+number
            if len(number)==N:
                break
    s=0
    for i in range(N-1-e,N-s):
        if(number[i] != '.'):
            a = int(number[i])
            s=s+a
    return s
for q in queries:
    type=q[0]
    l=q[1]
    r=q[2]
    s=q[3]
    e=q[4]
    if type==1:
        temp_nums=arr[l:r+1]
        arr[l:r+1]=[nine_complement(N,s,e,t) for t in temp_nums]
    if type==2:
        temp_nums=arr[l:r+1]
        s=0
        for t in temp_nums:
            s=s+adder(N,s,e,t)
        print(s)
##Driver Code Test Case from Chegg
arr=[616,3922,1185,560]
queries=[[1,0,2,2,3],[1,1,2,1,2],[2,0,3,0,3],[1,0,2,0,2],[2,0,0,0,1]]
arr=list(map(str,arr))
N=4
Result Obtained
['9316', '6022', '8885', '560']
['9316', '6972', '8115', '560']
69
['9683', '6027', '8884', '560']
11

```

Result test case from chegg

- After the first query, the array would be : [9316, 6022, 8885, 560].
- After the second query, the array would be : [9316, 6972, 8115, 560].
- The answer to the third query would be the sum of all digits of the current array which is 69.
- After the fourth query, the array would be : [9683, 6027, 8885, 560].
- The answer to the fifth and last query would be the sum of the last two digits of the first number which is $3 + 8 = 11$.

Numbers and Cards

01.05.15

Question 11 Max. score: 100.00

Numbers and cards

You are given 10 types of cards. Every card has a specific number written on it from 0 to 9. There are K cards for each number. You have a total of $10^9 K$ cards.

Task

Select N cards from the provided $10^9 K$ cards and determine the number of different sets of N cards that can be generated.

The size of these sets must be N . If the order of the cards is different in two sets, then these sets are considered equal.

Note: Since the output can be large, print output modulo $(10^9 + 7)$.

Example

Assumptions

- $K = 1$
- $N = 2$

Approach

You have total 10 cards each of 0 to 9. You have to select a total of 2 cards. So there are $(10 \cdot 9)/2 = 45$ ways.

Hence, the answer is 45.

Example

Assumptions

- $K = 1$
- $N = 2$

Approach

You have total 10 cards each of 0 to 9. You have to select a total of 2 cards. So there are $(10 \cdot 9)/2 = 45$ ways.

Hence, the answer is 45.

Function description

Complete the `solve` function provided in the editor. This function takes the following 2 parameters and returns the count of ways.

- N : Represents the number of cards to be selected
- K : Represents the count of each type of card

Input format

Note: This is the input format that you must use to provide custom input (available above the **Compile** and **Test** button).

- The first line contains two space-separated integers N and K .

Output format

Print the answer representing the number of different sets of N cards *modulo* $(10^9 + 7)$.

Note: This is the input format that you must use to provide custom input (available above the **Com** and **Test** button).

- The first line contains two space-separated integers N and K .

Output format

Print the answer representing the number of different sets of N cards *modulo* ($10^9 + 7$).

Constraints

$1 \leq N, K \leq 10^9$

$N \leq 10 \times K$

Code snippets (also called starter code/b boilerplate code)

This question has code snippets for C, CPP, Java, and Python.

Sample input 1

[Copy](#)

Sample output 1

18 1

1

Explanation