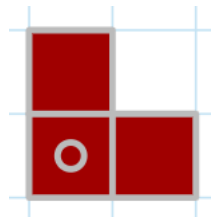


# Tromino Packing

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            2 seconds  
Memory limit:         512 megabytes

Busy Beaver found a strange puzzle. The puzzle consists of a box divided into an  $N \times M$  grid of squares. Each cell of the grid is either occupied (denoted by a '#'), empty (denoted by a '.'), or empty but marked with a 'o'.

The puzzle comes with a bunch of L-tromino pieces, one for each 'o' cell of the grid. An L-tromino is composed of three square blocks in the shape of an L as shown. The center of the L-tromino is the square that is adjacent to both of the other two squares (marked with an o in the picture).



Busy Beaver realizes that to solve the puzzle, he needs to pack all the L-trominos into the box such that the L-trominos are aligned to the grid, each L-tromino center is at an empty cell marked with a 'o' and no two L-trominos overlap each other or an occupied cell. All L-trominos can be rotated freely.

Can you help Busy Beaver count the number of ways to solve the puzzle? As the answer may be large, output it modulo  $10^9 + 7$ .

## Input

Each test contains multiple test cases. The first line contains the number of test cases  $T$  ( $1 \leq T \leq 500$ ). The description of the test cases follows.

The first line of each test case contains two integers  $N$  and  $M$  ( $2 \leq N, M \leq 1000$ ), the dimensions of the grid.

The next  $N$  lines each contain  $M$  characters, describing a row of the grid. Each character is either '#', '.', or 'o'.

It is guaranteed that the sum of  $N$  over all test cases does not exceed 1000.

It is guaranteed that the sum of  $M$  over all test cases does not exceed 1000.

## Output

For each test case, output a single integer: the number of ways to solve the puzzle modulo  $10^9 + 7$ .

## Scoring

Let  $(r, c)$  denote the cell at row  $r$  and column  $c$  ( $1 \leq r \leq N, 1 \leq c \leq M$ ).

- (10 points) Each 'o' cell is Manhattan distance at least 3 away from each other 'o' cell. That is, if  $(r_1, c_1)$  and  $(r_2, c_2)$  are two different 'o' cells, then  $|r_1 - r_2| + |c_1 - c_2| \geq 3$ .
- (30 points) Each 'o' cell is vertically adjacent to at least one other 'o' or '#' cell. That is, for any 'o' cell at  $(r, c)$ , either  $(r - 1, c)$  or  $(r + 1, c)$  is an '#' cell or another 'o' cell.
- (60 points) No additional constraints.

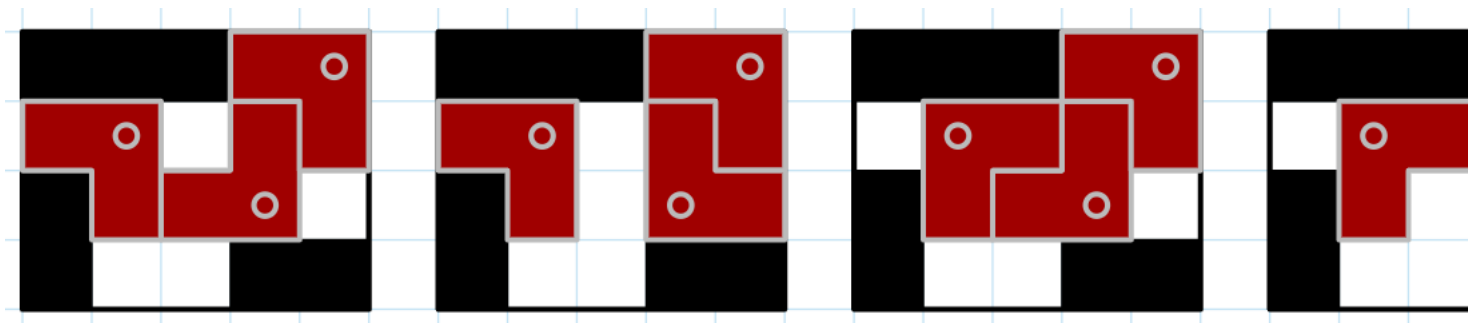
## Example

standard input	standard output
5	4
4 5	6
###.o	1
.o...	0
#..o.	1
#..##	
5 5	
..###	
.o..#	
.o.o.	
..#o.	
###..	
8 31	
.#####.#..oo..o..o#o..o....o	
o.#####.o#o...o..o..#..o..oo..	
o..####.o.#####.#####o..###	
..o.o..o..#####.o#####.o###	
.o##..##.o#####o.#####.o###	
o.##.o##.o#####.#####o..###	
..#####.##..o..o..o..####.o###	
.o#####o.#o..o..o..o####o..###	
2 3	
#.o	
.o.	
2 2	
..	
..	

