

题目：

机器人在一个无限大小的网格上行走，从点 (0, 0) 处开始出发，面向北方。该机器人可以接收以下三种类型的命令：

- -2：向左转 90 度
- -1：向右转 90 度
- $1 \leq x \leq 9$ ：向前移动 x 个单位长度

在网格上有一些格子被视为障碍物。

第 i 个障碍物位于网格点 (`obstacles[i][0]`, `obstacles[i][1]`)

机器人无法走到障碍物上，它将会停留在障碍物的前一个网格方块上，但仍然可以继续该路线的其余部分。

返回从原点到机器人所有经过的路径点（坐标为整数）的最大欧式距离的平方。

示例 1：

输入：`commands = [4,-1,3]`, `obstacles = []`
输出：25
解释：机器人将会到达 (3, 4)

示例 2：

输入：`commands = [4,-1,4,-2,4]`, `obstacles = [[2,4]]`
输出：65

解释：机器人在左转走到 (1, 8) 之前将被困在 (1, 4) 处

提示：

1. $0 \leq \text{commands.length} \leq 10000$
2. $0 \leq \text{obstacles.length} \leq 10000$
3. $-30000 \leq \text{obstacle}[i][0] \leq 30000$
4. $-30000 \leq \text{obstacle}[i][1] \leq 30000$
5. 答案保证小于 2^{31}

思路：

设置东南西北四个方向：（可参考扫雷的思路）

(0,1) 北 (1, 0) 东 (0, -1) 南 (-1, 0) 西

`dx = [0, 1, 0, -1]`

`dy = [1, 0, -1, 0]`

`di`是当前方向，

向左： $(di + 3) \% 4$

向右： $(di + 1) \% 4$

判断是否遇到障碍物

```

1 def robotSim(self, commands, obstacles):
2     """
3     :type commands: List[int]
4     :type obstacles: List[List[int]]
5     :rtype: int
6     """
7     # (dx,dy)
8     # (0,1) 北 (1, 0) 东 (0, -1) 南 (-1, 0) 西
9     dx = [0, 1, 0, -1]
10    dy = [1, 0, -1, 0]
11    # x, y是当前值, (0, 0)
12    # di是当前指的方向 0, 0:向北 (上), 1, 向东 (右), 2 向南 (下), 3 向西,
13    x = y = di = 0
14    # 注意, 这里要是set
15    obstacleSet = set(map(tuple, obstacles))
16    ans = 0
17
18    for cmd in commands:
19        if cmd == -2: #left
20            di = (di + 3) % 4
21        elif cmd == -1: #right
22            di = (di + 1) % 4
23        else:
24            for k in xrange(cmd):
25                # 判断当前点是否是障碍物, 是, 就更新
26                if (x + dx[di], y + dy[di]) not in obstacleSet:
27                    x += dx[di]
28                    y += dy[di]
29                    ans = max(ans, x*x + y*y)
30
31    return ans

```