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 `(`$start_{row}$`, `$start_{col}$`)`.

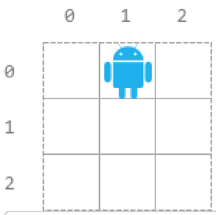
You are also given a **0-indexed** string `s` of length `m` where `s[i]` is the i^{th} instruction for the robot: `'L'` (move left), `'R'` (move right), `'U'` (move up), and `'D'` (move down).

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.

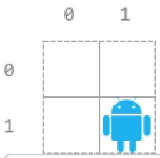
Return *an array* `answer` *of length* `m` *where* `answer[i]` *is* ***the number of instructions*** *the robot can execute if the robot* ***begins executing from*** *the* i^{th} *instruction in* `s`.

Example 1:



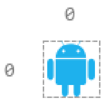
```
Input: n = 3, startPos = [0,1], s = "RRDDLU"
Output: [1,5,4,3,1,0]
Explanation: Starting from startPos and beginning execution from the ith instruction:
- 0th: "RRDDLU". Only one instruction "R" can be executed before it moves off the grid.
- 1st:  "RDDLU". All five instructions can be executed while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

- 4th:     "LU". Only one instruction "L" can be executed before it moves off the grid.
- 5th:      "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:
- 0th: "LURD".
- 1st:  "URD".
- 2nd:   "RD".
- 3rd:    "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 <=` $\mathtt{start}_{\mathtt{row}}$, $\mathtt{start}_{\mathtt{col}}$ `< n`
- `s` consists of `'L'`, `'R'`, `'U'`, and `'D'`.

JavaScript

```javascript
const executeInstructions = (n, startPos, s) => {
    let res = [], m = s.length;
    for (let i = 0; i < m; i++) {
        let [x, y] = startPos;
        // pr("\nstart", x, y);
        let j;
        for (j = i; j < m; j++) {
            let c = s[j];
            if (c == 'U') {
                if (x - 1 >= 0) {
                    x--;
                } else {
                    break;
                }
            } else if (c == 'D') {
                if (x + 1 < n) {
                    x++;
                } else {
                    break;
                }
            } else if (c == 'L') {
                if (y - 1 >= 0) {
                    y--;
                } else {
                    break;
                }
            } else if (c == 'R') {
                if (y + 1 < n) {
                    y++;
                } else {
                    break;
                }
            }
            // pr(c, x, y);
        }
        // pr("check", i, s[i], j, s[j]);
        let step = j - i;
        res.push(step);
    }
    return res;
};
```

□ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/607207111/) More Details (/submissions/detail/607207111/)

Share your acceptance!

2