CHECKERS 解题报告

‹1› 题目大意

给定一个 N*N 的网格，上面有一些棋子，保证棋子数为三角形数。按照跳棋的规则，方向为(+,0)，(-,0)，(0,-)，(0,+)，(-,+)，(+,-)六个，问需要跳多少步使得所有的棋子填满左上角的三角形区域，希望步数尽量少。

‹2› 解题思路

跳棋为了减少步数应该更多的连跳。而对于一个串连跳，一个棋子需要的不是一条都是棋子的链，也不是空的链，而是一个有棋子和没有棋子交叉的链。因此可以将链中有棋子的位置看作边(跳的方向是固定的)，没有棋子的位置看作点(棋子可以在这个位置停留，选择方向)。根据给定的六个方向，我们可以将坐标(x,y)中满足 x，y 均为偶数的格子当作点（称作点格子），其他点当作边（称作边格子）。这样可以发现，每一个做边的格子，连接两个点，每个点周围有六条边。如果边上有棋子，相当与其连接的两个点可以互相到达。当然这需要满足所有的偶数格子的点是空的。这样就可将棋盘完全转化成对应的图了。

然后要使整个图中的所有边都与三角区域内的点联通，这样才能保证任何一条边都能快速到位。接着先将所有位于目标三角形内的边填满，再将点填满。在填目标三角形的时候，有可能使得边不再联通，所以要每次都要重新调整，使得所有边联通。

综上算法总共分为 4 大块，1 使所有点格子置空，2 使整个图中的边都与三角区域内的点联通，3 覆盖三角区域的边格子，4 覆盖三角区域的点格子。

比较麻烦的时 2 操作。如果当前不联通，则需要找到一个不联通的一块，使得它尽量朝另一个联通块移动(不一定是与三角区域联通的块)，并且不能使得这个块分裂成两个块。

3 操作每次至多移动 3 步。

4 操作每次至多移动 2 步。

所以可以达到一个较优的解。

‹3› 算法流程 & 复杂度分析

首先执行 1，然后执行 3 直到都填满，然后执行 4 直到都填满，每次在 3，4 操作前都要执行 2 操作。

1 操作对于每个有棋子的点搜索空的边，每次进行 bfs，总共可能移动 O(n^2)次，总时间复杂度为 O(n^4)。

2 操作对于最多每个块需要移动 O(n)次，每一次需要做双联通分量找可以割的边，还需要通过 bfs 找能移到的最好的格子，时间复杂度为 O(n^2)。最多有 O(n^2)个联通块。总时间复杂度为 O(n^5)。

3，4 操作，需要 bfs 查找可以移动到的格子，总共有 O(n^2)个格子，总时间复杂度为 O(n^4)。

综上，总时间复杂度为 O(n^5)，空间复杂度为 O(n^2)。

题目链接 https://www.codechef.com/problems/CHECKERS