



时间限制：C/C++/Rust/Pascal 2秒，其他语言4秒

空间限制：C/C++/Rust/Pascal 1024 M，其他语言2048 M

Special Judge, 64bit IO Format: %lld

① C++ (clang++18)

1

ACM模

请通过

入输出

出描述

题目描述

"Super Mario Maker" is a commemorative work for the 30th anniversary of the "Super Mario" series. The biggest feature of "Super Mario Maker" is that the entire game includes a complete level editor system, allowing players to design and create Mario levels using the Wii U GamePad; players can also upload their designed levels to Nintendo's servers to share with players around the world.

As a loyal player of "Super Mario," you have created a balloon level! This level can be viewed as a finite-sized grid, with some cells containing impassable obstacles such as spikes. Players need to control "Balloon Mario" to move from the first column of the grid to the last column by navigating adjacent cells and avoiding obstacles like spikes. As a side-scrolling game, its unique feature lies in its limited field of vision, with areas outside the visible range being unreachable. This setting poses a significant challenge for reasonable level design: even for very clever players, the limitations of visibility may lead them into a "dead end" where they cannot complete the level. In this problem, you need to determine whether such a phenomenon exists in the level you designed.

In this problem, we will consider the following simplified model. A level will be viewed as a grid of size $N \times M$, with the "Balloon Mario" character starting at $(1, 1)$, and the player aiming to maneuver "Balloon Mario" to $(1, M)$. Some cells contain obstacles, and the "Balloon Mario" character cannot pass through these cells. **No obstacles are placed in the first and last columns.** When the "Balloon Mario" character is at (X, Y) , the player's field of vision is $\{(U, V) \mid 1 \leq U \leq N, Y \leq V \leq \min(Y + K, M)\}$, meaning the player can see whether there are obstacles in these cells. When the "Balloon Mario" character is at (X, Y) , the character can move up to $(X + 1, Y)$, down to $(X - 1, Y)$, or right to $(X, Y + 1)$, as long as the target cell is within the map boundaries and does not contain an obstacle.

You need to determine whether there exists a path from the starting point $s_0 = (1, 1)$ to some point $s_\ell = (X, Y)$ such that the following conditions are simultaneously satisfied:

- "Balloon Mario" can move from s_0 to s_ℓ along the path $P = s_0 s_1 \dots s_\ell$.
- For any $1 \leq i \leq \ell$, the player cannot rule out the possibility of reaching $(1, M)$ from s_i based on the visibility information at s_{i-1} .
- However, the player cannot move "Balloon Mario" from s_ℓ to $(1, M)$.

输入描述:

The first line of input contains a positive integer T ($1 \leq T \leq 10^4$), representing the number of test cases. Each test case is described as follows.

The first line of each test case contains three positive integers N, M, K ($2 \leq N, M, N \cdot M \leq 10^6$, $1 \leq K \leq M - 1$), representing the size parameters of the map and the visibility range parameter.

The next N lines each contain a binary string of length M , s_i , representing the obstacle information for the i -th row of the map. Here, $s_{i,j} = 1$ if and only if there is an obstacle at position (i, j) . The input guarantees that there are no obstacles in the first and last columns, and that there exists a path from

运行结果

自测