String Transmission

Group Members:

Aimal Wajihuddin
Zach Saegesser
Ryan Edelstein

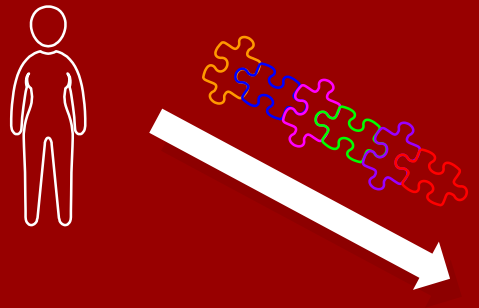
WHAT WAS THE PROBLEM

Alice

Alice and Bob are trying to communicate

Bob

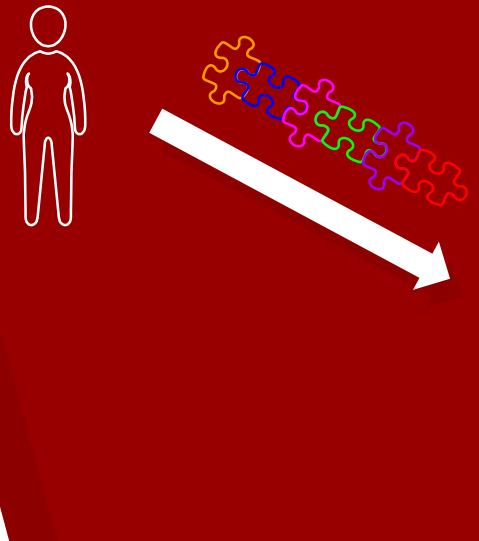
Alice



Alice sends a message

Bob

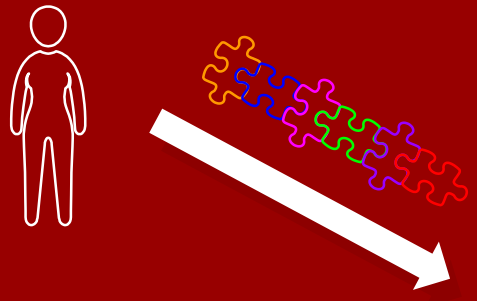
Alice

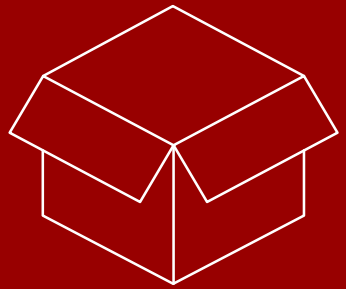


Bob

The message is transmitted...

Alice

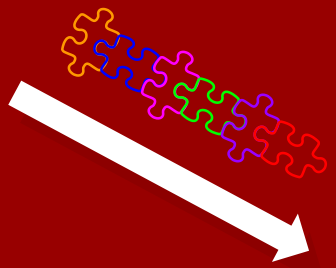


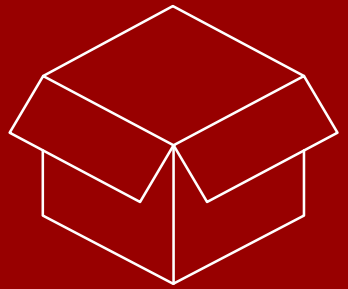


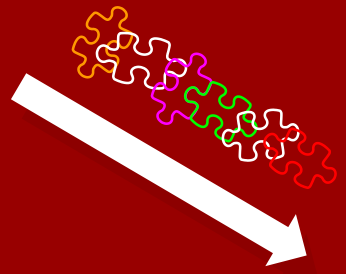
Bob

Bob receives the message, but...

Alice







Bob

Some of the message may *(or may not)* have been messed with

String Transmission

- Bob receives a binary string from Alice
- Unfortunately, there are up to k errors in the string of length n
  - **Up to** k bits in that string could be inverted
- However, Bob knows that the string that Alice sent was NOT periodic…
  - This means that the string cannot be represented by a single, repeated pattern
    - 01010101 is periodic
    - 0001 is not

PROGRAM STRUCTURE

Input

- The first line contains the number of test cases T.
- T test cases follow.
- Each test case contains:
- Two integers N and K on the first line, and a binary string of length N on the next line.

Output

- Output T lines, one for each test case.
- Since the answers can be really big, output the numbers modulo 1000000007.

961 solutions submitted

That’s a lot of solutions

60 max score

And a lot of score to be had

Hard

“

This is really a difficult one, can someone provide a editorial?

- liurenjie

EXAMPLE I/O

EXAMPLE I/O

Input

```
1
5 0
00000
```

Output

EXAMPLE I/O

Input

```
1
5 0
00000
```

Output

```
0
```

EXAMPLE I/O

Input

```
1
5 0
00000
```

Output

```
0
```

Explanation: Since “`00000`” is periodic, and there were `0` errors, there are no possible strings that Alice could have transmitted...

EXAMPLE I/O

Input

```
1
3 1
001
```

Output

EXAMPLE I/O

Input

```
1
3 1
001
```

Output

```
3
```

Explanation: Alice could have transmitted "`001`", or "`011`" or "`101`".

EXAMPLE I/O

Input

```
1
3 1
101
```

Output

EXAMPLE I/O

Input

```
1
3 1
101
```

Output

```
6
```

Explanation: Alice could have transmitted either “001”, “010”, “100”, “011”, “101”, or “110”...

EXAMPLE I/O FOR T > 1

Input

```
3
5 0
00000
3 1
001
3 3
101
```

Output

```
0
3
6
```

INITIAL APPROACH

What We Did

- Initially focused on generating non-periodic permutations of the string from the get go
- Find divisors of length of *n* to decrease number of strings generated
- Generate strings and see if that matches the given constraints of possible errors in string

WHY WE ARE DUMB

Are We Dumb?

- Our initial attempt was going to be too slow for hackerrank

- Changed solution methodology to employ dynamic programming
  - Used a 2D array to keep track of progress and number of possible result

SOLUTION

The Right Answer

- DYNAMIC PROGRAMMING

- Must be dynamic because calculating with brute force would take to long
  - With super large strings, the results can be **HUGE** which is why we use that mod, 1000000007, and why we don't use brute force algorithms

CODE WALKTHROUGH

Not Going to Happen

- General Algorithm
  - Manipulate a matrix iteratively using the divisors of k as offsets

- Tracing takes too long…
  - Here are some of the cool parts though!

Pascal's Rule

- Like nCr but not

```python
def pascalRule(n, k):
    # Simple variable. Set answer equal to -1.
    ans = -1
    for i in range(0, k+1):
        # // is integer division in python3
        ans = ans + factorial(n) // ( factorial(i) * factorial(n-i) )
    return ans
```

$$\begin{aligned}\binom{n}{k} + \binom{n}{k-1} &= \frac{n!}{k!(n-k)!} + \frac{n!}{(k-1)!(n-k+1)!} \\ &= n!\left[\frac{n-k+1}{k!(n-k+1)!} + \frac{k}{k!(n-k+1)!}\right] \\ &= \frac{n!(n+1)}{k!(n-k+1)!} = \binom{n+1}{k}\end{aligned}$$

Python > Everything

Cool Python way

```
result = [[0]*(nkSum) for i in range(len(div))]
```

Boring way

```
result = []
for i in range(len(div)):
    result.append([0] * nkSum)
```

Python > Everything

Cool Python way

```
for i, d in enumerate(div):
```

Boring way

```
i = 0
for d in enumerate(div):
    ...
    i+=1
```

RUNNING THE CODE

THANKS!

Any questions?

We pledge our honor that we have abided by the Stevens Honor System