1041. Robot Bounded In Circle

Created	@June 19, 2021 8:03 AM
Tags	Medium
link	https://leetcode.com/problems/robot-bounded-in-circle/

Description

On an infinite plane, a robot initially stands at `(0, 0)` and faces north. The robot can receive one of three instructions:

- `"G"`: go straight 1 unit;
- `"L"`: turn 90 degrees to the left;
- `"R"`: turn 90 degrees to the right.

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.

Approach

- If the robot is not facing north, then after 4 cycles it will return to its original position in all cases. (consider leetcode article for proof)
- If the robot reaches 0,0 within one cycle then final direction does not matter.

```
class Solution {
public:

    int getFinalDistance(int x, int y) {
        return sqrt(x*x + y*y);
    }

    char isFinalDirectionChanged(int x, int y, char cur_direction) {
        if (cur_direction != 'N') return true;

        if (getFinalDistance(x, y) == 0) return true;

        return false;
```

```
    }

    void updateDirection(char &cur_direction, char dir) {
        switch (cur_direction) {
            case 'N':
                dir == 'L' ? cur_direction = 'W' : cur_direction = 'E';
                break;

            case 'S':
                dir == 'L' ? cur_direction = 'E' : cur_direction = 'W';
                break;

            case 'E':
                dir == 'L' ? cur_direction = 'N' : cur_direction = 'S';
                break;

            case 'W':
                dir == 'L' ? cur_direction = 'S' : cur_direction = 'N';
                break;

        }
    }

    void updatePosition(int &x, int &y, char cur_direction) {

        if (cur_direction == 'N') y += 1;
        else if (cur_direction == 'S') y -= 1;
        else if (cur_direction == 'E') x += 1;
        else if (cur_direction == 'W') x -= 1;
    }

    bool isRobotBounded(string instructions) {
        int x = 0; int y = 0; // initial position of robot
        char cur_direction = 'N'; // initial direction of robot

        for (auto i: instructions) {

            if (i == 'G')
                updatePosition(x, y, cur_direction);
            else
                updateDirection(cur_direction, i);

        }

        return isFinalDirectionChanged(x,y,cur_direction) or getFinalDistance(x,y) == 0;

    }
};
```