6028. Count Collisions on a Road

My Submissions (/contest/weekly-contest-285/problems/count-collisions-on-a-road/submissions/) Back to Contest (/contest/weekly-contest-285/)

There are `n` cars on an infinitely long road. The cars are numbered from `0` to `n - 1` from left to right and each car is present at a **unique** point.

You are given a **0-indexed** string `directions` of length `n`. `directions[i]` can be either `'L'`, `'R'`, or `'S'` denoting whether the i^{th} car is moving towards the **left**, towards the **right**, or **staying** at its current point respectively. Each moving car has the **same speed**.

The number of collisions can be calculated as follows:

- When two cars moving in **opposite** directions collide with each other, the number of collisions increases by `2`.
- When a moving car collides with a stationary car, the number of collisions increases by `1`.

After a collision, the cars involved can no longer move and will stay at the point where they collided. Other than that, cars cannot change their state or direction of motion.

Return *the **total number of collisions** that will happen on the road*.

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

Example 1:

```
Input: directions = "RLRSLL"
Output: 5
Explanation:
The collisions that will happen on the road are:
- Cars 0 and 1 will collide with each other. Since they are moving in opposite directions, the number of collisions becomes
- Cars 2 and 3 will collide with each other. Since car 3 is stationary, the number of collisions becomes 2 + 1 = 3.
- Cars 3 and 4 will collide with each other. Since car 3 is stationary, the number of collisions becomes 3 + 1 = 4.
- Cars 4 and 5 will collide with each other. After car 4 collides with car 3, it will stay at the point of collision and get
Thus, the total number of collisions that will happen on the road is 5.
```

Example 2:

```
Input: directions = "LLRR"
Output: 0
Explanation:
No cars will collide with each other. Thus, the total number of collisions that will happen on the road is 0.
```

Constraints:

- `1 <= directions.length <= 10^5`
- `directions[i]` is either `'L'`, `'R'`, or `'S'`.

JavaScript

```
const cutMaxConsecutive = (a_or_s) => { let d = [], start = 0, n = a_or_s.length; for (let i = 0; i + 1 < n; i++) {
if (a_or_s[i + 1] != a_or_s[i]) { d.push(a_or_s.slice(start, i + 1)); start = i + 1; } } d.push(a_or_s.slice(start));
return d; };

const countCollisions = (s) => {
    let d = cutMaxConsecutive(s), n = d.length, stop = Array(n).fill(0), res = 0;
    // pr(d);
    for (let i = 0; i < n;) {
        let cur = d[i], right;
        if (i + 1 < n) right = d[i + 1];
        if (cur[0] == 'S') {
            stop[i] = 1;
            i++;
            continue;
        }
        if (cur[0] == 'R' && right && right[0] == 'L') {
```

```
            stop[i] = 1;
            stop[i + 1] = 1;
            let add = 2 + (cur.length - 1) + (right.length - 1);
            // pr("cur", d[i], "right", d[i + 1], "add", add)
            res += add;
            i += 2;
            continue;
        }
        i++;
    }
    // pr(stop, res);
    for (let i = 0; i < n; i++) { // all others hit stop
        let dir = d[i][0];
        if (stop[i] == 0) {
            if (i == 0) {
                if (stop[i + 1] == 1 && dir == 'R') res += d[i].length;
            } else if (i == n - 1) {
                if (stop[i - 1] == 1 && dir == 'L') res += d[i].length;
            } else {
                if ((dir == 'L' && stop[i - 1] == 1) || (dir == 'R' && stop[i + 1] == 1)) res += d[i].length;
            }
        }
    }
    return res;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/663455801/) | More Details > (/submissions/detail/663455801/)

Share your acceptance!

2