Problem D. Trie Sharding

Small input 9 points	Solve D-small
Large input 30 points	Solve D-large

Problem

A set of strings **S** can be stored efficiently in a *trie*. A trie is a rooted tree that has one node for every prefix of every string in **S**, without duplicates.

For example, if **S** were "AAA", "AAB", "AB", "B", the corresponding trie would contain 7 nodes corresponding to the prefixes "", "A", "AA", AAA", "AAB", "AB", and "B".

I have a server that contains **S** in one big trie. Unfortunately, **S** has become very large, and I am having trouble fitting everything in memory on one server. To solve this problem, I want to switch to storing **S** across **N** separate servers. Specifically, **S** will be divided up into disjoint, non-empty subsets $\mathbf{T_1}$, $\mathbf{T_2}$, ..., $\mathbf{T_N}$, and on each server i, I will build a trie containing just the strings in $\mathbf{T_i}$. The downside of this approach is the total number of nodes across all **N** tries may go up. To make things worse, I can't control how the set of strings is divided up!

For example, suppose "AAA", "AAB", "AB", "B" are split into two servers, one containing "AAA" and "B", and the other containing "AAB", "AB". Then the trie on the first server would need 5 nodes ("", "A", "AA", "AAA", "B"), and the trie on the second server would also need 5 nodes ("", "A", "AA", "AAB", "AB"). In this case, I will need 10 nodes altogether across the two servers, as opposed to the 7 nodes I would need if I could put everything on just one server.

Given an assignment of strings to **N** servers, I want to compute the worst-case total number of nodes across all servers, and how likely it is to happen. I can then decide if my plan is good or too risky.

Given **S** and **N**, what is the largest number of nodes that I might end up with? Additionally, how many ways are there of choosing $\mathbf{T_1}$, $\mathbf{T_2}$, ..., $\mathbf{T_N}$ for which the number of nodes is maximum? Note that the **N** servers are different -- if a string appears in $\mathbf{T_i}$ in one arrangement and in $\mathbf{T_j}$ (**i** != **j**) in another arrangement, then the two arrangements are considered different. Print the remainder of the number of possible arrangements after division by 1,000,000,007.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case starts with a line containing two space-separated integers: **M** and **N**. **M** lines follow, each containing one string in **S**.

Output

For each test case, output one line containing "Case #**i**: **X** **Y**", where **i** is the case number (starting from 1), **X** is the worst-case number of nodes in all the tries combined, and **Y** is the number of ways (modulo 1,000,000,007) to assign strings to servers such that the number of nodes in all **N** servers are **X**.

Limits

$1 \le \mathbf{T} \le 100$.
Strings in **S** will contain only upper case English letters.
The strings in **S** will all be distinct.
$\mathbf{N} \le \mathbf{M}$

Small dataset

$1 \le \mathbf{M} \le 8$
$1 \le \mathbf{N} \le 4$
Each string in **S** will have between 1 and 10 characters, inclusive.

Large dataset

$1 \le \mathbf{M} \le 1000$
$1 \le \mathbf{N} \le 100$
Each string in **S** will have between 1 and 100 characters, inclusive.

Sample

Input	Output
2 4 2 AAA AAB AB B 5 2 A B C D E	Case #1: 10 8 Case #2: 7 30