Problem 2731: Movement of Robots

Problem Information

Difficulty: Medium
Acceptance Rate: 27.51%
Paid Only: No
Tags: Array, Brainteaser, Sorting, Prefix Sum

Problem Description

Some robots are standing on an infinite number line with their initial coordinates given by a **0-indexed** integer array `nums` and will start moving once given the command to move. The robots will move a unit distance each second.

You are given a string `s` denoting the direction in which robots will move on command. `'L'` means the robot will move towards the left side or negative side of the number line, whereas `'R'` means the robot will move towards the right side or positive side of the number line.

If two robots collide, they will start moving in opposite directions.

Return _the sum of distances between all the pairs of robots _`d` _seconds after the command. _Since the sum can be very large, return it modulo `109 + 7`.

Note:

* For two robots at the index `i` and `j`, pair `(i,j)` and pair `(j,i)` are considered the same pair. * When robots collide, they **instantly change** their directions without wasting any time. * Collision happens when two robots share the same place in a moment. * For example, if a robot is positioned in 0 going to the right and another is positioned in 2 going to the left, the next second they'll be both in 1 and they will change direction and the next second the first one will be in 0, heading left, and another will be in 2, heading right. * For example, if a robot is positioned in 0 going to the right and another is positioned in 1 going to the left, the next second the first one will be in 0, heading left, and another will be in 1, heading right.

Example 1:

Input: nums = [-2,0,2], s = "RLL", d = 3 **Output:** 8 **Explanation:** After 1 second, the positions are [-1,-1,1]. Now, the robot at index 0 will move left, and the robot at index 1 will move right. After 2 seconds, the positions are [-2,0,0]. Now, the robot at index 1 will move left, and the robot at index 2 will move right. After 3 seconds, the positions are [-3,-1,1]. The distance between the robot at index 0 and 1 is abs(-3 - (-1)) = 2. The distance between the robot at index 0 and 2 is abs(-3 - 1) = 4. The distance between the robot at index 1 and 2 is abs(-1 - 1) = 2. The sum of the pairs of all distances = 2 + 4 + 2 = 8.

Example 2:

Input: nums = [1,0], s = "RL", d = 2 **Output:** 5 **Explanation:** After 1 second, the positions are [2,-1]. After 2 seconds, the positions are [3,-2]. The distance between the two robots is abs(-2 - 3) = 5.

Constraints:

* `2 <= nums.length <= 105` * `-2 * 109 <= nums[i] <= 2 * 109` * `0 <= d <= 109` * `nums.length == s.length ` * `s` consists of 'L' and 'R' only * `nums[i]` will be unique.

Code Snippets

C++:

```
class Solution {
public:
int sumDistance(vector<int>& nums, string s, int d) {

}
};
```

Java:

```
class Solution {
public int sumDistance(int[] nums, String s, int d) {

}
}
```

Python3:

```
class Solution:
def sumDistance(self, nums: List[int], s: str, d: int) -> int:
```