Problem 6: Indomitable Human Spirit vs AAA Company 3+5 Points

Problem ID: `stickdrift`
Rank: `2+3`

Introduction

Unlocking the manifolds has become easy, but you have been confronted with a fatal flaw in your over-engineered Among Us robot. You designed your robot to control the cursor with a Nintendo switch Joy-Con that has inevitably fallen victim to the curse of stick drift. Being too lazy (and broke) to replace the controller, you decide to fight the stick drift through clever software. Is your will to win in the greatest game of all time, Among Us by InnerSloth LLC, enough to overcome under-engineered controllers from the $84.952 billion USD valuated company Nintendo?

Problem Statement

This is a harder version of `unlockmanifolds`! Key changes in `stickdrift` are highlighted.

You are given a grid $\mathbf{G}$ of integers with $\mathbf{N}$ rows and $\mathbf{M}$ columns. Each tile on the grid represents a button that is labeled with an integer from 1 to $\mathbf{N} \times \mathbf{M}$. Starting with your cursor at the top left of the grid $\mathbf{G}$[0][0], your goal is to count the minimum number of actions needed to visit every button in order from 1 to $\mathbf{N} \times \mathbf{M}$.

An action is defined as one of the following:

- Move the cursor to an adjacent button: up, down, left, right
- Do nothing (take an action to do nothing!)

However, every action you take is **combined** with an additional drift input from the sequence $\mathbf{S}$. The potential drift inputs are:

- `U`: up
- `D`: down
- `L`: left
- `R`: right

The i^{th} action you take is combined with the i^{th} corresponding drift input in $\mathbf{S}$. If $\mathbf{S}$ runs out of inputs, the next drift input loops back to the first input in $\mathbf{S}$. Note that the effects of the i^{th} action and the i^{th} drift input occur **simultaneously**. In other words, the cursor **teleports** to the corresponding button as a result of combining the i^{th} action and the i^{th} drift input.

The cursor also wraps around the screen. So if the cursor moves such that it exits the top of the screen, it will appear at the bottom of the same column.

Find the minimum number of actions required to visit each button in ascending order.

Input Format

The first line of the input contains a single integer $\mathbf{T}$ denoting the number of test cases that follow. For each test case:

- The first line contains 2 spaced integers $\mathbf{N}$ $\mathbf{M}$ denoting the rows and columns of the board, respectively.
- The second line contains a single string $\mathbf{S}$ representing the sequence of drift inputs.
- The next $\mathbf{N}$ lines each contain $\mathbf{M}$ space-separated integers, representing a row of the grid.

Output Format

For each test case, output a single integer denoting the minimum number of actions required to visit each button in ascending order.

Constraints

Time Limit: **1 Second**
Memory Limit: **1024 MB**

Main Test Set

$1 \leq \mathbf{T} \leq 100$
$1 \leq |\mathbf{S}| \leq 10^4$
$1 \leq \mathbf{N} \times \mathbf{M} \leq 10^3$
It is guaranteed the sum of $\mathbf{N} \times \mathbf{M}$ across all test cases in a test file is less than 10^3

Bonus Test Set

$1 \leq \mathbf{T} \leq 100$
$1 \leq |\mathbf{S}| \leq 10^4$
$1 \leq \mathbf{N} \times \mathbf{M} \leq 2 \times 10^5$
It is guaranteed the sum of $\mathbf{N} \times \mathbf{M}$ across all test cases in a test file is less than 2×10^5

Sample Test Cases

Sample Input Download

```
5
2 2
UUD
1 2
3 4
1 4
L
4 1 2 3
2 3
UD
1 3 5
2 4 6
3 3
RRLLUUDD
1 2 3
4 5 6
7 8 9
1 1
UDLR
1
```

Sample Output Download

```
5
8
5
11
0
```

Main Sample Explanations

For test case #1, 5 actions are enough to visit every button in ascending order. This is achieved with the following sequence of actions:

Note that we start already having visited **button 1**.

1. Move Right and Drift Up, ending up on button 4. Note that the cursor teleports to button 4 and does not visit button 2 in the process.
2. Do Nothing and Drift Up, ending up on **button 2**.
3. Move Right and Drift Down, ending up on **button 3**.
4. Move Right and Drift Up, ending up on button 2. Note that the drift input here has looped back to the beginning of $\mathbf{S}$.
5. Do Nothing and Drift Up, ending up **button 4**.

For test case #2, 8 actions is enough to visit every button in ascending order. This is achieved with the following sequence of actions:

1. Move Left and Drift Left, ending up on button 2.
2. Do Nothing and Drift Left, ending up on **button 1**.
3. Move Left and Drift Left, ending up on button 3.
4. Do Nothing and Drift Left, ending up on **button 2**.
5. Move Left and Drift Left, ending up on button 4.
6. Do Nothing and Drift Left, ending up on **button 3**.
7. Move Left and Drift Left, ending up on button 1.
8. Do Nothing and Drift Left, ending up on **button 4**.

Note the drift sequence repeatedly loops over the only drift input $\mathtt{L}$ in $\mathbf{S}$.

For test case #4, 11 actions is enough to visit every button in ascending order.

For test case #5, 0 actions is enough to visit every button in ascending order as we start having already visited **button 1**.