Bloomberg CODECON

Memory Limit: 1350 MB
Time Limit: 5 s

Snake Growth Model (500 points)

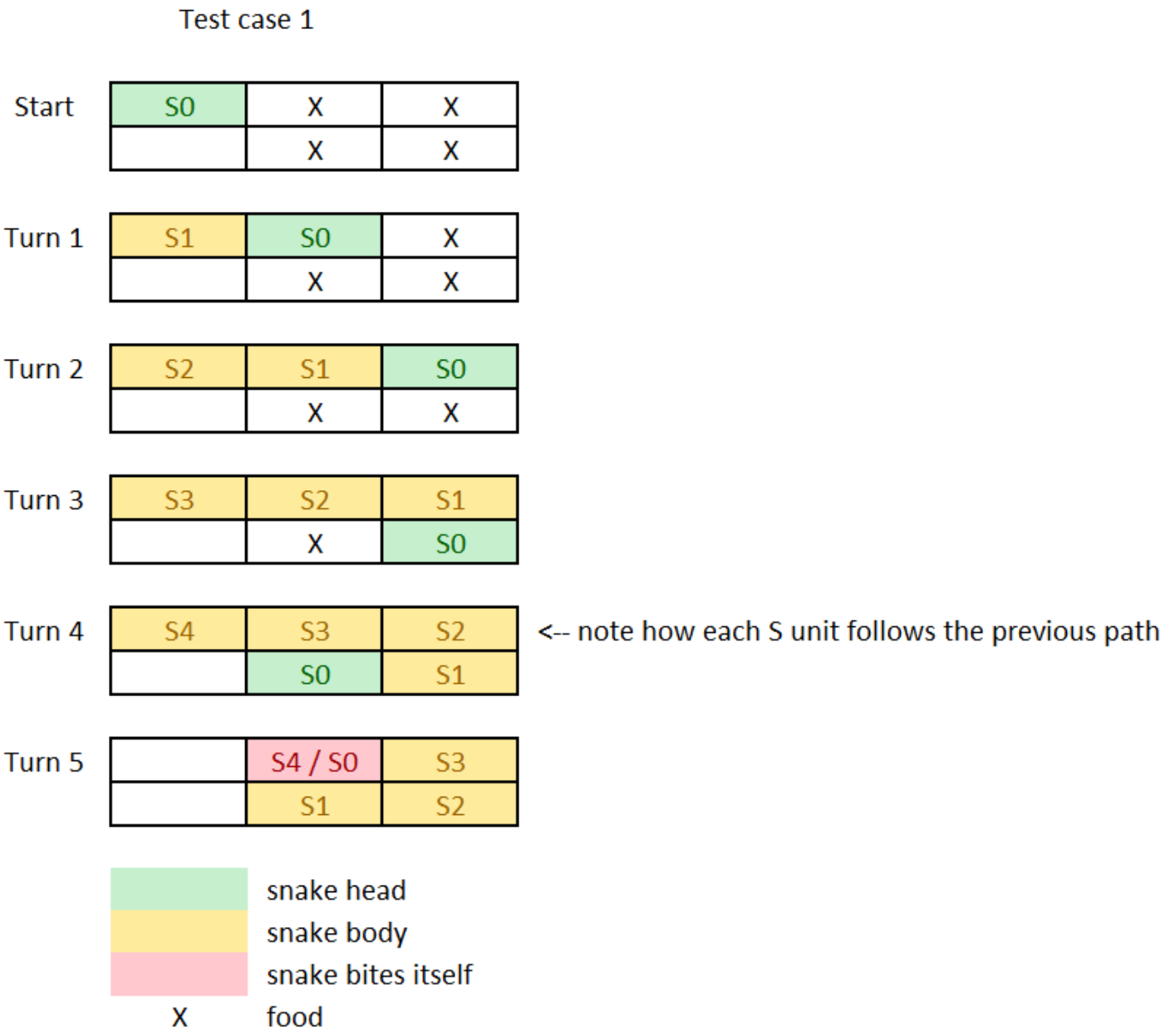
Introduction

You've invented a new model to simulate snake growth. Given the locations of the snake, its food, and the path it travels, determine the point at which the snake will bite itself (and no longer grow).

The snake can move around in a pixelated sandbox of some specified size. You will be given the exact sequence of the snake's movements, as well as the location of food in the sandbox. Each movement will move the snake's head by one cell in the specified cardinal direction. Each cell of the snake's tail will be dragged along behind the head, and follow the exact path the head originally took; that is, any given tail cell will follow the same directions the head did, however many turns back that the cell is cells behind the head.

The snake starts being only one cell in size, but each item that the snake eats adds 1 unit to its tail length (at the end of the tail, the moment the food is consumed). The snake starts at the top-left of the sandbox.

Note that the snakes head & tail all move together at once. This means that if the snake's head is moving into the cell currently occupied by the tip of its tail, the snake wont actually bite itself since in that same turn the tip of its tail would also move out of the way.



[Example for sample 1 (http://i.imgur.com/11ejvbA.png)](http://i.imgur.com/11ejvbA.png)

Input Specifications

The first line will be the size of the sandbox `R C` in rows & columns separated by a space, where $1 < R \leq 100$ and $1 < C \leq 100$. The next line will be the total number of food locations, $2 < M \leq 400$, followed by that many lines of R_i C_i (coordinates here are zero-indexed). The next line will be the total number of moves $3 < N \leq 600$, followed by that many lines each having a single-character directional code. The codes are:

- `U` : Up (row-1)
- `R` : Right (col+1)
- `L` : Left (col-1)
- `D` : Down (row+1)

Output Specifications

A single integer value that gives the number of actions that have been executed until the snake bites itself or -1 if the snake does not bite itself.

Sample Input/Output

Input

```
2 3
4
0 1
0 2
1 2
1 1
8
R
R
D
L
U
R
D
L
```

Output

```
5
```

Explanation

Snake eats at each move and it will bite itself after going in a tight loop.

Input

```
1 6
3
0 1
0 2
0 3
5
R
R
R
R
R
```

Output

```
-1
```

Explanation

Snake eats the three items but never turns onto itself.