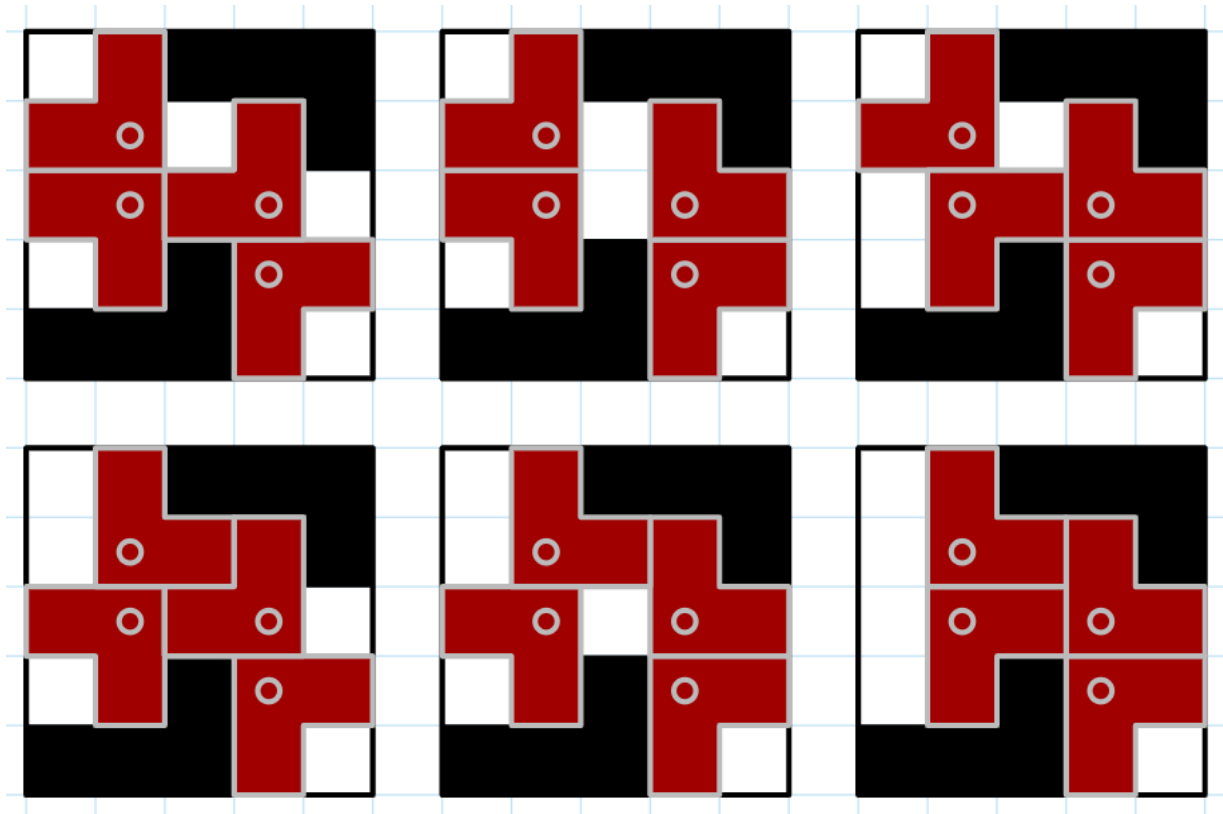
## Note

Note that the first test case satisfies the constraints for the first subtask and the second test case satisfies the constraints for the second subtask.

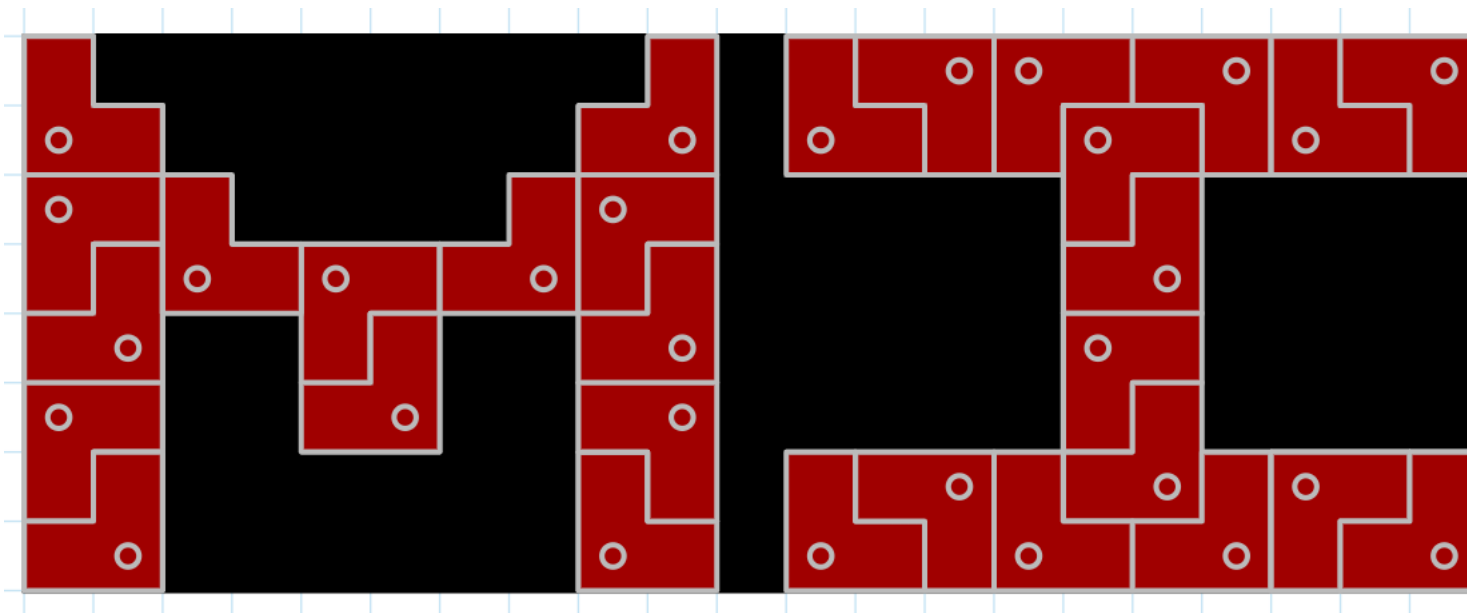
In the first test case, the 4 ways to solve the puzzle are as shown:



In the second test case, the 6 ways to solve the puzzle are as shown:



In the third test case, the only way to solve the puzzle is as shown:



In the fourth test case, there are no ways to solve the puzzle.

In the fifth test case, there are no 'o' cells, so the unique way to solve the puzzle is to not place any L-trominos.