657 Judge Route Circle

2018年4月22日 13:52

Initially, there is a Robot at position (0, 0). Given a sequence of its moves, judge if this robot makes a circle, which means it moves back to **the original place**.

The move sequence is represented by a string. And each move is represent by a character. The valid robot moves are R (Right), L (Left), U (Up) and D (down). The output should be true or false representing whether the robot makes a circle.

Example 1:

Input: "UD"

Output: true

Example 2:

Input: "LL"

Output: false

来自 <https://leetcode.com/problems/judge-route-circle/description/>

初始位置 (0, 0) 处有一个机器人。给出它的一系列动作，判断这个机器人的移动路线是否形成一个圆圈，换言之就是判断它是否会移回到**原来的位置**。

移动顺序由一个字符串表示。每一个动作都是由一个字符来表示的。机器人有效的动作有 R（右），L（左），U（上）和 D（下）。输出应为 true 或 false，表示机器人移动路线是否成圈。

Solution for Python3:

```
1  class Solution1:
2      def judgeCircle(self, moves):
3          """
4          :type moves: str
5          :rtype: bool
6          """
7          import collections
8          C = collections.Counter(moves)
9          return C['L'] == C['R'] and C['U'] == C['D']
10
11 class Solution2:
12     def judgeCircle(self, moves):
13         """
14         :type moves: str
15         :rtype: bool
16         """
17         x = y = 0;
```

```
        for move in moves:
            if move == 'U':
                y -= 1
            elif move == 'D':
                y += 1
            elif move == 'L':
                x -= 1
            elif move == 'R':
                x += 1
        return x == y == 0
```

Solution for C++:

```
class Solution1 {
public:
    bool judgeCircle(string moves) {
        unordered_map<char, int> m;
        for (char s : moves) {
            m[s]++;
        }
        return m['L'] == m['R'] && m['U'] == m['D'];
    }
};

class Solution2 {
public:
    bool judgeCircle(string moves) {
        int x = 0, y = 0;
        for (char c : moves) {
            if (c == 'U')
                y--;
            else if (c == 'D')
                y++;
            else if (c == 'L')
                x--;
            else if (c == 'R')
                x++;
        }
        return x == 0 && y == 0;
```

```
27         }
28     };
```