Problem 1041: Robot Bounded In Circle

Problem Information

Difficulty: Medium
Acceptance Rate: 56.39%
Paid Only: No
Tags: Math, String, Simulation

Problem Description

On an infinite plane, a robot initially stands at `(0, 0)` and faces north. Note that:

* The **north direction** is the positive direction of the y-axis. * The **south direction** is the negative direction of the y-axis. * The **east direction** is the positive direction of the x-axis. * The **west direction** is the negative direction of the x-axis.

The robot can receive one of three instructions:

* `"G"`: go straight 1 unit. * `"L"`: turn 90 degrees to the left (i.e., anti-clockwise direction). * `"R"`: turn 90 degrees to the right (i.e., clockwise direction).

The robot performs the `instructions` given in order, and repeats them forever.

Return `true` if and only if there exists a circle in the plane such that the robot never leaves the circle.

Example 1:

Input: instructions = "GGLLGG" **Output:** true **Explanation:** The robot is initially at (0, 0) facing the north direction. "G": move one step. Position: (0, 1). Direction: North. "G": move one step. Position: (0, 2). Direction: North. "L": turn 90 degrees anti-clockwise. Position: (0, 2). Direction: West. "L": turn 90 degrees anti-clockwise. Position: (0, 2). Direction: South. "G": move one step. Position: (0, 1). Direction: South. "G": move one step. Position: (0, 0). Direction: South. Repeating the instructions, the robot goes into the cycle: (0, 0) --> (0, 1) --> (0, 2) --> (0, 1) --> (0, 0). Based on that, we return true.

Example 2:

Input: instructions = "GG" **Output:** false **Explanation:** The robot is initially at (0, 0) facing the north direction. "G": move one step. Position: (0, 1). Direction: North. "G": move one step. Position: (0, 2). Direction: North. Repeating the instructions, keeps advancing in the north direction and does not go into cycles. Based on that, we return false.

Example 3:

Input: instructions = "GL" **Output:** true **Explanation:** The robot is initially at (0, 0) facing the north direction. "G": move one step. Position: (0, 1). Direction: North. "L": turn 90 degrees anti-clockwise. Position: (0, 1). Direction: West. "G": move one step. Position: (-1, 1). Direction: West. "L": turn 90 degrees anti-clockwise. Position: (-1, 1). Direction: South. "G": move one step. Position: (-1, 0). Direction: South. "L": turn 90 degrees anti-clockwise. Position: (-1, 0). Direction: East. "G": move one step. Position: (0, 0). Direction: East. "L": turn 90 degrees anti-clockwise. Position: (0, 0). Direction: North. Repeating the instructions, the robot goes into the cycle: (0, 0) --> (0, 1) --> (-1, 1) --> (-1, 0) --> (0, 0). Based on that, we return true.

Constraints:

* `1 <= instructions.length <= 100` * `instructions[i]` is `'G'`, `'L'` or, `'R'`.

Code Snippets

C++:

```
class Solution {
public:
bool isRobotBounded(string instructions) {

}
};
```

Java:

```
class Solution {
public boolean isRobotBounded(String instructions) {

}
```

```
}
```

Python3:

```
class Solution:
def isRobotBounded(self, instructions: str) -> bool:
```