5964. Execution of All Suffix Instructions Staying in a Grid

My Submissions (/contest/weekly-contest-273/problems/execution-of-all-suffix-instructions-staying-in-a-grid/submissions/)

Back to Contest (/contest/weekly-contest-273/)

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

There is an `n x n` grid, with the top-left cell at `(0, 0)` and the bottom-right cell at `(n - 1, n - 1)`. You are given the integer `n` and an integer array `startPos` where `startPos = [start_row, start_col]` indicates that a robot is initially at cell `(start_row, start_col)`.

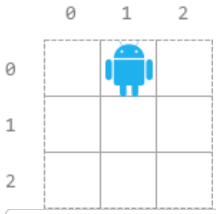
You are also given a **0-indexed** string `s` of length `m` where `s[i]` is the `i^th` instruction for the robot: `'L'` (move left), `'R'` (move right), `'U'` (move up), and `'D'` (move down).

The robot can begin executing from any `i^th` instruction in `s`. It executes the instructions one by one towards the end of `s` but it stops if either of these conditions is met:

- The next instruction will move the robot off the grid.
- There are no more instructions left to execute.

Return *an array* `answer` *of length* `m` *where* `answer[i]` *is* ***the number of instructions*** *the robot can execute if the robot* ***begins executing from*** *the* `i^th` *instruction in* `s`.

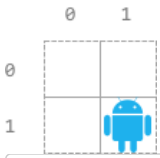
Example 1:




```
Input: n = 3, startPos = [0,1], s = "RRDDLU"
Output: [1,5,4,3,1,0]
Explanation: Starting from startPos and beginning execution from the i^th instruction:
- 0^th: "RRDDLU". Only one instruction "R" can be executed before it moves off the grid.
- 1^st:  "RDDLU". All five instructions can be executed while it stays in the grid and ends at (1, 1).
- 2^nd:   "DDLU". All four instructions can be executed while it stays in the grid and ends at (1, 0).
- 3^rd:    "DLU". All three instructions can be executed while it stays in the grid and ends at (0, 0).

- 4^th:     "LU". Only one instruction "L" can be executed before it moves off the grid.
- 5^th:      "U". If moving up, it would move off the grid.
```

Example 2:




```
Input: n = 2, startPos = [1,1], s = "LURD"
Output: [4,1,0,0]
Explanation:
- 0^th: "LURD".
- 1^st:  "URD".
- 2^nd:   "RD".
- 3^rd:    "D".
```

Example 3:

```
Input: n = 1, startPos = [0,0], s = "LRUD"
Output: [0,0,0,0]
Explanation: No matter which instruction the robot begins execution from, it would move off the grid.
```

Constraints:

- m == s.length
- 1 <= n, m <= 500
- startPos.length == 2
- 0 <= $start_{row}$, $start_{col}$ < n
- s consists of 'L', 'R', 'U', and 'D'.

JavaScript

```
/**
 * @param {number} n
 * @param {number[]} startPos
 * @param {string} s
 * @return {number[]}
 */
var executeInstructions = function(n, startPos, s) {

};
```

Custom Testcase Use Example Testcases

Run Submit