Problem Statement for **Doubly Sorted Grid**

Problem:
A rectangular grid with lower case English letters in each cell is called doubly sorted if in each row the letters are non-decreasing from the left to the right, and in each column the letters are non-decreasing from the top to the bottom. In the following examples, the first two grids are doubly sorted, while the other two are not:

```
abc    ace    aceg    base
def    ade    cdef    base
ghi    bdg    xxyy    base
```

You are given a partially-filled grid, where some of the cells are filled with letters. Your task is to compute the number of ways you can fill the rest of the cells so that the resulting grid is doubly sorted. The answer might be a big number; you need to output the number of ways modulo 10007.

Input:
The first line of input gives the number of test cases, T. T test cases follow. Each test case starts with a line containing two integers R and C, the number of rows and the number of columns respectively. This is followed by R lines, each containing a string of length C, giving the partially-filled grid. Each character in the grid is either a lower-case English letter, or '.', indicating that the cell is not filled yet.

Output:
For each test case, output one line. That line should contain the number of possible doubly-sorted grids, modulo 10007.

Constraints:

1 ≤ T ≤ 40
Each character in the partially-filled grid is either '.' or a lower-case English letter.
1 ≤ R, C ≤ 4
Compilation time: 10 seconds,
Execution time: 5 seconds.
Memory usage: 256 MB.

Examples:

Input:

```
3
2 2
ad
c.
3 3
.a.
a.z
.z.
4 4
....
.g..
.cj.
....
```

Output:

```
23
7569
0
```