

LCP 3. 机器人大冒险

难度 中等

👍 13



📖 收藏

🔗 分享

🌐 切换为英文

📖 题目描述

💬 评论 (12)

🔑 题解 (13) New

🕒 提交记录

力扣团队买了一个可编程机器人，机器人初始位置在原点 $(0, 0)$ 。小伙伴事先给机器人输入一串指令 `command`，机器人就会无限循环这条指令的步骤进行移动。指令有两种：

1. U: 向 y 轴正方向移动一格
2. R: 向 x 轴正方向移动一格。

不幸的是，在 xy 平面上还有一些障碍物，他们的坐标用 `obstacles` 表示。机器人一旦碰到障碍物就会被损毁。

给定终点坐标 (x, y) ，返回机器人能否完好地到达终点。如果能，返回 `true`；否则返回 `false`。

示例 1:

```
输入: command = "URR", obstacles = [], x = 3, y = 2
输出: true
解释: U(0, 1) -> R(1, 1) -> R(2, 1) -> U(2, 2) -> R(3, 2)。
```

第一种方法

暴力求解

1. new 一个二维 `boolean` 类型数组，初始化为 `true`
2. 将二维 `boolean` 类型的数组中对应的障碍物的坐标设置为 `false`，仅仅考虑障碍物的坐标在 x, y 的范围之内
3. While 循环每一个 `command` 的每一个字符，并做出相应的调整，字符为‘U’调整 `pathX++`，字符为‘R’调整 `pathY++`。每次改动 `pathX, pathY` 判断是否遇到障碍，然后在判断是否到达终点。

```
1. public static boolean robot(String command,int[][] obstacles,int x,int y) {
2.     int pathX=0;
3.     int pathY=0;
4.     boolean flag=false;
5.     boolean[][] m=new boolean[x+1][y+1];
6.     for(int i=0;i<=x;i++){
7.         for(int j=0;j<=y;j++){
8.             m[i][j]=true;
9.         }
10.    }
11.    for(int i=0;i<obstacles.length;i++){
12.        int a=obstacles[i][0];
13.        int b=obstacles[i][1];
14.        if(a<=x&&b<=y){
```

```

15.             m[a][b]=false;
16.         }
17.     }
18.     int count=0;
19.     while(pathX<=x&&pathY<=y){
20.         char temp=command.charAt(count++);
21.         if(count==command.length()){
22.             count=0;
23.         }
24.         if(temp=='U'){
25.             pathY++;
26.         }else if(temp=='R'){
27.             pathX++;
28.         }
29.         if(!m[pathX][pathY]){
30.             break;
31.         }
32.         if(pathX==x&&pathY==y){
33.             flag=true;
34.             break;
35.         }
36.
37.     }
38.     return flag;
39. }

```

提交结果

机器人迷宫

提交记录

48 / 53 个测试用例通过

状态: 超出内存限制

提交时间: 0 分钟之前

第二种方法

```

1. public static boolean robot(String command,int[][] obstacles,int x,int y) {
2.     // TODO Auto-generated method stub
3.     for(int[] a:obstacles){
4.         int pX=a[0];
5.         int pY=a[1];
6.         if(pX+pY>x+y){
7.             continue;
8.         }
9.         if(getC(pX,pY,command)){
10.             return false;

```

```

11.     }
12. }
13.
14.     return getC(x,y,command);
15.
16. }
17.
18. private static boolean getC(int X, int Y, String command) {
19.     // TODO Auto-generated method stub
20.     int time=(X+Y)/command.length();
21.     int extra=(X+Y)%command.length();
22.     int[] xyStep=getDirectionCount(command,command.length());
23.     int[] xyStepE=getDirectionCount(command,extra);
24.     xyStep[0]=xyStep[0]*time+xyStepE[0];//U的个数
25.     xyStep[1]=xyStep[1]*time+xyStepE[1];//R的个数
26.     return Y==xyStep[0]&&X==xyStep[1];
27. }
28.
29. private static int[] getDirectionCount(String command, int length) {
30.     // TODO Auto-generated method stub
31.     int[] count=new int[]{0,0};
32.     char[] charArr=command.toCharArray();
33.     System.out.println("length:"+length);
34.     System.out.println("charArr.length:"+charArr.length);
35.     for(int i=0;i<length;i++){
36.         if(charArr[i]=='U'){
37.             count[0]++;
38.         }else {
39.             count[1]++;
40.         }
41.     }
42.     return count;
43. }

```

提交结果

机器人大冒险

提交记录

53 / 53 个通过测试用例

执行用时: 0 ms

状态: 通过

提交时间: 8 分钟之前