

演员的自我修养

Leetcode 1001 Grid Illumination

📅 2019-02-24

给定一个NxN的网格，每个格子(x,y)都放着一盏灯。

初始时，有些灯是亮着的，`lamps[i]` 告诉我们第i个亮着的灯的位置。每盏灯会照亮和它处于同一水平线，同一垂直线，同一对角线（左右）上的所有灯。

对于第i个询问 `queries[i] = (x,y)`，我们需要回答此时(x,y)位置上是否被照亮，1代表是，0反之。

在每一次询问结束之后，会将询问处的灯自身以及周围的8个格子（包括对角）的所有灯熄灭。

返回一个数组answer，answer[i]对应第i个queries。

Example 1:

Input: N = 5, lamps = [[0,0],[4,4]], queries = [[1,1],[1,0]]

Output: [1,0]

Explanation:

Before performing the first query we have both lamps [0,0] and [4,4] on.

The grid representing which cells are lit looks like this, where [0,0] is the top left corner, and [4,4] is the bottom right corner:

1 1 1 1 1

1 1 0 0 1

1 0 1 0 1

1 0 0 1 1

1 1 1 1 1

Then the query at [1, 1] returns 1 because the cell is lit. After this query, the lamp at [0, 0] turns off, and the grid now looks like this:

```
1 0 0 0 1
```

```
0 1 0 0 1
```

```
0 0 1 0 1
```

```
0 0 0 1 1
```

```
1 1 1 1 1
```

Before performing the second query we have only the lamp [4,4] on. Now the query at [1,0] returns 0, because the cell is no longer lit.

Note:

1. $1 \leq N \leq 10^9$

2. $0 \leq \text{lamps.length} \leq 20000$

3. $0 \leq \text{queries.length} \leq 20000$

4. $\text{lamps}[i].\text{length} == \text{queries}[i].\text{length} == 2$

题意分析：

在一个很大很大的方格中($N < 10^9$)有若干灯，每个灯能照亮横排，竖排，左斜排，右斜排

现在开始询问，问某个格子是否被照亮，然后将这个格子为中心的9个空格的灯全部熄灭，进行下一次询问

思路分析：

那我们知道，既然方格很大，我们肯定是不可能用数组去存每个格子的状态的

那如果你做过 **N皇后问题**，你应该对这个结论有印象：

在同一条左斜线上的点，方程式都形如 $x+y = c$ ，也就是他们的坐标之和相等

在同一条右斜线上的点，方程式都形如 $y = x+c$ ，也就是他们的坐标之差相等

有了这个结论，这道题中我们只需要去记录坐标之间的关系即可

建立四个集合，分别为横轴，竖轴，左斜轴，右斜轴，将坐标加入其中，然后每次判断询问点的坐标是否存在于这些集合中即可

```
1 class Solution(object):
2     def gridIllumination(self, N, lamps, queries):
3         from collections import Counter
4         lamps = list(map(tuple, lamps))
5         light_set = set(lamps)
6         horizontal = Counter() # 横
7         vertical = Counter()   # 竖
8         l_oblique = Counter()  # 左斜
9         r_oblique = Counter()  # 右斜
10        for x, y in lamps:
11            horizontal[x] += 1
12            vertical[y] += 1
13            l_oblique[x+y] += 1
14            r_oblique[y-x] += 1
15
16        res = []
17        for x,y in queries:
18            if x in horizontal or y in vertical or x+y in l_oblique:
19                res.append(1)
20            else:
21                res.append(0)
22            for dx in [-1, 0, 1]:
23                for dy in [-1, 0, 1]:
24                    xpos, ypos = x + dx, y + dy
25                    if (xpos, ypos) in light_set: # 如果附近有灯, 则熄
26                        light_set.remove((xpos, ypos))
27                        horizontal[xpos] -= 1
28                        if horizontal[xpos] == 0: del horizontal[xpos]
29
30                        vertical[ypos] -= 1
31                        if vertical[ypos] == 0: del vertical[ypos]
32
33                        l_oblique[xpos+ypos] -= 1
34                        if l_oblique[xpos+ypos] == 0: del l_oblique[xpos+ypos]
35
36                        r_oblique[ypos-xpos] -= 1
37                        if r_oblique[ypos-xpos] == 0: del r_oblique[ypos-xpos]
38        return res
```

Leetcode problems # implementation

◀ Leetcode 998 Maximum Binary Tree II

Leetcode 993 Cousins in Binary Tree ▶

昵称

邮箱

网址(http://)

anything you want



表情 | 预览

回复

2 评论



Anonymous

Chrome 73.0.3683.86

Mac OS 10.14.3

2019-04-20

回复

haoti



lei_hsia

Opera 58.0.3135.68

Mac OS 10.11.2

2019-02-28

回复

同学， 你的网站的js写的真好

Powered By Valine

v1.3.6

© 2019 王琛

本站访客数:20972 本站总访问量43924次